

# gabc-parser / gabc-converter

License: GPLv3

[github.com/saybaar](https://github.com/saybaar)

Lydia Simmons <[saybaar@gmail.com](mailto:saybaar@gmail.com)>

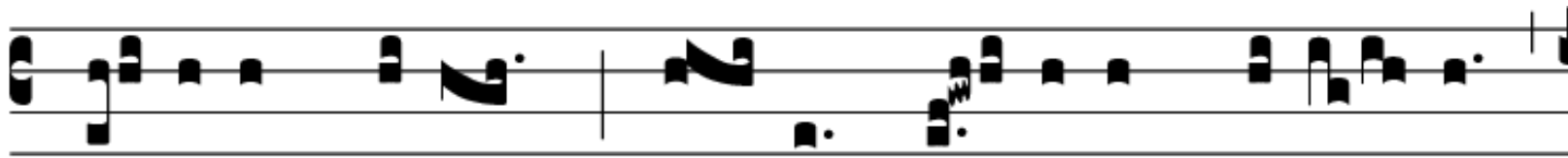
# gabc

name: Populus Sion;

%%

(c3) Pó(eh/hi)pu(h)lus(h) Si(hi)on,(hgh.) \*(;) ec(hihi)ce(e.)  
Dó(e.f!gwh/hi)mi(h)nus(h) vé(hi)ni(ig/ih)et(h.) (,)

P



ó- pu-lus Si- on, \* ec- ce Dó- mi-nus vé-ni- et

source: <http://gregorio-project.github.io/gabc/details.html>

# To JSON

```
name: Populus Sion;
```

```
%%
```

```
(c3) Pó(eh/hi)pu(h)lus(h) Si(hi)on,(hgh.) *(;) ec(hihi)ce(e.)  
Dó(e.f!gwh/hi)mi(h)nus(h) vé(hi)ni(ig/ih)et(h.) (,)
```

```
lydia@lydia-ideapad:~$ gabc-converter json -i ./populus_sion.gabc  
{  
  "attributes":  
    [[  
      "name",  
      " Populus Sion"  
    ]],  
  "syllables":  
    [{  
      "text": "",  
      "music": []  
    },  
    {  
      "text": " Pó",  
      "music":  
        [{  
          "Note":  
            {  
              "prefix": "",  
              "position": "e",  
              "suffix": "",  
              "current_clef": "c3"  
            }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "h",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        },  
          {  
            "Separator": "/"  
          },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "h",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "i",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        }  
      ],  
      "text":  
        "pu"  
    },  
    {  
      "text": "nus",  
      "music":  
        [{  
          "Note":  
            {  
              "prefix": "",  
              "position": "h",  
              "suffix": "",  
              "current_clef": "c3"  
            }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "i",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "h",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "i",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        }  
      ],  
      "text":  
        "vé"  
    },  
    {  
      "text": "ni",  
      "music":  
        [{  
          "Note":  
            {  
              "prefix": "",  
              "position": "h",  
              "suffix": "",  
              "current_clef": "c3"  
            }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "i",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "h",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        },  
          {  
            "Note":  
              {  
                "prefix": "",  
                "position": "i",  
                "suffix": "",  
                "current_clef": "c3"  
              }  
        }  
      ],  
      "text":  
        "et"  
    }  
  ],  
  "text":  
    "Populus Sion"  
}
```

# To Lilypond

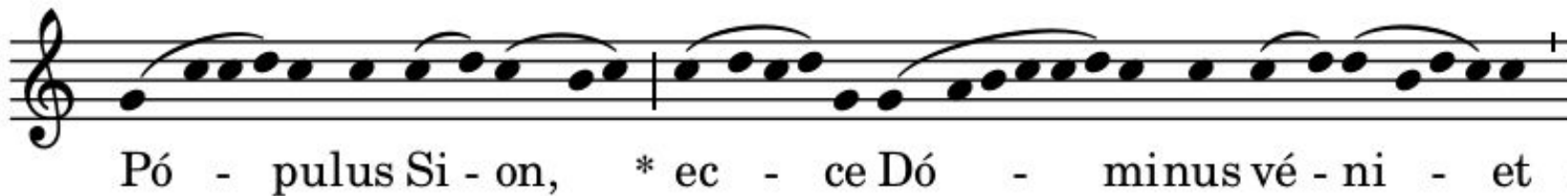
```
lydia@lydia-ideapad:~$ gabc-converter lilypond -i ./populus_sion.gabc
```

```
verba = \lyricmode {  
  Pó -- pu -- lus Si -- on, \set stanza = " *"  ec -- ce Dó -- mi --  
  nus vé -- ni -- et
```

