QubecTalk Domain Specific Language

A Samuel Pottinger

2025-10-30

Contents

Summary: Modelers may engage with a domain specific language called QubecTalk in order to construct analyses relevant to the Montreal Protocol and Kigali Amendment. Specifically, this human-readable text-based format can describe scenarios to a simulation engine called Kigali Sim. Written in JavaScript with an ANTLR4 parser, the interpreter for this DSL can execute in browser via WebAssembly or locally on a machine using Java (Temurin 21+). It may run as part of a web application with both a graphical user interface (Designer tab) and code editor (Advanced Editor tab). However, it may also be invoked directly through a command line JAR interface or Docker. Additionally, these programs can be written "manually" within a code editor or through a graphical user interface (which does not require authors to engage code directly). This document describes QubecTalk version 2.0 and offers realistic full featured examples of simulations written in the language.

1 Purpose

This document details a domain specific language called QubecTalk. This "DSL" is part of a larger system called Kigali Sim which allows for the construction of mathematical simulations relevant to the Montreal Protocol with a focus on hydrofluorocarbons (HFCs) and their alternatives within the Kigali Amendment. These longitudinal[1] projections simulate applications, substances, and possible interventions. In this process, this system seeks to support a broad ecosystem of actors across a longer journey of evaluating and comparing options through analysis outputs. The tool is currently in use by multiple Article 5 nations and was co-created in consultation with over a dozen countries and supporting organizations.

[1] Over time projections which simulate relevant metrics across a discrete set of years as specified by model authors.

1.1 Audience

Authors[2] of these models range from those seeking a cursory or exploratory projection to experts crafting highly specific or detailed scenarios. To support this wide range of uses, we offer a domain specific language (DSL) called QubecTalk. Analyses which may include multiple simulations are described in single file programs which are run through an interpreter. This "engine" is written in JavaScript / ECMAScript with an ANTLR4 parser and compiled to both WebAssembly (for browser execution) and Java bytecode (for standalone JAR execution). Authors can interact with this technology through different modalities:

- **Graphical interface (Designer)**: Some at the start of a project may simply author simulations through a graphical user interface in the Designer tab, building out projections and simple policy scenarios without interacting with any code directly.
- **Code editor (Advanced Editor)**: Others may write code in the Advanced Editor tab to tailor sophisticated simulations to very specific scenarios. These authors may also take advantage of algorithms like Monte Carlo to express uncertainties or explore many potential futures.
- **Command line**: Technical users may prefer to run simulations via the standalone JAR file or Docker container, integrating Kigali Sim into automated workflows or scripts.

Of course, analyses may migrate from quick sketch without editing code to intricate projections engaging the source extensively as ideas are refined and explored. This may also take place socially: one author may start a rough "sketch" of an analysis in a graphical interface later refined by another collaborator working in a code editor. Regardless, this document describes QubecTalk and offers examples of realistic models translated into this format.

1.2 Objectives

QubecTalk serves multiple roles in this context:

- **Internal representation**: In the case that source code is generated and manipulated by a graphical interface to support authors not wishing to program in code, scripts become internal representations of inputs. This allows for saving and loading of configuration in a structured purpose-built format.
- **Expressiveness**: In the case that scripts are further refined by modifying the source code in a more traditional programming context, these scripts may be engaged in tools like full featured editors or version control. This format may be more expressive (ergonomic and flexible) for expert authors seeking to create very sophisticated models.

[2]In alignment with prior community perspectives, we refer to our audience as authors regardless of skill level. This nomenclature is used in place of "users" as a term. This recognizes and values their agency and their centrality in co-creation within this system.

- **Standardization**: Regardless of how scripts are constructed, these programs offer a unified and structured interface to the engine. This also affords some reusability of this system, allowing for the adoption of other interfaces (graphical or otherwise) in the future.
- **Approachability**: Some authors may be less likely to experience this system from a blank text file and, instead, will edit code started by the graphical user interface. Therefore, this DSL attempts to optimize for readability by audiences which may be less familiar with programming.

This document first outlines a language definition and then offers example implementations which demonstrate the full feature set.

1.3 License

2 Definition

This document first provides an outline of language features.

- We start with discussion of the structure of a QubecTalk program.
- Next, we transition to discussion of system definition including applications and substances.
- Then, we consider policy interventions and their mechanisms.
- Finally, we discuss simulation definition and advanced language features.

A program may contain multiple simulations where each includes zero or more policies on top of a "business as usual" simulation state. For more information, see the examples.

2.1 Structure

Generally there is a definition of a system under a business as usual scenario followed by policies and simulation tasks. This DSL operates through a "stanza" structure:

```
start about
  # Name: "Simulation Name"
  # Description: "Simulation Description"
  # Author: "Simulation Author"
  # Date: "Simulation Date"
end about

start default
  # Business as usual scenario definition
end default
```

```
start policy "Policy 1"
  # Policy interventions
end policy

start policy "Policy 2"
  # Policy interventions
end policy

start simulations
  # Simulation scenarios
end simulations
```

Each section provides a type and, if appropriate, label for a set of commands. The `about` stanza is optional and provides metadata about the simulation.

2.2 Comments

Comments in QubecTalk use the hash symbol (`#`) and run to the end of the line:

```
# This is a full line comment
set domestic to 25 mt during year 1  # This is an end-of-line comment
```

2.3 System

The components of the system interact with each other under policy modification to determine substance consumption, GHG impacts, equipment populations, and energy consumption.

