layout: index title: GCAM Policy Examples

gcam-version: v5.2

This page includes examples of the inputs required to create policies. Note that each example will need to be tailored to your own needs.

Policy Set-up Overview

This section gives an overview of the available policy options and the tags needed to implement these options. Their use is demonstrated in the examples that follow this section.

Basic Policy Types

`policy-portfolio-standard`: a policy type that allows you to set up taxes, subsidies, constraints, and renewable energy standards. The discussion below refers to `policy-portfolio-standard` setup unless otherwise noted. There are three types of policies: tax, subsidy, or renewable energy standard as specified by the `policyType` tag (`tax`, `subsidy`, `RES`). These polices apply to either prices or quantities, depending on the presence of either a `fixedTax` or `constraint` tag.
`ghgpolicy`: in effect, a special case of a `policy-portfolio-standard` that applies to emissions.

Price policies

`tax` and `subsidy` polices are implemented as price policies when only a `fixedTax` is read in (e.g., no `constraint` is present).

- `policyType: tax`; the model will add values specified in the `fixedTax` to the cost.
- `policyType: subsidy`: the values supplied will be subtracted from the cost.

Either of these creates an unsolved market (see solver section) where the value specified is added to (tax) or subtracted from (subsidy) the cost of technologies included in the policy.

Constraint policies

When a `constraint` is read in along with a `tax` and `subsidy` policy, this sets up a constraint market. In years for which a value is set, the solver will find a constraint price that ensures the constraint is met.
The constraint can either be an upper bound (if the `tax` tag is used), a lower bound (if the `subsidy` tag is used), or an exact amount to match (see notes below on `min-price`). See below for more information.
If a `fixedTax` is read in before a `constraint` for the same market and model period, then the `fixedTax` values will be used as an initial guess for the constraint price.
`tax`: If `tax` policy is specified with a `constraint`, then the constraint values will upper bounds unless `min-price` is set (see below). Note that mechanically the `constraint` value is set as a "supply" and the model will adjust the "demand" (by changing the price) to ensure the constraint is met.
`subsidy`: If `subsidy` policy is specified with a `constraint`, then the constraint values will be lower bounds unless `min-price` is set (see below). Note that mechanically the `constraint` value is set as a "demand" and the model will adjust the "supply" (by changing the price) to ensure the constraint is met.
`RES`: Specifies a renewable energy standard, which is always a constraint. In this case, the value of the `constraint` is ignored. (While most often used for renewable energy, this can be applied more generally.)
`min-price`: Mechanically, this is the cut off price below which the model will consider an inequality constraint solved. If the price is below `min-price` and the supply (demand) is larger (smaller) than the constraint for a tax (subsidy), then the model will consider the market solved. The default is zero. If this is set to a large enough negative value (allowing negative prices as part of the solution process), constraints must be met exactly. That is, production must equal the amount specified in the constraint rather than the constraint being an upper bound (for a tax) or a lower bound (for a subsidy) on production. This can be used with either a `tax` or `subsidy` policy and would have the same result in either case.

Other Options

`market`: a means of grouping different regions together. This can be set to any string. All regions that have a common string will add to/demand from the same market.

Note that the string used for market has no relationship to actual region names, although region names are often used for convenience. (e.g., a user could specify that India and EU-12 are in a market called "Canada". which will have nothing to do with the GCAM region Canada.)

Setting up inputs for policies

In addition to setting up a policy market using the `policy-portfolio-standard` or `ghgpolicy` options described above, you also need to indicate what is included in the policy.
For implementing a `ghgpolicy`, the necessary emissions are usually already in place. So additional inputs may not be needed. For example, MAC curves already are set up to look at `market-name` (default CO2; be careful about units, see mac-price-conversion). Users can add additional tags to allow more specific constraints. For example, see the example below of a constraint on electricity CO_2 emissions.
For `policy-portfolio-standard` policies, one of the following tags below **must be used** in the relevant technology. As shown in the examples, these tags must be named with the corresponding policy name.