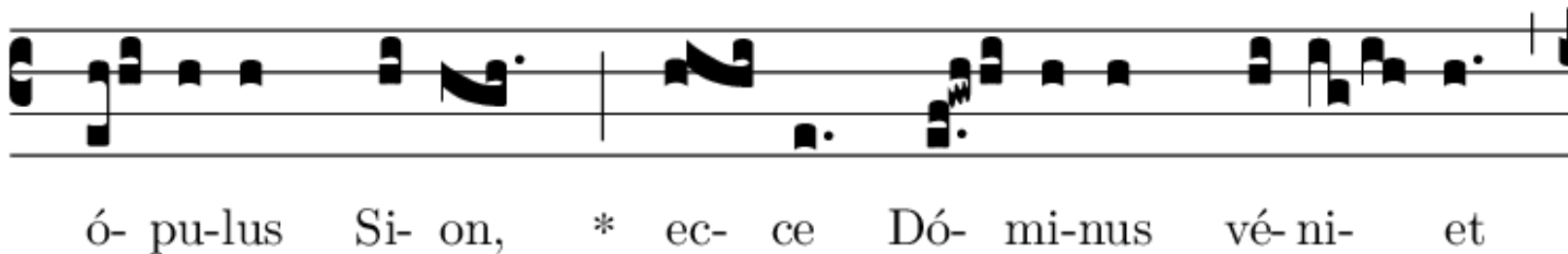
```
name: Populus Sion;  
%%  
(c3) Pó(eh/hi)pu(h)lus(h)  
Si(hi)on, (hgh.) *(;) ec(hihi)ce(e.)  
Dó(e.f!gwh/hi)mi(h)nus(h) vé  
(hi)ni(ig/ih)et(h.) (,)
```

```
g(c' c' d')  
c'  
c'  
c'(d')  
c'(b c')  
\divisioMaior  
c'(d' c' d')  
g  
g(a b c' c' d')  
c'
```

# To Lilypond



P



# Other gabc -> Lilypond converters

- by hand!
- **gabctk**: Elegant output, many additional conversion tools...in French
  - <<https://github.com/jperon/gabctk>>
- **gabc-to-ly**: Couldn't run fully
  - <<https://github.com/ahinkley/gabc-to-ly>>
- **lygre**: Difficult to install and use correctly (Ruby), with basic Lilypond output similar to gabc-converter
  - <<https://github.com/igneus/lygre>>

# Decisions & lessons

- Using the right tools can make things easy (pest, serde, clap)
- Building for extensibility & maintainability

## Improvements

- Handle a bigger subset of gabc syntax
- Translate into more sophisticated Lilypond (note shapes, accents)
- Find out what JSON format(s) would be most useful and implement those
- Publish to crates.io

# gabc-parser / gabc-converter

License: GPLv3

[github.com/saybaar](https://github.com/saybaar)

Lydia Simmons <[saybaar@gmail.com](mailto:saybaar@gmail.com)>



# pest grammar

```
file = { soi ~ (attribute ~ "\n") * ~ "%%\n" ~ (syllable | "\n") * ~ "\n"* ~ eoi }
```

```
attribute = { attribute key ~ ":" ~ attribute value ~ ";" }
```

```
attribute key = { (!(":"|";") ~ any)* }
```

```
attribute value = { (!(":"|";") ~ any)* }
```

$$\text{clef} = \{ ("c" \mid "f") \sim ('1'.. '4') \}$$

```
barline = { ( ";" ~ ('1'..'6') | ";" | ":::" | ":?" | ":" | ",_" | ",0" | ", "
| "'") }
```

```
spacer = { ( "!" | "@" | "/" | "//" | "/0" | "/[" ~ "-" ? ~ ('0'..'9') ~ "]" | " "
) }
```

$$\text{note} = \{ \text{prefix}^* \sim \text{position} \sim \text{suffix}^* \}$$

```
prefix = { "-" }
```

```
position = { ('a'..'m') | ('A'..'M') }
```

```
suffix = { ("~" | ">" | "<" | "v" | "V" | "o~" | "o<" | "o" | "w" | "s<" | "s" |
"x" | "y" | "#" | "." | ".." | "'" | " ") }
```

```
syllable = { string ~ "(" ~ (clef | note | barline | spacer)* ~ ")" }
```

```
string = { (! (" (" ~ any) * }
```