2.3.1 Applications

Analysis may consider substances across various different applications. Some examples:

- **Refrigeration:** Domestic, small commercial, large commercial, industrial, transport.
- **Air conditioning**: Residential, commercial, central / large, mobile.
- **Medical aerosols**: MDIs, etc.

QubecTalk defines these as follows:

```
define application "domestic refrigeration"

  # Optional variables across substances within an application.

  uses substance "HFC-134a"
    # Logic for a substance
  end substance
```

```
end application
```

Note that the same substances may be included across multiple applications. Furthermore, these may be imported or built domestically or exported. When defining these substances, an initial charge should be provided for streams that will have sales:

```
initial charge with 0.12 kg / unit for domestic
initial charge with 0.30 kg / unit for import
initial charge with 0.10 kg / unit for export
```

If this is omitted and the substance has sales, an error message may be shown. The initial charge determines how much substance is contained in each unit of new equipment for that stream.

2.3.2 Sales

The sale of substances typically defines the equipment population. QubecTalk considers sales to fit into three groups: `domestic`, `import`, and `export`.

Enable Statement: Before setting sales values, streams must be explicitly enabled using the `enable` command. This command indicates which streams (domestic, import, export) will be used for a substance:

```
enable domestic
enable import
enable export
```

By convention, enable statements should be placed at the top of substance definitions, though this is not enforced by the language. Note that streams can also be implicitly enabled by setting non-zero values for them, but explicit `enable` statements are recommended for clarity and are required for certain operations like `recharge` that need to know the sales distribution.

The `enable` command can be applied to other stream names, but it will have no effect. Only `domestic`, `import`, and `export` streams can be enabled.

Domestic: Starting with an example of specifying substances both produced and consumed domestically. There are typically three important parameters to specify:

- Manufacturing (volume)
- Sales growth (YoY %)
- Initial charge (kg per unit)

This can be established through the following example statements:

```
set domestic to 350000 kg during year 1
change domestic by +5 % each year during years 1 to 6
change sales by +3 % each year during years 6 to onwards
```

Note that, if not specifying a year, a statement will apply across the entire timeseries. If no stream is specified, a statement will apply proportionally across sub-streams. For example, in the case of sales, changes apply proportionally across domestic manufacturing and imports.

Trade: Imported and exported equipment follows a similar pattern with typical parameters:

- Imported sales (volume)
- Export sales (volume)
- Sales growth (YoY %)
- Initial charge (kg per unit)

These are specified using the same statements but indication that the sales flow through the imports or exports channel.

```
set import to 90000 kg during year 1
change import by +5 % each year
change export by +3 % each year during years 6 to 9
```

If no stream is specified, it will apply proportionally across sub-streams (domestic manufacturing, imports, and exports).

Equipment Units and Implicit Recharge: When specifying sales using equipment units (e.g., `set domestic to 100 units`), the system automatically calculates and includes implicit recharge. This means the engine assumes sufficient substance is available to service (recharge) existing equipment in addition to the substance needed for initial charges on new equipment. This behavior applies because users specifying equipment units are expressing intent about how much new equipment is to be sold, and the simulation ensures servicing needs are met automatically.

2.3.3 Population

Historic equipment populations are typically inferred by sales before applying a retirement rate into the future. However, they may be specified manually as well.

Initializing equipment: Equipment levels at any year may be specified manually and this approach can be used to initialize the equipment population:

```
set equipment to 2000 units during year 1
set bank to 2000 units during year 1  # bank is syntactic sugar for equipment
```

This sets the current year equipment which assumes the prior year's levels stay the same. To specify a new level for equipment not sold in the current year, use `priorEquipment`:

```
set priorEquipment to 8 * sales during year 1
set priorEquipment to (get equipment as units * 7) units during year 1
```

These approaches can be mixed but, generally, authors will apply a multiplier against a specific (typically first) year of sales or equipment.

Adding equipment: Equipment is added based on sales. For most authors, this is preferred. However, to override this behavior, use `set equipment` to override the value.

Removing equipment: Most analyses will remove equipment through retirement (also called hazard rate or scrap rate). Consider these examples:

```
retire 6.7 %
retire 5 % during years 1 to 5
retire 6.7 % each year
```

If a year is not specified, it will be apply across the entire timeseries. In addition to accepting percentages, an absolute number of units may be given. Note that authors may also choose to use `set equipment` as seen in adding equipment.

To maintain a constant equipment population, use the `with replacement` modifier:

```
retire 6.7 % with replacement
```

This causes retired equipment to be replaced with new equipment, maintaining population size while still consuming substance for new charges.

Service schedule: The language allows for specifying annual recharge quantity from equipment (kg or mt) which is typically applied to an annual percentage of total equipment.

```
recharge 10 % with 0.12 kg / unit
recharge 5 % with 0.12 kg / unit during years 1 to 5
recharge 10 % each year with 0.12 kg / unit
```

One may also recharge a number of units by changing `%` to `units`.

2.3.4 Consumption

The `equals` command can be used to specify two types of intensities:

1. GHG intensity to calculate greenhouse gas emissions in tonnes of CO2 equivalent (tCO2e):

```
equals 1430 tCO2e / mt
equals 1430 kgCO2e / kg  # Alternative unit format
```

2. Energy intensity to calculate energy consumption in kilowatt hours (kwh):

```
equals 100 kwh / unit
equals 100 kwh / kg
```

Both can be specified in the same statement:

```
equals 1430 kgCO2e / kg 100 kwh / unit
```

Note that `tCO2e` refers to tonnes of CO2 equivalent, `kgCO2e` refers to kilograms of CO2 equivalent, and `kwh` refers to kilowatt hours. Both metrics help track different aspects of environmental impact.

