DISPLAYFAB

USER GUIDE

Guide Revision 1.0

TECHOOKA LABS

Thank you!

Thank you for purchasing this product. This user guide serves as an introduction to using DisplayFab. Please visit our website for full documentation.

This product package comes with various examples ranging from Basic to Advanced levels to help you get started with DisplayFab.

About DisplayFab

DisplayFab is a Unity3D Extension by Techooka Labs that lets you display data from Unity Components, PlayerPrefs, Singletons, Game Managers and other scripts and display it on UI elements without writing any code.

Eliminate the need to create scripts (and writing code) to display data on your UI elements. By using DisplayFab, you can speed up your UI design process by up to 400%.

You can display data on your UI elements in as few as four point-and-click steps: Choose your target UIs, choose your source data, link-up the target UI to your source data and Execute. That's as simple as it gets!

Or you can reverse the process and save or update the data from your Unity UI elements. You just have to reverse the targets and the sources and you're done!

Website

http://displayfab.techooka.com

Complete and most up-to-date Documentation is available at

http://displayfab.techooka.com/documentation

DisplayFab Overview

The DisplayFab Extension has many different modules and addons but the core component and functionality revolves around the DisplayFabSystem component.

A typical DisplayFabSystem setup involves 4 operations: Setting up Targets, Setting Up Sources, Linking Targets with Sources, and finally, executing those link-ups.

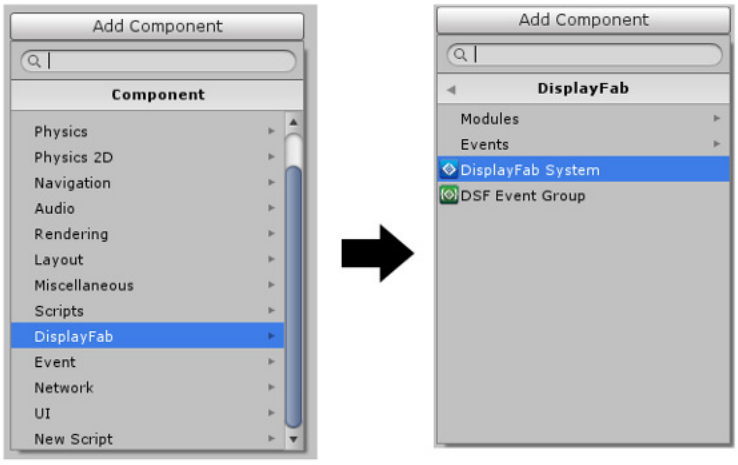
Typically, Targets are UI Displays while Sources are data sources. However, in many cases, a source could be another UI element. If you wanted to store/update data from a UI element, you would inverse the process and setup a data source as a target instead and the UI target as a source.

Executions happen typically upon a particular UI event (such as a button click), or upon loading of a scene (such as in a Start() function).

This user guide explains the basics of DisplayFabSystem Component only.
The extended and most up-to-date documentation is available online at our website and includes documentation for all other modules and addons of DisplayFab as well as advanced features of the DisplayFabSystem.

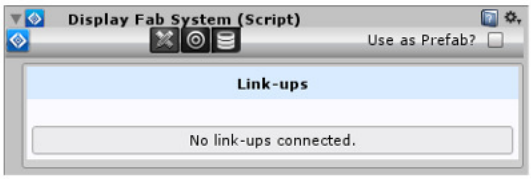
Adding DisplayFabSystem Component

You can add a DisplayFabSystem component in Unity from the Component menu (Components -> DisplayFab -> DisplayFabSystem).



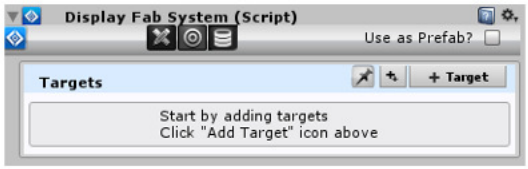
The DisplayFabSystem Basics

The DisplayFabSystem consists of three tabs : *Link-Ups, Targets, Sources.*



Link-Ups

This is the first tab. Link-Ups automatically show up once you link the targets with the sources.
You can remove a link-up by right clicking on a link-up.

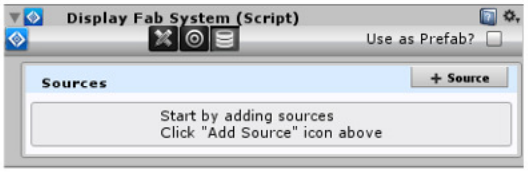


Targets

Clicking on this button adds a new Target entry. We call it a 'ts' Entry (ts stands for a target/source).
Every ts Entry has a unique identifier and this number is unique across all targets and sources within a DisplayFabSystem

Batch-Select mode: Lets you select and add more than one target.