- `input-tax`: This should be used with technologies that are a part of a `tax` policy. Mechanically, this tag links the technology to the policy, adding the tax value to the cost of the technology. For constraints, the quantity of the input associated with this technology is added to the demand for the policy market.
- `input-subsidy`: This should be used with technologies that are a part of a `subsidy` policy. This tag links the technology to the policy, subtracting the subsidy value from the cost of the technology. For constraints, the quantity of input associated with this technology is added to the supply for the policy market.
- `res-secondary-output`: This is needed for use with the `RES` policy. The quantity of this output is added to the supply for the policy market.

Example Policies

Carbon Price

The following input file will create a carbon price of $1/tC (in $1990) in the USA, starting in the year 2020 and going through out the model time horizon. Additional regions can be added to this file (see the emissions constraint example below). Regions with the same `market` name will use the same carbon price.

```
<scenario>
    <world>
        <region name="USA">
            <ghgpolicy name="CO2">
                <market>USA</market>
                <fixedTax year="2020" fillout="1">1</fixedTax>
            </ghgpolicy>
        </region>
    </world>
</scenario>
```

Remember that the market identifier has no relationship to GCAM region names. In the above example, for example, the market identifier for the fixedTax market is "USA", which is used for convenience in the USA region. The model output, however, would be identical if `<market>Fred</market>` were used instead.
The name attribute of the `ghgpolicy` creates a market with that name. If there is no `linked-policy-market` object read in (see below) the name attribute must be the name of one or more emissions objects (*e.g.*, CO2). Otherwise, with `ghgpolicy` and `linked-policy-market` are used together, then the `ghgpolicy` object name is used for linking markets as described in the next section.

Note that GCAM internally expects carbon prices to be units of dollars per tonne C using $1990.

Another example of a global carbon tax can be seen in the GCAM release file carbon*tax*10_5.xml. In this file the carbon tax starts at $10/tC (in $1990) in 2025 and increases at 5% per year thereafter. Note that the carbon tax is specified in just one region, but because all regions are given the same `<market>global</market>` policy market identifer, the specified tax value will be used in all markets.

To implement a GHG policy for a single model run, a line should be added to the end of the `<ScenarioComponents>` section of the GCAM `configuration.xml` file pointing to file where the GHG policy is defined. Policy files can also be added to batch files.

Emissions Constraint

The following input file will set-up a constraint in the EU-15 region for a policy called "GHG". Total GHG emissions are constrained to be less than or equal to the `constraint` values.
A linked market would need to be defined with an additional file to link the different CO_2 emissions to the GHG market, which is also done in the GCAM release linked ghg policy file (see below). The units used in the `ghgpolicy` defined below will need to be consistent with the units defined in the `linked-ghg-policy`, as discussed in the next section.

```
<scenario>
   <world>
      <region name="EU-15">
         <ghgpolicy name="GHG">
            <market>EU-15</market>
            <constraint year="2020">867.3</constraint>
            <constraint year="2030">660.5</constraint>
            <constraint year="2050">415.8</constraint>
         </ghgpolicy>
      </region>
    </world>
</scenario>
```

For a cap on just one emission species, the form of the `ghgpolicy` is the same. The linked policy file described in the next section defines which emission species are included in a particular policy.

A constraint can also be applied across multiple regions. So if we defined a policy as follows:

```
<scenario>
   <world>
      <region name="EU-15">
         <ghgpolicy name="GHG">
            <market>EU-15plus</market>
            <constraint year="2020">900.0</constraint>
            <constraint year="2030">700.0</constraint>
            <constraint year="2050">430.0</constraint>
         </ghgpolicy>
      </region>
      <region name="Europe_Non_EU">
         <ghgpolicy name="GHG">
            <market>EU-15plus</market>
         </ghgpolicy>
      </region>
    </world>
```

```
</scenario>
```

then this constraint will apply across these two regions because each region has a GHG policy with the same name and market defined. The actual constraint values should only be specified once. A reminder that the "market" tag within the `ghgpolicy` is an arbitrary string that just needs to be unique in this context. Use something that helps describe your setup.

Linked Policies

Linked policies are used to tie the price of one policy to another. In the example below, taken from the GCAM release, CO_2 and CH_4 are linked to a policy called "GHG"; that is, the price of the GHG market will determine the price applied to both CO_2 and CH_4.