2.3.5 Bank Tracking

The system tracks the "bank" of substance - the total amount of refrigerant contained within the equipment population currently in service. This is calculated using the initial charge values and equipment population.

The `bank` keyword is syntactic sugar for the `equipment` stream and can be used interchangeably with `equipment` for semantic clarity:

```
set bank to 1000 units during year 1
assume only recharge bank during years 2025 to onwards
```

The bank is reported in simulation outputs as both substance volume (kg) and GHG potential (tCO2e).

2.4 Policies

Policies can be defined by name and typically make changes to the business as usual created within the `default` stanza.

```
start policy "Policy Name"
  modify application "application name"
    modify substance "substance name"
      # Policy interventions
    end substance
  end application
end policy
```

Policy names must be unique. Note the use of `modify` instead of `define` - policies modify existing applications and substances defined in the default scenario.

2.4.1 Permit / prohibition

Many policies work by placing a limit on sales using `cap`:

```
cap sales to 75 %
cap import to 750 mt during years 4 to 5
cap domestic to 0 mt during years 10 to onwards
```

Sales, domestic, import, export, and equipment streams can accept the cap. If specifying an aggregate stream, it will be applied proportionally. For example, for sales, the effect will be split proportionally between domestic and imports.

When using units (equipment) in cap commands, those limits apply after recharge calculations. So a cap of 100 units means 100 units of new equipment on top of recharge needs for prior equipment.

2.4.2 Floor

A minimum level can be set using `floor`:

```
floor sales to 10 %
floor import to 100 mt during years 5 to onwards
```

This prevents unrealistic reductions below specified levels.

2.4.3 Displacement

Caps and floors can specify displacement to alternative substances or streams:

```
cap sales to 80 % displacing "R-600a" during years 3 to 5
cap domestic to 0 mt displacing "low-GWP alternative" during years 10 to onwards
floor import to 50 mt displacing domestic during years 2 to 8
```

This causes the reduction in one substance to be offset by an increase in another.

2.4.4 Recycling

Recycling programs are defined with recovery percentage, reuse yield rate, and optional induction behavior:

```
recover 5 % with 100 % reuse during year 3
recover 10 % with 100 % reuse during years 4 to onwards
recover 20 % with 80 % reuse with 50 % induction during years 5 to onwards
```

Recycling Stages: Recycling can occur at different lifecycle stages:

```
recover 20 % with 80 % reuse at recharge during years 3 to onwards
recover 30 % with 90 % reuse at eol during years 5 to onwards
```

- `at recharge` - Recycling occurs during equipment servicing (default if not specified)
- `at eol` - Recycling occurs during equipment retirement (end-of-life)

Induction Control: Induction determines how much recycled material adds to the market versus displaces virgin production:

```
recover 20 % with 80 % reuse with 0 % induction    # Full displacement
recover 20 % with 80 % reuse with 100 % induction  # Full induced demand
recover 20 % with 80 % reuse with 50 % induction   # Mixed behavior
recover 20 % with 80 % reuse with default induction # Use system default
```

- **0% Induction (Displacement)**: Recycled material reduces virgin material demand. Total equipment population stays constant.
- **100% Induction (Induced Demand)**: Recycled material adds to total supply. Equipment population increases due to additional material availability.
- **Partial Induction (Mixed)**: Combines both behaviors proportionally.
- **Default Induction**: Uses default behavior based on specification type:

- Units-based specs: Default 0% induction (displacement behavior)
- Non-units specs (kg/mt): Default 100% induction (induced demand behavior)

100% induction is recommended for users who are uncertain as induction is a complex economic phenomenon. It can be paired with other policies like caps on virgin material to ensure desired reductions.

The `displacing` clause can also be used to specify which substance or stream is offset:

```
recover 10 % with 100 % reuse displacing "HFC-134a" during years 4 to onwards
```

2.4.5 Replace

Transition by incentive or other similar mechanisms can be expressed through a `replace` command:

```
replace 5 % of sales with "low gwp"
replace 100 units of sales with "low gwp" during years 3 to onwards
replace 10 mt of domestic with "low gwp" during years 3 to onwards
```

This will change the equipment population where percentages are assumed to be percentage of equipment or sales. If providing mt or kg, this will convert to equipment units. Furthermore, this will displace from manufacturing and imports proportionally if a sales stream is not specified.

2.4.6 Sales Assumptions

The `assume` command provides control over sales carryover behavior, particularly useful for bank tracking:

```
assume no domestic during year 2030              # Sets domestic to 0 kg
assume only recharge sales during years 2025 to onwards  # Sales only cover recharge
assume continued import                           # Continue from previous year (default)
assume no import during years 5 to 10
assume only recharge bank during years 3 to onwards
```

Supported streams: `domestic`, `import`, `export`, `sales`, `bank`, `equipment`

The `assume only recharge` variant is particularly useful when modeling that new equipment sales should cease but servicing continues.

2.5 Simulation

A set of zero or more policies defines a scenario.

```
simulate "Business as Usual" from years 1 to 20

simulate "Standalone Import Ban"
  using "Import Ban"
```

```
from years 1 to 20

simulate "Combined"
  using "Import Ban"
  then "Domestic Permit"
  then "Efficiency Incentive"
from years 1 to 20

simulate "Combined with Uncertainty"
  using "Import Ban"
  then "Domestic Permit"
  then "Efficiency Incentive"
from years 1 to 20 across 1000 trials
```

Each scenario will project out a number of years and report consumption of HFCs and alternatives in CO2 tonnes, equipment populations, energy consumption, and other metrics. In order to report changes to business as usual, do not provide any policies.

