

AF Countries & Flags

User Manual & API Reference

Arcana Futura

Introduction

AF Countries & Flags is a ready-to-use Unity uGUI country selector with a supplied country database, rectangular and square flags, ISO and calling-code data, currencies, continents, groups, search, sorting, filtering, and multiple row layouts.

Each selector instance is independent. You can expose all supplied published countries or curate a subset for a particular screen without changing the supplied database.

Requirements

- Unity uGUI
- TextMeshPro
- No third-party dependencies
- No internet connection or external service is required at runtime.

Quick Start

1. Add the supplied AF Countries And Flags prefab to a Canvas.
2. Select the component and configure it in the Inspector.
3. Choose the button display, row type, popup placement and flag style you want.
4. Use the Selections tab if the selector should expose only a subset of the supplied data.
5. Enter Play Mode and click the selector.

The popup is created when first needed and then reused. A Custom row containing several text fields is naturally heavier to initialize than the simpler row layouts; subsequent opens reuse the initialized popup.

Inspector Overview

The custom Inspector is divided into five main tabs: Settings, Design, Selections, Assignments, and Events.

Settings

Country Name controls which supplied country name is displayed. English uses the main country name. Local uses the local name when available. Local Latin prefers the Latin-script local name and falls back when needed.

Flag Style selects Rectangular or Square flags.

Button Display controls the selected-country button presentation. The supplied options cover flag-and-name, name-only, and flag-focused presentations.

Row Type selects Standard, Compact, Tiny, or Custom. Standard is the general-purpose flag-and-name row. Compact is text focused. Tiny is intended for narrow, flag-focused selectors. Custom can display additional country and currency data.

Popup Placement can be Centered, Below, or Above. Popup width can match the selector button or use a custom width. Popup height is configurable.

Filters lets you independently enable Search, Sort, Continent, and Group controls. Select All and Select None are provided for convenience. Disabling all four removes the filter bar, which is useful for very small selectors.

Design

Typography lets you assign a TextMeshPro font and font size for the selector rows.

Continent Filter Popup and Group Filter Popup settings control the width, maximum height, and alignment of their filter popups.

Selections

Selections controls which supplied data is available to each selector instance. It does not modify the supplied database.

Countries: choose which published countries can appear in the selector. Search and filter tools make large selections easier to manage.

Continents: choose which continent choices are available in the Continent runtime filter.

Groups: choose which group choices are available in the Group runtime filter.

Currencies: choose which currency data may be displayed or returned by this selector. Currency selection does not remove a country from the selector.

Reset Default returns a selection category to its unrestricted default. For Countries, the default is all supplied published countries. For Continents, Groups, and Currencies, the default is all available entries.

Assignments

The supplied prefab is already configured. Assignments exposes the database, internal popup/blocker/filter prefabs, row prefabs, and button objects so the component remains transparent and customizable.

Internal prefabs are implementation parts of the selector. Normally there is no reason to replace them unless you are intentionally restyling or extending the UI.

Events

On Country Selected is a `UnityEvent<AFCountry>`. It is invoked after a country is selected and the selector state/button have been updated. Use it to connect selection to your own application logic without modifying AF Countries & Flags.

Runtime Behaviour

The selector creates its popup lazily on first open and reuses it afterward. Clicking outside the popup closes it. Continent and Group filter popups use their own blocker so an outside click closes the small filter popup without closing the main country selector.

Search is live. It matches country name, local name, Latin local name, ISO Alpha-2, ISO Alpha-3, and calling code. Currency fields are not part of the search.

Sorting switches between ascending and descending country-name order. Continent and Group filters can be combined with search and sorting.

Selection persistence belongs to your application. AF Countries & Flags reports and displays the current selection but does not impose a save system.

Custom Row Data

The Custom row can expose additional fields such as ISO Alpha-2, ISO Alpha-3, calling code, currency name, currency code, and currency symbol. The exact fields are selected per selector in Settings.

If a selected currency restriction excludes a country's currency, the country remains available; only that currency data is omitted.

Data Notes

The supplied database is intended as practical application data. Published records are the entries exposed by the Regular selector. Some special or nonstandard records may remain in the database unpublished and are not selectable by default.

Continents and groups are separate concepts. Continents provide geographic classification; groups provide useful regional or organizational groupings supplied with the asset.

AF Countries & Flags does not attempt to adjudicate sovereignty or political status. The supplied data is a practical baseline for software UI.

The Regular edition is designed for curating the supplied dataset per selector. It does not include the development seed JSON or the internal database rebuild tool.

Customization

The supplied prefabs are ordinary Unity uGUI objects and can be styled to match your project. You can change colors, sprites, spacing, layout settings, TextMeshPro presentation, and other visual properties in the normal Unity workflow.

Keep the required component references intact when modifying prefabs. The Assignments tab makes those references visible if you need to inspect or replace them.

API Reference

AFCountriesAndFlags

Namespace: ArcanaFutura.UI.CountriesAndFlags

SelectedCountry

```
public AFCountry SelectedCountry { get; }
```

Returns the currently selected country, or null when no country has been selected.

IsOpen

```
public bool IsOpen { get; }
```

Returns true while the main selector popup is open.

Toggle()

```
public void Toggle()
```

Opens the selector when closed and closes it when open.

Open()

```
public void Open()
```

Opens the selector. The popup is initialized on first use and reused afterward.

Close()

```
public void Close()
```

Closes the selector and its blocker.