The policy referred to in the `<linked-policy>GHG</linked-policy>` tag must be previously defined by a policy. Therefore, the file defining the `linked-ghg-policy` must be read in *after* the policy it points to has been defined.
The `linked-policy` tag provides the name of the GHG policy that this links to.
The `market` tag is the identifier used to define which regions are part of this market. Regions using this identifier in their `linked-ghg-policy` objects will belong to the same market and share the same market prices. This identifier has no relationship with actual region names.
The `price-adjust` tag is the price multiplier used to convert units; the example below assumes that the GHG price is in 1990$/tC and converts that to CH_4 using GWPs. The `price-adjust` conversion for CO_2 is 1 since no conversion is needed if the GHG market is already in the unit needed by GCAM.
The `demand-adjust` is a multiplier used when adding emissions to a common market to be used in a constraint. In this example, GWPs are used to convert each gas to its CO_2-equivalent value, which is generally the unit used for a multi-gas GHG emissions target. For CO_2 the conversion is 3.667 tonnes CO_2 per tonne C, which converts from GCAM's output of CO_2 emissions in units of Mt C to Mt CO_2. Because GCAM's CO_2, CH_4, and N_2O emissions have units of Mt (= Tg), the GHG market defined in the `linked-ghg-policy` below, therefore, is defined for a policy that is defined in units of Mt CO2-Equivalant.

```
<scenario>
   <world>
      <region name="USA">
         <linked-ghg-policy name="CO2">
            <price-adjust fillout="1" year="1975">1</price-adjust>
            <demand-adjust fillout="1" year="1975">3.667</demand-adjust>
            <market>global</market>
            <linked-policy>GHG</linked-policy>
            <price-unit>1990$/tC</price-unit>
            <output-unit>MtC</output-unit>
         </linked-ghg-policy>
         <linked-ghg-policy name="CH4">
            <price-adjust fillout="1" year="1975">5.7272</price-adjust>
            <demand-adjust fillout="1" year="1975">21</demand-adjust>
            <market>global</market>
            <linked-policy>GHG</linked-policy>
            <price-unit>1990$/GgCH4</price-unit>
```

```
    ...
```

For a cap on just on one emission species, the `linked-ghg-policy` can contain information on that one species.
For a CO_2 only constraint, the user could set `demand-adjust` to 1, and define the `ghgpolicy` constraint in units of MtC. Alternatively, the `ghgpolicy` constraint could be defined in units of Mt CO_2, and `demand-adjust` could be set to 3.667 as in the example above.
Note that the combination of `price-adjust` and `demand-adjust` can be used to change regional market participation over time. If some regions should not be part of a market in early periods these values can both be set to zero for those regions and time periods and there will be zero price passed and that region will not contribute to quantity totals used in constraints (if specified).
Note that, in operation, each linked GHG policy object sets up a market that is used to implement the specified policy. The name of this market should be equal to the GHG or CO2 objects targeted by the policy. The default is that the name of this market will be set equal to the name of the `linked-ghg-policy` object. So in the example above, a market will be created for `CO2` and for `CH4`, respectively, with the attributes given in each of those sections of the xml. Emission objects of those same names will look, by default, to those markets for prices. (e.g., a `CH4` emission object will look a market called `CH4` for its price.)
For more complex policy setups, particularly where market characteristics vary over time, the name of the market can be set directly by the user by reading in a `policy-name` item within the `linked-ghg-policy`.
Note that `linked-ghg-policy` and `ghg-policy` objects should not be given the same name (Read-in will not operate properly in that case).

Sectoral Emissions Constraint

It is also possible to set up either a tax or emissions policy that applies to just specific technologies or sectors. In order to do this, it is necessary to add extra emissions to each relevant technology.