Monte Carlo simulations can be run by specifying `across N trials` where N is the number of simulation runs with different random samples.

Time specifications support keywords: - `beginning` - first year of simulation - `onwards` - last year and all subsequent years - Absolute years like `2025` or `2030` - Relative years like `1` or `20`

2.6 Language Features

A few places where other language features may be helpful to more expert users.

2.6.1 Variables

Full arithmetic expressions with local variables:

```
define x as 5 + 6 * 7 ^ 8
define baselineYear as 2025
define phaseOutRate as 20
```

Variables can be used in any expression:

```
set domestic to (100 * phaseOutRate / 100) mt during year baselineYear
```

These are limited to the scope of the enclosing `start` and `end` pair. There is also an optional global `variables` stanza:

```
start variables
  define globalVar as 100
  define targetYear as 2030
end variables
```

2.6.2 Conditional

Conditionals are expressed as ternary operators:

```
set domestic to 100 if testVar > 10 else 50 endif mt
define salesLevel as 200 if yearAbsolute >= 2030 else 150 endif
```

The operators supported: `<`, `>`, `<=`, `>=`, `==`, and `!=` for less than, greater than, less than or equal, greater than or equal, equals, and not equals. Boolean expressions may be combined through logical operators (`and`, `or`, `xor`). True is returned as 1 and false as 0.

```
set domestic to 90 if (testA or testB) and testC else 55 endif mt
```

2.6.3 Logical Operators

Boolean expressions can be combined:

- `and` - Both conditions must be true
- `or` - At least one condition must be true
- `xor` - Exactly one condition must be true (exclusive or)

```
define testA as 1
define testB as 0
set domestic to 100 if testA and testB else 30 endif mt
set import to 50 if testA or testB else 20 endif mt
set export to 60 if testA xor testB else 40 endif mt
```

2.6.4 Constraints

The `limit` function constrains values to a specified range:

```
limit expression to [min, max]
limit expression to [, max]    # min of negative infinity
limit expression to [min, ]    # max of infinity
```

Example:

```
define level as limit (yearAbsolute - 2025) * 10 to [0, 100]
replace level % of domestic with "R-600a" during years 2026 to onwards
```

2.6.5 Uncertainties

Typically uncertainties are applied on demand such as sales growth rates, retirement, and recharge. In general, sampling can be applied in any expression:

```
sample uniformly from 5 to 10
sample normally from mean of 5 std of 2
```

For example, one may specify a distribution of possible values for retirement rate:

```
retire sample normally from mean of 6.7 std of 1 %
change sales by sample uniformly from 3 to 7 % each year
```

When building these simulations, users should also include a number of trials (`across 1000 trials`). Different random values will be drawn for each trial.

2.6.6 Timeseries

The current year can be found in a variable called `yearsElapsed` and `yearAbsolute` for years since the start of the simulation and absolute year number respectively. These variables cannot be set or redefined.

```
define scalingFactor as yearAbsolute * 5 + 2
set domestic to (100 * yearsElapsed / 10) mt
```

These values can be used in any expression. Similar variables are also available for other timescales: `monthsElapsed`, `monthAbsolute`, `daysElapsed`, and `dayAbsolute`. These do not reset from year to year but accumulate from the start of the simulation. Note that month and day timescales are reserved for future use.

2.6.7 Get and set

Developers can manually set a value of a stream:

```
set import to 1 mt during year 2
set sales of "HFC-134a" to 1 mt
set domestic of "HFC-134a" to 1 mt during year 2
set priorEquipment to 1000 units during year 1
```

If year is not specified, it is applied in all years. Values can also be retrieved in expressions:

```
define currentSales as get sales
define currentSalesKg as get sales as kg
define salesOther as get sales of "HFC-134a" as kg
define equipmentCount as get equipment as units
```

Setting an aggregate stream like sales will cause the value to be distributed proportionally to the prior value of the sub-streams. In the case of this example, this would be applied across import, domestic, and export.

2.6.8 Age stream

The `age` stream provides access to the weighted average age of equipment in the population. This is a read-only computed stream:

```
define currentAge as get age as years
retire (get age as years) * 1 %  # Age-dependent retirement
```

Key characteristics: - Age starts at 0 years for initial equipment - Age increases by 1 year each simulation year - With new equipment, age is the weighted average based on equipment quantities - Read-only: cannot be set by users - Usage: `get age as years` in formulas

3 Units

Various statements can specify units. These are not assignable to variables so are only included in supporting commands.

3.1 Supported units

Available units include the following:

- `kg` or `mt`: Volumes as kilograms or megatons (metric tons)
- `tCO2e` or `kgCO2e`: Consumption as tons or kilograms of CO2e
- `unit` or `units`: Population size as equipment units
- `year` or `years` or `yr` or `yrs`: Time as year or years elapsed from start of simulation
- `kwh`: Energy consumption in kilowatt hours
- `%`: Percentage

The following are not currently supported but reserved for future use:

- `day` or `days`: Time as day or days elapsed from start of simulation
- `month` or `months`: Time as month or months elapsed from start of simulation

These may be expressed as a ratio like `tCO2e / mt` or `kg / unit`. In the case that the division is by a time unit like `year`, this is assumed to be a compounding value per time unit where the first day / month / year is zero.

3.2 Number Formatting

Kigali Sim uses DecimalFormat number formatting (comma for thousands, period for decimal):

```
set domestic to 1,000,000 kg   # Valid: 1 million kg
set import to 25,500.75 mt     # Valid: 25,500.75 metric tons
```

Alternative formats like `123.456,7` will generate suggestions for the equivalent supported format `123,456.7`.