'Easy-Link-Up Mode': When enabled, 'pins' the target panel above and the source panel below to enable easy linking. If this mode is not ON, the inspector switches between both panel views during linking.



Sources

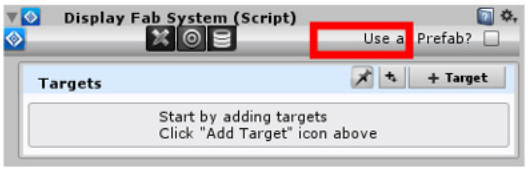
This panel is identical to the Targets panel, except that there is no way to do a 'Batch-select' of sources (as of now).

To add a source entry, click on the '+ Source' button.

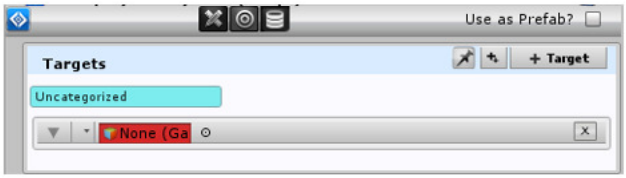
Use as Prefab

If DisplayFabSystem is being added to a Prefab or a Prefab instance (or if you intend to convert the GameObject into a prefab), checking this makes sure you don't accidentally select 'non-prefab' (or non-child of current prefab) targets and sources.

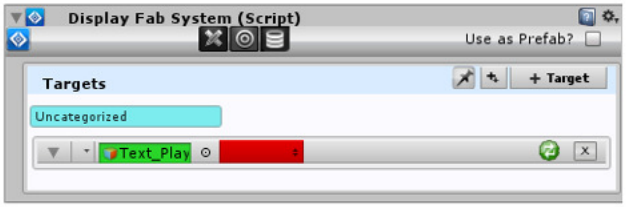
Adding Targets to DisplayFabSystem



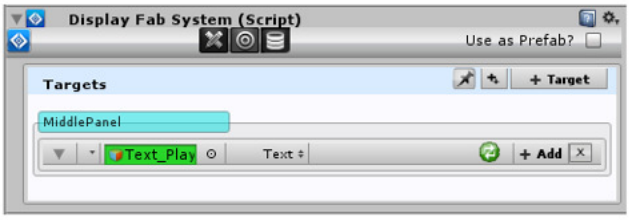
Step 1. Click '+ Target' button.



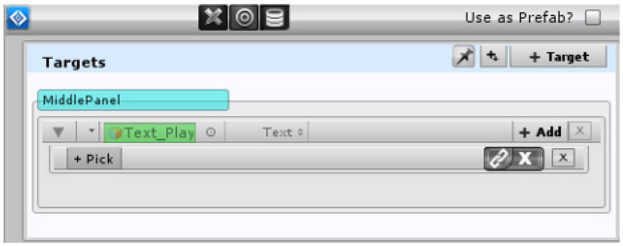
Step 2. Drag & Drop a GameObject to the red field.
If a field is not selected or if it is of incorrect type, this field will be highlighted in red. Otherwise, it will be highlighted in green.



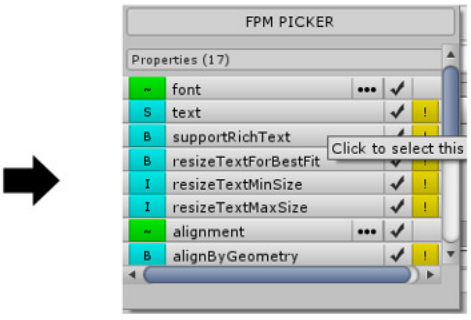
Step 3. Select the component by clicking on the red highlighted drop-down.
Alternatively, you can click on the button to let DisplayFab attempt to auto-select the component and it's most commonly used property. Currently, DisplayFab automatically detects most common UI properties such as Text.text, Slider.value etc. For now, we'll add it using the regular way.



Step 4. Click "+ Add" to add a new empty 'FPM' entry.
FPM stands for a 'field/property/method'. When you click on that button, an empty fpm entry is added. Every fpmEntry has an unique Identifier within a tsEntry.



Step 5. Click "+ Pick" to pick a property/-field/method from the FPM Picker.