For example, to constrain CO_2 emissions from the electric power generation sector, we first add an additional CO_2 emissions to each relevant technology for each model year. Note that in this example we have added these to the `global-technology-database`. While this could be added to each relevant GCAM region, it is more convenient to add these to the global database if these are going to be used in multiple regions.

```
</scenario>
  <world>
    <global-technology-database>
       <location-info sector-name="base load generation" subsector-
name="coal">
            <technology name="coal_base_conv pul">
                <period year="1975">
                    <CO2 name="CO2_ELEC"/>
                </period>
                <period year="1990">
                    <CO2 name="CO2_ELEC"/>
                </period>
                <period year="2005">
                    <CO2 name="CO2_ELEC"/>
```

```
                </period>
                <period year="2010">
                    <CO2 name="CO2_ELEC"/>
                </period>
                <period year="2015">
                    <CO2 name="CO2_ELEC"/>
                </period>
                <period year="2020">
                    <CO2 name="CO2_ELEC"/>
                </period>
                <period year="2025">
                    <CO2 name="CO2_ELEC"/>
                </period>
                ... (and so on for all model years)
            </technology>
            ... (and so on for all relevant technologies)
          </location-info>
       </global-technology-database>
   </world>
</scenario>
```

Note that if you are setting up sectoral markets for an emission species that has emission control or MAC curves, those will need to be duplicated as well. In other words, any additional emissions that are added to facilitate a sectoral market should be identical to the existing emissions, but with an alternative name.

Now that we have added the appropriate emissions, we can now define the policy.

```
<scenario>
    <world>
        <region name="Brazil" nocreate="1">
            <ghgpolicy name="CO2_ELEC">
                <market>Brazil_Elec</market>
                <constraint year="2020">1.2345</constraint>
                <constraint year="2025">1.2345</constraint>
                <constraint year="2030">1.2345</constraint>
            </ghgpolicy>
        </region>
        <region name="Japan" nocreate="1">
            <ghgpolicy name="CO2_ELEC">
                <market>Japan_Elec</market>
                <constraint year="2020">0.123456</constraint>
                <constraint year="2025">0.123456</constraint>
                <constraint year="2030">0.123456</constraint>
            </ghgpolicy>
        </region>
```

```
    </world>
</scenario>
```

Because the new CO2_ELEC emission has been added to all electric technologies, we can set markets wherever we need. In the above example we have set a constraint for Brazil electricity CO_2 emissions and a separate constraint for Japan electricity CO_2 emissions. Separate markets just need to have unique names in this context. (e.g., a GHG market can have the same name as a region. GCAM will know that this does not refer to a GCAM region.)

The `<CO2 name="CO2_ELEC"/>` tag creates a CO_2 emissions called `CO2_ELEC`. This will generate CO_2 emissions that are identical to the default ('built in') CO_2. Because the name of this emission species is `CO2_ELEC` (and not the default CO_2, which will also already exist in each of these technologies), this will have no impact on the model. For example CO_2 concentrations. The additional `CO2_ELEC` emissions will only interact with whatever market mechanism we implement that uses these. Conversely, it is important that the names of these additional emissions are not set equal to any emission names already used in GCAM (*e.g.*, CH4, CH4_AWB, and CH4_AGR).

Note that additional emissions added in this way may also show up in some query results. For example database queries that look for emissions from "CO2" by object type would also return values from the "CO2_ELEC" emissions added above.

Energy Constraint

The following inputs will set a constraint on bioenergy use in the USA, limiting it to 10 EJ/yr. Note that the `input-tax` tag will need to be added to all model periods (the example only uses 2020 for brevity). Additionally, this tag needs to be added to all production technologies and regions that are included in the target. For example, corn ethanol, sugarcane ethanol, and biodiesel do not consume or produce "regional biomass" and therefore would be excluded from the policy below. If you wanted to include these options in this constraint, you would need to add a tag to those technologies in the refinery sector.
This example sets an upper bound on production. If instead you wanted a lower bound, then you would use `input-subsidy` in the technology and `policyType` equals "subsidy" in the policy-portfolio-standard. If you wanted to set an exact constraint, you can use either a tax or a subsidy with the additional tag `<min-price year="2020" fillout="1">-100</min-price>` which will allow the tax or subsidy to go negative, effectively enabling either a tax or a subsidy within the same constraint.
If you wanted to only constrain one type of bioenergy, then would only put the `input-tax` tag in the technology producing that type of bioenergy (e.g., you could include `input-tax` in the biomass resource to just limit MSW production).

```
<scenario>
   <world>
      <region name="USA">
         <supplysector name="regional biomass">
            <subsector name="regional biomass">
               <technology name="regional biomass">
                  <period year="2020">
                     <input-tax name="bio-constraint"/>
                  </period>
                  ...
```

```
                </technology>
            </subsector>
        </supplysector>
        <policy-portfolio-standard name="bio-constraint">
                <market>USA</market>
                <policyType>tax</policyType>
                <constraint year="2020" fillout="1">10</constraint>
        </policy-portfolio-standard>
    </region>
    </world>
</scenario>
```