3.3 Behavior

Interpretation and unit conversion depends on the command. The engine automatically converts between compatible units and propagates changes through dependent streams.

Op	Type	Units	Percent	Ratios
Initial Charge	Volume	Volume distributed across population. Consumption converted to volume.	Not supported.	Div by units will multiply by pop size.
Set Sales	Volume	Volume. Consumption converted to volume, units converted to volume. Proportional if not sub-stream.	$x = x * \%$	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by consumption multiplies consumption.
Set Equipment	Population	Units. Consumption converted to volume, volume converted to units.	$x = x * \%$	Div by time multiplies `yearsElapsed`, div by volume multiplies volume, div by consumption multiplies consumption.
Set Consumption	Consumption	Consumption distributed across volume streams.	$x = x * \%$	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by volume converts to div by units.
Change Sales	Volume	Volume delta. Consumption converted to volume, units converted to volume. Proportional if not sub-stream.	$x = x*(1+\%)$	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by consumption multiplies consumption.

Op	Type	Units	Percent	Ratios
Change Equipment	Population	Units delta. Consumption converted to volume, volume converted to units. Proportional.	$x = x*(1+\%)$	Div by time multiplies `yearsElapsed`, div by volume multiplies volume, div by consumption multiplies consumption.
Change Consumption	Consumption	Consumption delta distributed across volume.	$x = x*(1+\%)$	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by volume converts to div by units.
Retire	Population	Consumption converted to volume, volume converted to units.	$r = x * \%$	Div by time multiplies `yearsElapsed`, div by volume multiplies volume, div by consumption multiplies consumption.
Cap Sales	Volume	Volume. Consumption converted to volume, units converted to volume. Proportional if not sub-stream.	$c = x * \%$	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by consumption multiplies consumption.
Cap Equipment	Population	Units. Consumption converted to volume, volume converted to units.	$c = x * \%$	Div by time multiplies `yearsElapsed`, div by volume multiplies volume, div by consumption multiplies consumption.

Op	Type	Units	Percent	Ratios
Cap Consumption	Consumption	Consumption total distributed across volume streams.	$c = x * \%$	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by volume converts to div by units.
Floor Sales	Volume	Volume. Consumption converted to volume, units converted to volume. Proportional if not sub-stream.	$f = x * \%$	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by consumption multiplies consumption.
Floor Equipment	Population	Units. Consumption converted to volume, volume converted to units.	$f = x * \%$	Div by time multiplies `yearsElapsed`, div by volume multiplies volume, div by consumption multiplies consumption.
Recycle	Volume	Volume. Consumption converted to volume, units converted to volume. Proportional.	Uniform across streams.	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by consumption converts to div by units.
Replace Sales	Volume	Volume. Consumption converted to volume, units converted to volume. Proportional if not sub-stream.	Uniform across streams.	Div by time multiplies `yearsElapsed`, div by units multiplies population, div by consumption converts to div by units.

Op	Type	Units	Percent	Ratios
Replace Equipment	Volume	Converted to volume.	Uniform across streams.	Div by time multiplies `yearsElapsed`, div by volume converts to units, div by consumption converts to div by volume.
Replace Consumption	Volume	Converted to volume.	Uniform across streams.	Div by time multiplies `yearsElapsed`, div by volume converts to units, div by units converts to volume.
Get Sales	Volume	Volume (`kg`)	N/A	N/A
Get Equipment	Population	Population (`units`)	N/A	N/A
Get Consumption	Consumption	Consumption (`tCO2e`)	N/A	N/A

In addition to the strategies described above, conversion may invert ratio units to achieve a valid configuration. Otherwise, if behavior is not specified above, the correct behavior is undefined. In these cases, units are "unsupported" for a command and will result in an error at runtime. Note that units are converted to consumption and volumes with amortized values.

3.4 Proportionality with units

Finally, "proportional" operations on percentages will apply the same to all sub-streams if using a percentage or unit which includes division (ex: `kg / mt`) otherwise the effect size will be proportional to the size of the sub-stream prior to the operation.

4 Engine

While the system can operate with constraints through `cap` and `floor`, updates are applied in the order specified within code. Within the context of streams, one stream can cause another to update with these changes propagating when a stream is changed. The following dependencies see changes made at the start of the chain cause changes downstream. Note that these are not transitive beyond this table so changes to sales does not cause a consumption trigger.

Trigger	Sales	Consumption	Population
Sales (domestic, import, export)	Changed directly	Recalc using equals value	Recalc new and total equip after subtract recharge and recycle, retiring at recharge if needed.
Consumption	Recalc after add recharge and subtract recycle.	Changed directly	Recalc new and total equip after subtract recharge and recycle, retiring at recharge if needed.
Population (equipment)	Recalc after add recharge and subtract recycle.	Recalc after sales update using equals	New equip and total current equip changes but prior equip untouched.
Population (priorEquipment)	No change	No change	Change prior equip. New and total equipment recalculated.

These updates happen automatically within the interpreter and all use the following which are recorded internally with original units.

4.1 Saved parameterization

For engine internal updates, some parameters are saved for each substance. Any compatible units are acceptable though examples are included for reference.

Parameter	Example units	Set by
GHG intensity	tCO2e / kg	`equals`
Energy intensity	kwh / unit	`equals`
Recharge population	%	`recharge`
Recharge intensity	kg / unit	`recharge`
Recovery rate	%	`recover`
Yield rate on recycling	%	`recover`
Induction rate	%	`recover`
Retirement rate	%	`retire`

If multiple statements write to these parameters, the latest value is used when propagating.

