Article: Modding Guide

Modding for Fun and profit (but mostly fun)

Introduction

Ara: History Untold brand-new and what can and can’t be modded will be a bit of a moving target as we race to harden the game to ensure stability while simultaneously opening up more and more of the game to modders.

In other words, what we can mod today is a lot less than what we will be able to mod in the future. This guide will provide modders some of the basic structure for modding in Ara as well as some examples.

Chapter 0: ZNO, ZSchema, & ZData

Ara: History Untold (or just Ara) utilizes a unique Data structure known as ZNO. ZNO is integrated into the Nitrous Engine that powers Ara. There are two key elements to understand when modding: .ZSchema and .ZData.

ZSchema

.ZSchema defines the various fields for a data type and the rules for formatting such a data type. The .ZSchema for a particular type of content can help clarify what fields exist and are expected in the Template Struct.

ZData

.ZData is used to implement new elements and content into Ara. When modding, you will edit or update ZData to achieve your creative vision. To experienced modders and scripters, this will be highly reminiscent of working with . JSON files.

Required Tools

To work with .ZData, you can use any text editor you are familiar with. Common editors used are Microsoft's Visual Studio Code and Notepad++.

Chapter 1: Data Modding

1.1 New Content

.ZData is used to define Items, Resources, Religions, Units, and much more. You can add new elements by adding new .ZData using existing .ZSchema.

We'll start with an example of a new Item. In this example, as a Modder, we want to add a new item to serve as a City Amenity—a Pet Frog.

Start by creating a Folder for the mod, the PetFrogsMod folder. In this folder we're going to create a new .ZData file. Create a blank file in you favored text editor and save it with the .zdata extension. For our example I've saved it as PetFrogItems.zdata in the new PetFrogsMod folder.

Using Items.zschema as a reference I can create a new item entry in the .ZData file. At the top of the file there needs to be a reference to the schema being used. This needs to reference the file name of the .ZSchema.

```
schema Items;
```

each new entry needs to open with “export” and a reference to the template being used. For the example we’re using the ItemTemplate from the Items.zschema.

```
export ItemTemplate itm_petFrog =
{
    .Name = "Pet Frog",
    .Description = "A happy, jumping boy that brings everyone joy.",
    .FlavorText = "He jumps, he brings joy",
    .RecipeID = "rcp_AIAssistants",
    .AtlasID = "AIAssistants",
    .uiRarity = RulesTypes.eRarity.Common,
    .uiType = RulesTypes.ItemType.Luxury,
    .uiHarvestType = RulesTypes.HarvestType.NoneOfTheAbove,
    .Flags = (Flags.HasActive | Flags.Resource| Flags.Consumable),
    .ActivateBuffs =
    {
        "buf_Item_Happiness_15",
        "",
        "",
        "",
        "",
        "",
        "",
        "",
    },
    .uiGiveConsumeItemsForNumTurns = 300u,
    .ActivateBuffsForImprovements =
    {
        "buf_Item_Happiness_15",
        "",
        "",
        "",
        "",
        "",
        "",
        "",
    },
};
```

Here we’ve set up the item to be usable as an Amenity in Cities.

1.2 Existing Content

Instead of adding a new Resource type let’s edit an existing type. To update Data in Ara you are not required to try and edit an existing file. Instead you will create another new file in your text editor. For our example I've saved it as PetFrogResources.zdata in the new PetFrogsMod folder.

Like before I’m using NaturalResources.zschema. So I’m opening the file with a reference to the Schema.

```
schema NaturalResources;
```