SelectCountry(AFCountry country)

```
public void SelectCountry(AFCountry country)
```

Selects the supplied country, refreshes the UI, invokes On Country Selected, and closes the selector. A null value is ignored.

SetCountryNameDisplay(AFCountryNameDisplay display)

```
public void SetCountryNameDisplay(AFCountryNameDisplay display)
```

Changes the runtime country-name presentation and refreshes existing rows and the selector button.

Selection Example

```
using ArcanaFutura.UI.CountriesAndFlags;
using UnityEngine;

public class CountrySelectionExample : MonoBehaviour
{
    [SerializeField]
    private AFCountriesAndFlags selector;

    public void PrintCurrentCountry()
    {
        AFCountry country = selector.SelectedCountry;

        if (country != null)
            Debug.Log(country.Name);
    }

    public void OpenCountrySelector()
    {
        selector.Open();
    }
}
```

AFCountryDatabaseRuntime

AFCountryDatabaseRuntime is the read/query layer over an AFCountryDatabase. Its constructor builds lookup indexes for countries, currencies, continents, groups, and their relationships.

Constructor

```
public AFCountryDatabaseRuntime(AFCountryDatabase database)
```

Database

```
public AFCountryDatabase Database { get; }
```

Returns the database used by this runtime wrapper.

Lookup methods

```
public AFCountry GetCountry(int id)

public AFCurrency GetCurrency(int id)

public AFContinent GetContinent(int id)

public AFGroup GetGroup(int id)
```

Each returns the matching object or null when the ID is not present.

Relationship methods

```
public IReadOnlyList<AFCurrency> GetCurrenciesForCountry(int countryId)

public IReadOnlyList<AFCurrency> GetCurrenciesForCountry(AFCountry country)


public IReadOnlyList<AFCountry> GetCountriesForCurrency(int currencyId)

public IReadOnlyList<AFCountry> GetCountriesForCurrency(AFCurrency currency)


public IReadOnlyList<AFCountry> GetCountriesForContinent(int continentId)

public IReadOnlyList<AFCountry> GetCountriesForContinent(AFContinent continent)


public IReadOnlyList<AFGroup> GetGroupsForCountry(int countryId)

public IReadOnlyList<AFGroup> GetGroupsForCountry(AFCountry country)


public IReadOnlyList<AFCountry> GetCountriesForGroup(int groupId)

public IReadOnlyList<AFCountry> GetCountriesForGroup(AFGroup group)
```

Relationship methods return an empty list when no relationship is found.

Country Queries

QueryCountries(AFCountryQuery query)

```
public IReadOnlyList<AFCountry> QueryCountries(AFCountryQuery query)
```

Returns countries matching an AFCountryQuery. The query supports published-only filtering, country-type inclusion, continent, group, included/excluded country IDs, and sorting. Passing null uses a default query.

ResolveCountryName(AFCountry country, AFCountryNameDisplay display)

```
public string ResolveCountryName(  
    AFCountry country,  
    AFCountryNameDisplay display)
```

Returns the appropriate country name for the requested display mode, with fallback to available supplied names.

Data Objects

AFCountry contains the country/entity data used by the selector, including ID, main/local names, ISO data, calling code, continent relationship, country type, published state, and rectangular/square flag references.

AFCurrency contains currency identity and display data such as ID, name, code, symbol, symbol position, and published state.

AFContinent and AFGroup provide the corresponding classification/group records used by filters and database relationships.

AFCountryType distinguishes Country, Territory, ConstituentCountry, and Other records.

Runtime API Notes

The selector's per-instance curation is intentionally separate from AFCountryDatabaseRuntime. The database runtime represents the supplied database relationships; the selector applies its own Countries, Continents, Groups, Currencies, and visible-filter configuration on top.

If your application needs long-term persistence, save a stable identifier such as the selected country's ID or ISO code in your own save system, then resolve/apply the corresponding country when appropriate for your project.

Troubleshooting

Popup does not open: confirm the supplied prefab assignments and that the selector is under an active Canvas.

No countries appear: check the Countries selection in the Inspector and confirm the intended records are enabled.

A continent or group is missing from its filter: check that category under Selections. Restricting Continents or Groups controls available filter choices.

Currency information is blank in a Custom row: check the Currencies selection and the Custom Row Fields settings.

UI looks different after replacing prefabs: compare the Assignments references and required uGUI/TMP components with the supplied prefab.

First Custom open is slower than later opens: Custom creates substantially more UI/text elements than the simpler row types. The popup is reused after initialization.

Contact

Have a question, need help, or found a bug? Please use our online contact form and we'll get back to you as soon as possible.

Website: <https://arcanafutura.eu/>

Support: <https://arcanafutura.eu/asset-support>

License and Disclaimer

AF Countries & Flags is provided under the applicable Unity Asset Store license. It may be used and modified as part of your own commercial or non-commercial projects, subject to the terms of that license.

You may modify the supplied source code for use in your projects. Modification does not grant the right to redistribute or resell AF Countries & Flags itself. The asset, its source code, or modified versions of the asset may not be redistributed, resold, sublicensed or otherwise made available as a standalone asset or development tool.

AF Countries & Flags is provided "as is", without warranties of any kind. Arcana Futura is not liable for losses or damages resulting from its use.

AF Countries & Flags is provided under the Unity Asset Store End User License Agreement (EULA).
<https://unity.com/legal/as-terms>

Arcana Futura