Indicates this fpm is a non-primitive type (a class, for example)

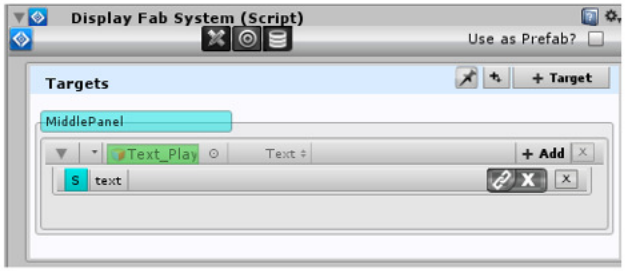
Indicates this is the last recognizable fpm in the hierarchy . (No more child FPMs available to traverse)

Indicates there are child FPMs in its hierarchy that you can navigate into and pick them.

Step 6 Click on the button to pick the appropriate field, property or method (fpm)

Adding Targets to DisplayFabSystem (Continued)

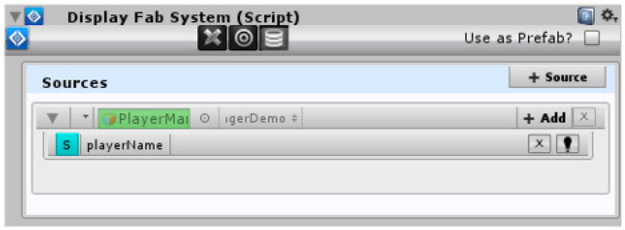
Once the fpm is picked, the end result is as follows:



Link-Button & Icon:
You'll notice that these buttons will be displayed only in the Targets.
However, if a source is linked, it will also show up next to a source FPM (but only as an icon).
Clicking on this button lets you link a target to a source.
1. If a Target has only 1 fpm entry, and if the target is minimized, then this icon shows up in the tsEntry line for convenience.
2. Each target can only have 1 link to a source but a source can have multiple targets connecting to it.

Adding Sources to DisplayFabSystem

Adding Sources is exactly similar to the steps we followed to add Targets in the previous page.

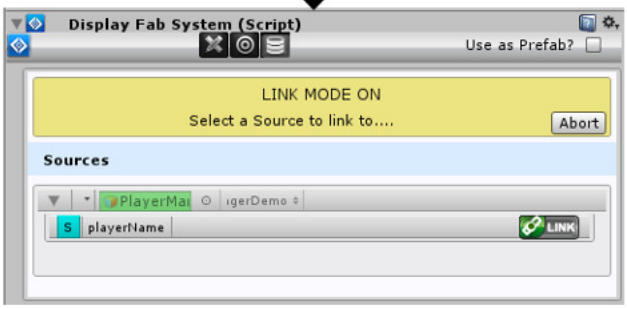


Quick-Peek: *Click on this button to view the output of this source fpm in the Editor. Useful for debugging purposes.*

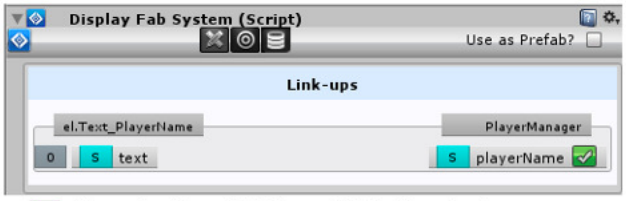
Removes the fpm entry.
Note: If a fpm entry has child FPMs, you have to remove all child FPMs first.

Linking Targets to the Sources

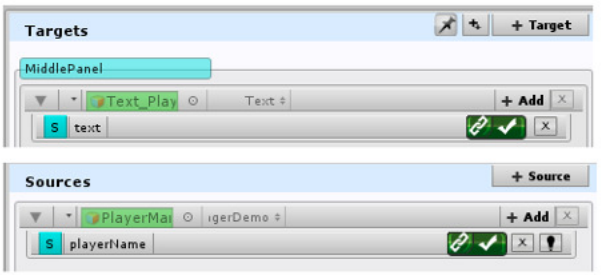
Step 1. Click the button next to the target you want to link from.
(You may have to switch to the 'Target' tab first.)



Step 2. Click the LINK button next to the source FPM you wish to link the target to.
Note that the view automatically switches to the 'Source' tab



Converter Icon: *Clicking on this button also shows up a 'Converter' popup that lets you define converters to convert source data from one type to another. Check online documentation on how to use converters.*



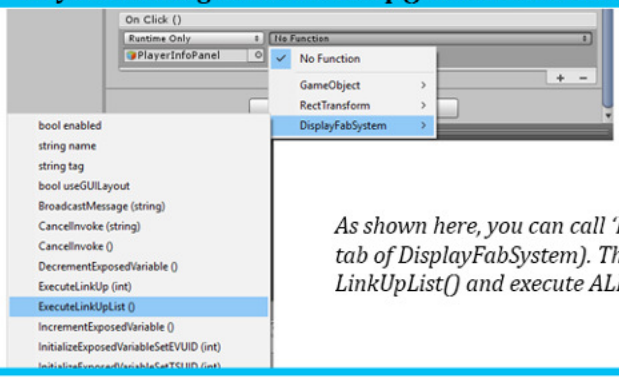
Step 3. Link Successfully setup!
The link icon turns green and a new link-up shows up in the Link-ups tab.