This time though I’m not creating an entirely new entry. I’m taking the entry Dyes from the base game NAturalResources0.zdata. I’m going to be making an addition to the type of things that can be harvested from Dyes to include our new Pet Frogs.

```
export NaturalResourceTemplate nrc_Dyes =
{
    .Name = "TXT_ITM_NATURAL_DYES",
    .Description = "TXT_ITM_DESC_NATURAL_DYES",
    .EncartaStrategy = "TXT_ENC_ITM_NATURAL_DYES_DESC_STRATEGY_0",
    .EncartaHistory = "TXT_ENC_ITM_NATURAL_DYES_DESC_HISTORY_0",
    .Asset = "Resource_Dyes",
    .AtlasID = "Dyes",
    .PlacementWeightAdjust = -0i,
    .Rarity = RulesTypes.eRarity.Common,
    .Flags = (NaturalResourceFlags.Hidden | NaturalResourceFlags.Renewable),
    .BiomesBitFlags = (RulesTypes.BiomeFlags.TemperateGrassland | RulesTypes.BiomeFlags.TemperateForest | RulesTypes.BiomeFlags.Mediterranean | RulesTypes.BiomeFlags.Savanna | RulesTypes.BiomeFlags.TemperateRainforest | RulesTypes.BiomeFlags.TropicalForest | RulesTypes.BiomeFlags.TropicalRainforest),
    .uiMinAlt = 2000u,
    .uiMaxAlt = 2100u,
    .fBonusForFreshWater = 1.500000000f,
    .fBonusForAdjacentToCoast = 1.000000000f,
    .MinDepositSize = -1i,
    .MaxDepositSize = -1i,
    .HarvestorType = (RulesTypes.HarvestorType.Farm),
    .HarvestType = (RulesTypes.HarvestType.Plant),
    .HarvestOptions =
    {
        {
            .Item = "itm_Dyes",
            .ProductionRequired = 150u,
            .HarvestCount = 1u,
        },
        {
            .Item = "itm_petFrog",
            .ProductionRequired = 10u,
            .HarvestCount = 1u,
        },
    },
    .BonusForNearbyPlacedResources =
    {
    },
};
```

When we run the project with our Mod enabled the data here will overrule the existing data for Dyes. So any other additional changes made to the data here will impact the game.

(Yes, I’ve made frogs really easy to harvest. Everyone deserves a pet frog.)

1.3 ZData Cheat Sheet

Data Type	Schema File	Overview
Buffs	Buffs.zschema	Buffs are data referenced by many aspects of Ara and is used to do things such as allow Improvements to raise Productivity or Amenities increase how much Food is harvested by Farms.
Governments	Governments.zschema	Governments are unlocked throughout a game of Ara and all

		have their own benefits, drawbacks, and can impact the number of Cities a player can manage.
Improvements	Improvements.zschema	Improvements are the buildings the player places in their city, each having a different effect or benefit.
Items	Items.zschema	Items cover a wide range of things in Ara. An item is everything from the Wool harvested from a Sheep node to the Game Console crafted at an Improvement and utilized as Supply and Amenity.
Masterpieces	Masterpieces.zschema	Masterpieces define the type of Masterpieces a Paragon can randomly unlock when set to produce Masterpieces at an Improvement.
Natural Resources	NaturalResources.zschema	Natural Resources defines the nodes that appear in the world with harvestable items. The NaturalResources data uses Flags to define what Biomes the resource can spawn in and has an array for the options of things that can be harvested from the node.
Paragons	Paragons.zschema	Paragons defines the various Paragons that can join the player randomly throughout a game.
Recipes	Resipes.zschema	Recipes define the rules for Crafting an item and is referenced by Improvements to determine what items can be crafted at them.
Religion Buffs	ReligionBuffs.zschema	Religion Buffs defines the Verses a Player can gain randomly as their Religion expands and what buffs the Verse provides.
Technologies	Technologies.zschema	Technologies defines the various things the player can Research to unlock new things in Ara. The pacing cost is centralized in a list of variables at the top of Technologies0.zdata that you can use as reference for adding new Technologies or editing existing Technologies.

Units	Units.zschema	Units defines the military Units that can be recruited by the player and deployed into Forces.

This list does not encompass all data that exist but some of the most common data types used for adding content to Ara.

Chapter 2: Setting Up The Mod

2.1 GameCoreData Setup

In order to properly load a mod you need to create a new .zdata file named GameCoreData.zdata and include it in the folder with our other .ZData changes. GameCoreData.zdata lets Ara know what all to load in as part of the mod and how to properly read the files.