4.2 Responses

In calculating downstream effects, some streams may be recalculated based on an author-specified change in a triggering stream.

4.2.1 Population response

Propagating effects to the equipment population can happen either by changing the new equipment deployment in a year or by changing the prior population size. However, as changing the prior population simply causes the change in population to be recalculated, both operations result in the same outcome:

- Determine retirement from prior population.
- Remove recharge from total available.
- Add recycling to sales to get total available substance using induction rate and yield rate.
- Convert remaining to added units.
- Determine final delta as added units minus retirement.

Note that retirement applies to prior population size. While the new population delta may be negative, set negative total populations as zero (interpret as net exports).

4.2.2 Consumption response

Recalculating consumption largely relies on manufacturing numbers and applies to the substance regardless of where the substance gets used.

- Remove imports from sales.
- Offset manufacturing with recycling (assumed proportional to import / domestic / export share) using induction rate and yield rate.
- Convert to consumption.
- Set negative total consumption to zero (treat net negative consumption as credit taken elsewhere).

Note that consumption is applied to point of manufacture (exporter not importer) under standard Montreal Protocol conventions.

4.3 Sales response

Sales updating to changes in other variables scales imports, manufacturing, and exports proportionally to meet the requirements of recharge and new equipment.

- Determine needs for recharge.
- Determine needs for new equipment deployment.
- Subtract recycling with consideration of induction.
- Update import, domestic, and export sales proportionally.

Note that negative sales are assumed as exports.

4.4 Outputs

The simulation produces detailed CSV outputs with the following information for each scenario, trial, year, application, and substance:

Production & Trade Volumes (kg/year): - `domestic` - Domestic manufacturing volume - `import` - Import volume (bulk substance) - `export` - Export volume (bulk substance) - `recycle` - Recycled substance volume

Consumption Impact (tCO2e/year): - `domesticConsumption` - GHG impact from domestic manufacturing - `importConsumption` - GHG impact from imports - `exportConsumption` - GHG impact from exports - `recycleConsumption` - GHG impact from recycling

Equipment Population (units): - `population` - Total equipment population in service - `populationNew` - New equipment added in current year

Emissions Sources (tCO2e/year): - `rechargeEmissions` - Emissions from equipment servicing/leakage - `eolEmissions` - Emissions from end-of-life equipment disposal - `initialChargeEmissions` - Informational metric for GHG potential of new charges

Energy Impact: - `energyConsumption` - Energy consumption from equipment operation (kwh/year)

Trade Attribution Supplement: - `importInitialChargeValue` - Substance volume in imported equipment initial charges (kg) - `importInitialChargeConsumption` - GHG impact from imported equipment charges (tCO2e) - `importPopulation` - Equipment units imported with initial charges - `exportInitialChargeValue` - Substance volume in exported equipment initial charges (kg) - `exportInitialChargeConsumption` - GHG impact from exported equipment charges (tCO2e)

Bank Tracking (substance in equipment): - `bankKg` - Total substance volume contained in all equipment in service (kg) - `bankTCO2e` - Total GHG potential of substance in all equipment in service (tCO2e) - `bankChangeKg` - Change in substance bank from previous year (kg) - `bankChangeTCO2e` - Change in GHG potential of substance bank from previous year (tCO2e)

Under standard Montreal Protocol reporting, initial charge for imported equipment is typically attributed to the exporting country (equipment manufacturer), while recharge/servicing is attributed to the importing country (equipment operator). The trade supplement columns enable user-configurable attribution.

