Modding XML

## Introduction

Each data def in GalCiv has an associated schema file which defines the names, types, and order of their members. If you are unfamiliar with schema, this is a good resource: <https://www.w3schools.com/xml/schema_intro.asp>

**Folder Structure**

The folder structure for the mod should match the folder structure of the game folder, a Data folder with subfolders Core, Gameplay, and ParticleScriptDefs. Options are not modable. Text can simply go in the Text folder without being nested in a language folder, but I believe that it would work if a modder did want to include multiple language folders, but I don’t think that adding additional fonts would work. Designs and Gfx are also possible subfolders. Screen modding is not enabled at this time.

**Data**

There are multiple kinds of data defs: those with InternalName fields which must be unique within the data type, those with UniqueID fields which use an enumerated type as to enforce uniqueness, data defs that do not have unique identifiers (UIDs) and are members of other data defs, and singleton data defs like GalCivGlobalDefs. In general, there is only one kind of data def in each xml file.

There are four ways to modify the data in GalCiv4: adding new data defs in a unique file, replacing all the defs in a file with the same name, modifying existing data defs, and removing data.

## Add

To add data defs, make a new xml file following the structure outlined in the schema, using unique InternalName or UniqueIDs. Here is an example of adding a tech def:

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>

<TechList

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="../Schema/TechDefs.xsd">

<Tech>

<InternalName>MasterTech\_AdvancedPlanetology</InternalName>

<GenericName>Tech\_AdvancedPlanetology</GenericName>

<DisplayName>MasterTech\_AdvancedPlanetology\_Name</DisplayName>

<TechTree>Master\_Tree</TechTree>

<ShortDescription>MasterTech\_Planetology\_ShortDec</ShortDescription>

<Description>MasterTech\_Planetology\_Dec</Description>

<FlavorName>Tech\_Planetology\_FlavorText\_DisplayName</FlavorName>

<FlavorShortDescription>Tech\_Planetology\_FlavorText\_ShortDescription</FlavorShortDescription>

<FlavorDescription>Tech\_Planetology\_FlavorText\_Description</FlavorDescription>

<ColorDef>TechGreen</ColorDef>

<Icon>Icon\_Planet.png</Icon>

<Image>Event\_Jungle.dds</Image>

<ResearchCost>36</ResearchCost>

<TechPoints>1</TechPoints>

<Tags>Growth</Tags>

<Tags>Critical</Tags>

<AICategoryWeight>

<Military>25</Military>

<Growth>50</Growth>

<Tech>25</Tech>

<Diplomacy>5</Diplomacy>

<Expansion>5</Expansion>

<Wealth>25</Wealth>

<Influence>5</Influence>

<Fortification>5</Fortification>

</AICategoryWeight>

<Prerequ>

<Techs>

<Option>Tech\_Planetology</Option>

</Techs>

<TechAge>

<Option>AgeOfExpansion</Option>

</TechAge>

</Prerequ>

</Tech>

</TechList>

## Modify by File Replacement

A file with the same name in a corresponding folder (either a DLC folder or a Mod folder) will replace the file in the base game folder. For example, in Supernova, we modify the map size defs by placing a file with the name MapSizeDefs.xml in GalCiv4\DLC\Data\Gameplay to override the one in GalCiv4\Data\Gameplay. Since the other file is not read, any defs that are not defined in the overriding file will not be created.

## Modifying Existing Defs

Data defs that have InternalName or UniqueID fields or are marked as singletons in the data can be modified using ModifyDataDefs, which are generated by the schema compiler. The data defs are the same as the normal def, except that all fields except for the UID field are marked as optional. Members which are structs or arrays must be fully replaced as the data compiler does not currently have a way to do comparison on fields that do not have UIDs.

A file with ModifyDataDefs in them can have multiple different types of data defs, as long as they folow the order in the schema (alphabetical).

Here is an example of how to use the modify data defs:

<ModifyDataList

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="../../../../Data/Schema/Generated/ModifyDataDef.xsd">

<BattleActionDef>

<InternalName>SingleMissileVolley</InternalName>

<PerUnitAttackReduction>0.20</PerUnitAttackReduction>

</BattleActionDef>

</ModifyDataList>

## Removing Data

Data defs that have InternalName or UniqueID fields can be removed and not replaced by using the <RemoveDef> structure.

The following removes the SubspaceResonatorB, PlasmaFurnaceB and PrecursorBio-ReplicatorB improvements from the game.

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>

<RemoveList

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="../Schema/Lib/RemoveDef.xsd">

<RemoveDef>

<TypeName>ImprovementDef</TypeName>

<RemoveID>SubspaceResonatorB</RemoveID>

<RemoveID>PlasmaFurnaceB</RemoveID>

<RemoveID>PrecursorBio-ReplicatorB</RemoveID>

</RemoveDef>

</RemoveList>

Tech Details: RemoveDef removes defs by InternalName. The data tables get defragmented so that the data doesn't remain and the indices are updated but empty ones are allowed to remain rather than trying to update the array keys stored on other data types. (This is something that we could address later if needed).

## Modding Art

You can add additional files which are referenced in the data files, or replace existing assets with a file of the same name in the DLC or Mod folder.