Land Constraint

The following input file will keep UnmanagedForest area in the USA GreatLakes region above 12 thous sq km, starting in the year 2020 and going through out the model time horizon. Because the policy type is specified as `<policyType>subsidy</policyType>` the model will add a subsidy to the associated market to achieve the specified 12 thous sq km target.
If an upper bound was needed instead, this can be implemented by changing `policyType` to "tax" in the below xml example. In this case the opposite will occur, with the model adding a tax to the associated market to keep land use below the specified value.
Note that the `land-constraint-policy` tag will need to be added to any `UnmanagedLandLeaf` you want to constrain (e.g., land leafs use a different tag than technologies, which were discussed above). If more than one `UnmanagedLandLeaf` is given the same policy name ("reduced_deforestation" in this example) and market identifier (USA in this example), then the sum of those leafs will be constrained to the specified amount.

```
<scenario>
    <world>
        <region name="USA">
            <LandAllocatorRoot name="root">
                <LandNode name="AgroForestLand_GreatLakes">
                    <LandNode name="AgroForest_NonPasture_GreatLakes">
                        <LandNode name="AllForestLand_GreatLakes">
                            <UnmanagedLandLeaf
name="UnmanagedForest_GreatLakes">
                                <land-constraint-
policy>reduced_deforestation</land-constraint-policy>
                            </UnmanagedLandLeaf>
                        </LandNode>
                    </LandNode>
                </LandNode>
            </LandAllocatorRoot>
            <policy-portfolio-standard name="reduced_deforestation">
                <market>USA</market>
                <policyType>subsidy</policyType>
```

```
                <constraint year="2020" fillout="1">12</constraint>
            </policy-portfolio-standard>
        </region>
    </world>
</scenario>
```

Energy Intensity Standard

The following inputs will set up a energy intensity standard. These policies differ from the energy constraint described above in that they specify a share of a sectoral output. The example below sets a biofuels target, where the biofuels constraint is set by adding an additional input of "BioFuelsCredits" into the refined liquids for transportation sector. This credit input is read in as a `minicam-energy-input`, so its units are EJ with its price in $/GJ just like any other energy input, and its input coefficient is simply the target percentage. This sets the demand for "BioFuelsCredits". (In this example, an additional pass-through sector called "refined liquids transport" was created to apply the constraint just on transportation rather than on all refined liquids. "refined liquids transport" is then used as the input to the transportation technologies.)

```
   <supplysector name="refined liquids transport">
            <subsector name="refined liquids transport">
               <technology name="refined liquids transport">
                  <period year="2020">
                     <minicam-energy-input name="BioFuelsCredits">
                        <coefficient>0.1</coefficient>
                     </minicam-energy-input>
                     <share-weight>1</share-weight>
                     <minicam-energy-input name="refining">
                        <coefficient>1</coefficient>
                     </minicam-energy-input>
                  </period>
```

The supply is created by putting a secondary output of "Biofuels Credits" on each of the biofuels liquids technologies, with an `output-ratio` of 1. The `output-ratio` specifies the quantity of "BioFuelsCredits" supplied for each GJ of biofuels produced. With a positive price of Biofuels Credits, the secondary output will have a value that reduces the biofuels technology costs and increases their market share. If the constraint is not binding (there are more biofuels than required), the price of "BioFuelsCredits" will be zero and have no impact on market shares.

```
<supplysector name="refining">
            <subsector name="biomass liquids">
               <technology name="cellulosic ethanol">
                  <period year="2020">
                     <res-secondary-output name="BioFuelsCredits">
                        <output-ratio>1</output-ratio>
                     </res-secondary-output>
                  </period>
```

The "BioFuelsMarket" and policy are created by reading in the policy as an RES policy.

```
  <policy-portfolio-standard name="BioFuelsCredits">
            <market>USA</market>
            <policyType>RES</policyType>
```