5 Example

This example translates a reference model into QubecTalk using the language defined above.

```
start about
  # Name: "Multi-Application HFC Analysis"
  # Description: "Comprehensive refrigeration and AC analysis with policy scenarios"
  # Author: "A Samuel Pottinger"
  # Date: "2025-01-30"
end about
```

```
start default

  define application "domestic refrigeration"

    uses substance "HFC-134a"
      enable domestic
      enable import
      equals 1430 kgCO2e / kg
      equals 100 kwh / unit

      # Domestic production
      initial charge with 0.12 kg / unit for domestic
      set domestic to 350000 kg during year 1
      change domestic by +5 % each year during years 1 to 5
      change domestic by +3 % each year during years 6 to 9

      # Trade
      initial charge with 0.30 kg / unit for import
      set import to 90000 kg during year 1
      change import by +5 % each year during years 1 to 5
      change import by +3 % each year during years 6 to 9

      # Service
      retire 6.7 % each year
      recharge 10 % with 0.12 kg / unit

      # Historic units already deployed
      set priorEquipment to (get equipment as units * 7) units during year 1
    end substance

    uses substance "R-600a"
      enable domestic
      enable import
      equals 6 kgCO2e / kg
      equals 80 kwh / unit

      # Domestic production
      initial charge with 0.05 kg / unit for domestic
      set domestic to 200000 kg during year 1
      change domestic by +8 % each year

      # Trade
      initial charge with 0.05 kg / unit for import
      set import to 10000 kg during year 1
```

```
    change import by +8 % each year

    # Service
    retire 6.7 % each year
    recharge 10 % each year with 0.05 kg / unit

    # Historic units already deployed
    set priorEquipment to (get equipment as units * 7) units during year 1
  end substance

end application

define application "commercial refrigeration"

  uses substance "HFC-134a"
    enable domestic
    enable import
    equals 1430 kgCO2e / kg
    equals 100 kwh / unit

    # Domestic production
    initial charge with 0.30 kg / unit for domestic
    set domestic to 90000 kg during year 1
    change domestic by +5 % each year during years 1 to 5
    change domestic by +3 % each year during years 6 to 9

    # Trade
    initial charge with 0.30 kg / unit for import
    set import to 90000 kg during year 1
    change import by +5 % each year during years 1 to 5
    change import by +3 % each year during years 6 to 9

    # Service
    retire 6.7 % each year
    recharge 10 % each year with 0.30 kg / unit

    # Historic units already deployed
    set priorEquipment to (get equipment as units * 7) units during year 1
  end substance

  uses substance "R-600a"
    enable domestic
    enable import
    equals 6 kgCO2e / kg
    equals 80 kwh / unit
```

```
    # Domestic production
    initial charge with 0.12 kg / unit for domestic
    set domestic to 10000 kg during year 1
    change domestic by +8 % each year

    # Trade
    initial charge with 0.12 kg / unit for import
    set import to 10000 kg during year 1
    change import by +8 % each year

    # Service
    retire 6.7 % each year
    recharge 10 % each year with 0.12 kg / unit

    # Historic units already deployed
    set priorEquipment to (get equipment as units * 7) units during year 1
  end substance

  uses substance "R-404A"
    enable domestic
    enable import
    equals 3922 kgCO2e / kg
    equals 100 kwh / unit

    # Domestic production
    initial charge with 0.30 kg / unit for domestic
    set domestic to 30000 kg during year 1
    change domestic by +5 % each year

    # Trade
    initial charge with 0.30 kg / unit for import
    set import to 10000 kg during year 1
    change import by +5 % each year

    # Service
    retire 6.7 % each year
    recharge 10 % each year with 0.30 kg / unit

    # Historic units already deployed
    set priorEquipment to (get equipment as units * 7) units during year 1
  end substance

end application

define application "residential AC"
```

```
uses substance "R-410A"
  enable domestic
  enable import
  equals 2088 kgCO2e / kg
  equals 80 kwh / unit

  # Domestic production
  initial charge with 0.90 kg / unit for domestic
  set domestic to 175000 kg during year 1
  change domestic by +5 % each year during years 1 to 5
  change domestic by +3 % each year during years 6 to 9

  # Trade
  initial charge with 0.90 kg / unit for import
  set import to 20000 kg during year 1
  change import by +5 % each year during years 1 to 5
  change import by +3 % each year during years 6 to 9

  # Service
  retire 6.7 % each year
  recharge 10 % each year with 0.90 kg / unit

  # Historic units already deployed
  set priorEquipment to (get equipment as units * 7) units during year 1
end substance

uses substance "HFC-32"
  enable domestic
  enable import
  equals 675 kgCO2e / kg
  equals 100 kwh / unit

  # Domestic production
  initial charge with 0.68 kg / unit for domestic
  set domestic to 85000 kg during year 1
  change domestic by +8 % each year

  # Trade
  initial charge with 0.68 kg / unit for import
  set import to 9000 kg during year 1
  change import by +8 % each year

  # Service
  retire 6.7 % each year
  recharge 10 % each year with 0.68 kg / unit
```

```
    # Historic units already deployed
    set priorEquipment to (get equipment as units * 7) units during year 1
  end substance

  uses substance "R-290"
    enable domestic
    enable import
    equals 3 kgCO2e / kg
    equals 80 kwh / unit

    # Domestic production
    initial charge with 0.68 kg / unit for domestic
    set domestic to 10000 kg during year 1
    change domestic by +5 % each year

    # Trade
    initial charge with 0.68 kg / unit for import
    set import to 10000 kg during year 1
    change import by +5 % each year

    # Service
    retire 6.7 % each year
    recharge 10 % each year with 0.35 kg / unit

    # Historic units already deployed
    set priorEquipment to (get equipment as units * 7) units during year 1
  end substance

end application

define application "mobile AC"

  uses substance "HFC-134a"
    enable domestic
    enable import
    equals 1430 kgCO2e / kg
    equals 100 kwh / unit

    # Domestic production
    initial charge with 0.90 kg / unit for domestic
    set domestic to 175000 kg during year 1
    change domestic by +5 % each year during years 1 to 5
    change domestic by +3 % each year during years 6 to 9

    # Trade
    initial charge with 0.90 kg / unit for import
```

```
      set import to 20000 kg during year 1
      change import by +5 % each year during years 1 to 5
      change import by +3 % each year during years 6 to 9

      # Service
      retire 6.7 % each year
      recharge 10 % each year with 0.90 kg / unit

      # Historic units already deployed