The Config file for our Pet Frogs Mod will look something like this:

```
schema ZNODataLibrary;
export Library Root =
{
    .Groups =
    {
        .ItemTemplates =
        {
            .FromFiles = { "PetFrogItems.zdata" },
        },

        .NaturalResourceTemplates =
        {
            .FromFiles = { "NaturalResources0.zdata" },
        },
    },
};
```

We are indicating that for this mode we are adding Item Data and Natural Resource data.

2.2 Proper Placement of Mod Folder

Our Mod Folder for Pet Frogs has all three of the files we’ve created in it. Now we move the file to the Mods folder for Ara. In your file explorer try to get to: **%LOCALAPPDATA%\Ara History Untold**

If that doesn’t work navigate to App Data and then to Local. Try finding your folder for Ara manually.

Once in the Local App Data Ara folder open the Mods folder. If one does not exist create a new folder named Mods. Move the folder for your mod into the Mods folder.

2.3 Updating Settings & Loading The Mod

There is one more step in order to make sure Ara will properly load your Mod. Back in the Ara folder under Local App Data there is a file for Settings. Open the file in a text editor. We are looking for a section that looks something like:

```
EnabledMods=0
GameCoreMod0Source=
GameCoreMod1Source=
GameCoreMod2Source=
GameCoreMod3Source=
```

Your setup may look slightly different or be broken up by other settings lines, but these are the variables we care about. We need to change EnabledMods to 1 (for true.) And we need to set one of the GameCourdeMode#Source to our folder name. So the end product will look something like:

```
EnabledMods=1
GameCoreMod0Source=Pet Frogs
GameCoreMod1Source=
GameCoreMod2Source=
GameCoreMod3Source=
```

2.5 Testing Your Work

Your mod should now be ready for testing! Congratulations!

Launch Ara and enjoy your Mod.

Chapter 3: Modding Technologies

Now that you have the basic structures down, let’s make a simple example.

Let’s make techs take longer to research.

In the game we have a single file called Technologies0.zdata that contains all the techs in the game.

Here’s an example:

```
export TechnologyTemplate tch_Archery =
{
    .Name = "TXT_TCH_ARCHERY",
    .Description = "TXT_TCH_DESC_ARCHERY",

    .AtlasID = "Archery",
    .EncartaEntry = "tch_Archery",
    .EncartaStrategy = "TXT_ENC_TCH_ARCHERY_STRATEGY_0",
    .EncartaHistory = "TXT_ENC_TCH_ARCHERY_HISTORY_0",
    .Domains = (RulesTypes.Domain.Military),
    .Era = RulesTypes.TechEras.AncientEra,
    .iRoughDate = -20000i,
    .uiResearchCost = AncientEraCost,
    .SearchTags =
    {
        "TXT_TECHNOLOGY",
        "TXT_ITM_RESEARCH",
        "TXT_DOMAIN",
        "TXT_RARITY",
        "TXT_RTY_UNCOMMON",
        "TXT_TECH_ERA_ANCIENT",
        "TXT_TCH_ARCHERY",
        "TXT_TCH_DESC_ARCHERY",
    },
    .Flags = 0,
    .UnlockRecipesIDs =
    {
        "rcp_Bow",
    },
    .UnlockImprovementsIDs =
    {
      "imp_Watchtower",
       "imp_Camp",
    },
    .UnlockGovernmentsIDs =
    {
    },
    .UnlockNaturalResourcesIDs =
    {
    },
    .UnlockedFormationsIDs =
    {
         "frm_Wedge",
    },
    .UnlockCitySpecialProjects =
    {
    },
    .AddItemsOnUnlock =
    {
        .itm_Prestige = 5u,
    },
    .GrantBuffs =
    {
    },
};
```

Notice that we use a variable for the cost: `AncientEraCost`

We define these costs at the top of the file:
int AncientEraCost = 50;
int BronzeAgeCost = 150;
int IronAgeCost = 300;
int AntiquitiesCost = 400;
int EarlyMiddleAgesCost = 600;
int HighMedievalCost = 800;
int RenaissanceCost = 1200;
int EnlightenmentCost = 1500;
int MachineAgeCost = 1800;
int AtomicAgeCost = 2100;
int InformationAgeCost = 2500;
int FutureAgeCost = 3000;

Let’s make it more expensive:

Rewrite it to be:

`.uiResearchCost = 100,`