Executing the DisplayFabSystem's LinkUp List

Once all desired targets (eg. UI displays) and sources (eg. data) are linked-up, we need to execute them. Typically, we'd execute it based on UI input events (such as button clicks, slider value change etc).

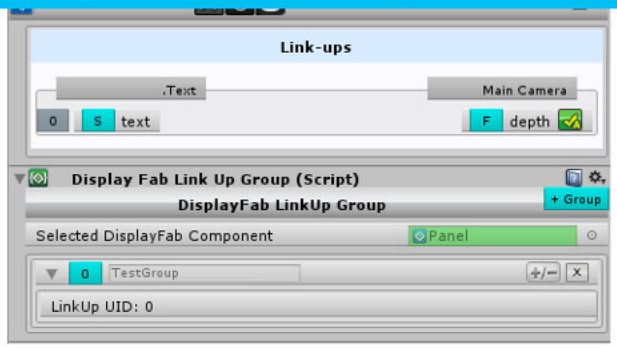
Different Ways to Execute a Link-Up

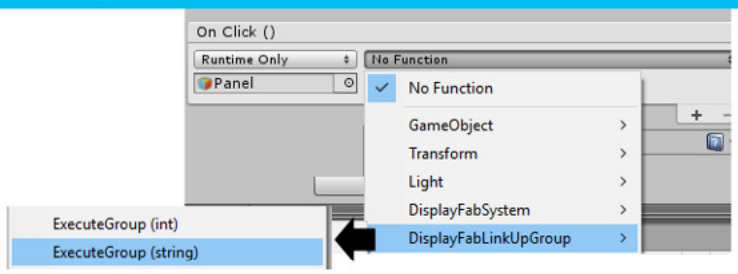
Way 1. Invoking ExecuteLinkUp() or ExecuteLinkUpList() by setting up an event in one of Unity's UI events



As shown here, you can call 'ExecuteLinkUp and pass the unique linkUp UID (check the UID in the LinkUp tab of DisplayFabSystem). This will execute only ONE specific Link-Up. Alternatively, you can call 'Execute-LinkUpList() and execute ALL link-ups defined in the DisplayFabSystem.

Way 2. Using DisplayFabGroup and invoking 'ExecuteGroup'



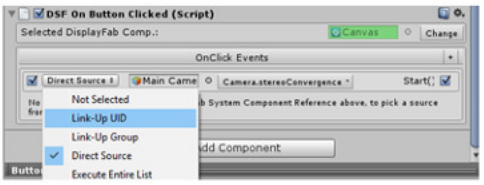


You can execute a group either using its unique ID or by the unique group Name.

First you will add the DisplayFab LinkUp Group component and add a group and add the link-ups to the group as desired.
Note: You can find how to use DisplayFabGroup in the online documentation.

Way 3. Using DisplayFabSystem's Event Executor Components

This is the most preferred and flexible way among all the three.
DisplayFab comes with several Event Executors for the primary UI controls.
Using those, you can execute the LinkUps in an intuitive way by using drop-down menus to select the LinkUps or LinkUp Groups.



There's also a 'Direct Source' execution type that allows you to directly modify properties (similar to the Unity's 'Dynamic Properties') and methods but with the added facility of using 'converters' to convert and/or format data.

Check out our tutorials and online documentation on how to use these Executors.

DisplayFab Modules

DisplayFab Modules are powerful complementary components of the DisplayFabSystem component that let you do more powerful and wonderful things using the DisplayFabSystem.

As of the initial version, there are 2 modules bundled with the Extension:
The DisplayFab Instantiator and DisplayFab Exposed Variable Executor.

As these modules are an advanced feature of DisplayFab, please refer to our online documentation for help with using these modules.

More modules might be added in future to extend the functionality of the DisplayFabSystem.

Further Help/Extensive Documentation

For further documentation and examples on advanced usage of DisplayFabSystem including using Arrays, Exposed Variables, Converters, Singletons, PlayerPrefs, Static Classes, etc, please refer to our online documentation website.