      set priorEquipment to (get equipment as units * 7) units during year 1
    end substance

    uses substance "R-1234yf"
      enable domestic
      enable import
      equals 4 kgCO2e / kg
      equals 80 kwh / unit

      # Domestic production
      initial charge with 0.90 kg / unit for domestic
      set domestic to 85000 kg during year 1
      change domestic by +8 % each year

      # Trade
      initial charge with 0.90 kg / unit for import
      set import to 9000 kg during year 1
      change import by +8 % each year

      # Service
      retire 6.7 % each year
      recharge 10 % each year with 0.90 kg / unit

      # Historic units already deployed
      set priorEquipment to (get equipment as units * 7) units during year 1
    end substance

  end application

end default


start policy "domestic refrigeration reuse"

  modify application "domestic refrigeration"

    modify substance "HFC-134a"
```

```
      recover 5 % with 100 % reuse with 100 % induction during year 3
      define level as limit (yearAbsolute - 3) * 10 to [0, 30]
      recover level % with 100 % reuse with 100 % induction during years 4 to onwards
    end substance

  end application

end policy


start policy "domestic refrigeration low-GWP"

  modify application "domestic refrigeration"

    modify substance "HFC-134a"
      define level as limit (yearAbsolute - 2) * 20 to [0, 100]
      replace level % of domestic with "R-600a" during years 3 to onwards
      replace level % of import with "R-600a" during years 3 to onwards
    end substance

  end application

end policy


start policy "commercial refrigeration reuse"

  modify application "commercial refrigeration"

    modify substance "HFC-134a"
      recover 5 % with 100 % reuse with 100 % induction during year 3
      define longTermRecovery as limit (yearAbsolute - 3) * 10 to [0, 30]
      recover longTermRecovery % with 100 % reuse with 100 % induction during years 4 to onw
    end substance

    modify substance "R-404A"
      recover 5 % with 100 % reuse with 100 % induction during year 3
      define longTermRecovery as limit (yearAbsolute - 3) * 10 to [0, 30]
      recover longTermRecovery % with 100 % reuse with 100 % induction during years 4 to onw
    end substance

  end application

end policy
```

```
start policy "commercial refrigeration low-GWP"

  modify application "commercial refrigeration"

    modify substance "HFC-134a"
      define level as limit (yearAbsolute - 2) * 20 to [0, 100]
      replace level % of domestic with "R-600a" during years 3 to onwards
      replace level % of import with "R-600a" during years 3 to onwards
    end substance

    modify substance "R-404A"
      define level as limit (yearAbsolute - 2) * 20 to [0, 100]
      replace level % of domestic with "R-600a" during years 3 to onwards
      replace level % of import with "R-600a" during years 3 to onwards
    end substance

  end application

end policy


start policy "residential AC reuse"

  modify application "residential AC"

    modify substance "R-410A"
      recover 5 % with 100 % reuse with 100 % induction during year 3
      define longTermRecovery as limit (yearAbsolute - 3) * 10 to [0, 30]
      recover longTermRecovery % with 100 % reuse with 100 % induction during years 4 to onw
    end substance

    modify substance "HFC-32"
      recover 5 % with 100 % reuse with 100 % induction during year 3
      define longTermRecovery as limit (yearAbsolute - 3) * 10 to [0, 30]
      recover longTermRecovery % with 100 % reuse with 100 % induction during years 4 to onw
    end substance

  end application

end policy


start policy "residential AC low-GWP"

  modify application "residential AC"
```

```
    modify substance "R-410A"
      define level as limit (yearAbsolute - 2) * 20 to [0, 100]
      replace level % of domestic with "R-290" during years 3 to onwards
      replace level % of import with "R-290" during years 3 to onwards
    end substance

    modify substance "HFC-32"
      define level as limit (yearAbsolute - 2) * 20 to [0, 100]
      replace level % of domestic with "R-290" during years 3 to onwards
      replace level % of import with "R-290" during years 3 to onwards
    end substance

  end application

end policy


start policy "mobile AC reuse"

  modify application "mobile AC"

    modify substance "HFC-134a"
      recover 5 % with 100 % reuse with 100 % induction during year 3
      define longTermRecovery as limit (yearAbsolute - 3) * 10 to [0, 30]
      recover longTermRecovery % with 100 % reuse with 100 % induction during years 4 to onw
    end substance

    modify substance "R-1234yf"
      recover 5 % with 100 % reuse with 100 % induction during year 3
      define longTermRecovery as limit (yearAbsolute - 3) * 10 to [0, 30]
      recover longTermRecovery % with 100 % reuse with 100 % induction during years 4 to onw
    end substance

  end application

end policy


start policy "mobile AC low-GWP"

  modify application "mobile AC"

    modify substance "HFC-134a"
      define level as limit (yearAbsolute - 2) * 20 to [0, 100]
      replace level % of domestic with "R-1234yf" during years 3 to onwards
      replace level % of import with "R-1234yf" during years 3 to onwards
```

```
    end substance

  end application

end policy


start simulations

  simulate "business as usual" from years 1 to 29

  simulate "domestic refrigeration high ambition"
    using "domestic refrigeration reuse"
    then "domestic refrigeration low-GWP"
  from years 1 to 29

  simulate "commercial refrigeration high ambition"
    using "commercial refrigeration reuse"
    then "commercial refrigeration low-GWP"
  from years 1 to 29

  simulate "residential AC high ambition"
    using "residential AC reuse"
    then "residential AC low-GWP"
  from years 1 to 29

  simulate "mobile AC high ambition"
    using "mobile AC reuse"
    then "mobile AC low-GWP"
  from years 1 to 29

end simulations
```

Note that this translation demonstrates the full feature set of QubecTalk including variables, conditionals, constraints, and policy stacking. Some language shortcuts could make this briefer, but this version demonstrates clear one-to-one mapping with business logic.

6 Additional Resources

For comprehensive documentation and examples:

- **User Guide**: https://kigalisim.org/guide/ - Interactive tutorials and step-by-step instructions
- **Web Application**: https://kigalisim.org - Browser-based Designer and Advanced Editor

- **GitHub Repository**: https://github.com/SchmidtDSE/kigali-sim - Source code and issue tracking
- **JAR Download**: https://kigalisim.org/kigalisim-fat.jar - Command-line interface
- **LLM Integration**: https://kigalisim.org/llms-full.txt - AI assistant documentation
- **JavaDoc**: https://kigalisim.org/guide/javadoc/ - Engine API documentation