

SqueakMap

SqueakMap is a distributed shared catalog for the Squeak developer community. In practice it forms a yellow pages inside the image holding primarily information about Squeak developers and their Squeak packages and releases of those. It also has functions to download and install packages in a range of formats with a single mouse click and to maintain a local cache of packages.

Keywords

meta catalog, Squeak, packages, distributed, community

Platforms

Squeak 3.2.1 and later

License

Squeak-L

Developers

Göran Krampe is the lead developer with contributions and enhancements (in no particular order) by:

- Ned Konz
- Brian Rice
- Daniel Vainsencher
- Doug Way
- Andreas Raab
- Diego Gomez Deck
- Avi Bryant

...and enhancements/bug fixes from many more. :)

Demo

SqueakMap has a web UI available at:

<http://map1.squeakfoundation.org/sm>

SqueakMap is also accessible from within Squeak under “World menu->open->SqueakMap Package Loader”.

Details

Currently SqueakMap has over **200 registered developers** of which about 125 together have published over **450 packages** over 2 years.

SqueakMap both has a web user interface that is used to maintain the map, and various Squeak tools that can use the local map mirror inside the image. The standard **SqueakMap Package Loader** is the most commonly used tool but there are others too and the domain model is separated from the user interface.

In fact, anyone can easily use the domain model, for example, this little snippet produces a graph showing how the number of packages and accounts have increased over time (see image next side):

```
| pm x y offset newx |
pm := PlotMorph new.
pm color: Color blue; extent: 300 @ 300.
pm xAxisFormatter: [:xx | xx rounded asStringWithCommas].
pm yAxisFormatter: [:yy | yy rounded asString].
pm title: 'SqueakMap packages and accounts over time'.
pm series: #packages color: Color yellow; series: #accounts color: Color red.
Offset := nil.
#(packages accounts) do: [:axis |
    coll := ((SMSqueakMap default perform: axis)
             collect: [:p | p created date julianDayNumber ])
           asBag sortedElements.
    offset ifNil: [offset := coll first key + 1].
    y := 0. x := 0.
    coll do: [:ass |
        newx := ass key - offset.
        [x < newx] whileTrue: [
            x := x + 1. pm series: axis addPoint: x @ y].
        x := x + 1. y := y + ass value.
        pm series: axis addPoint: x @ y]].
pm openInWorld
```

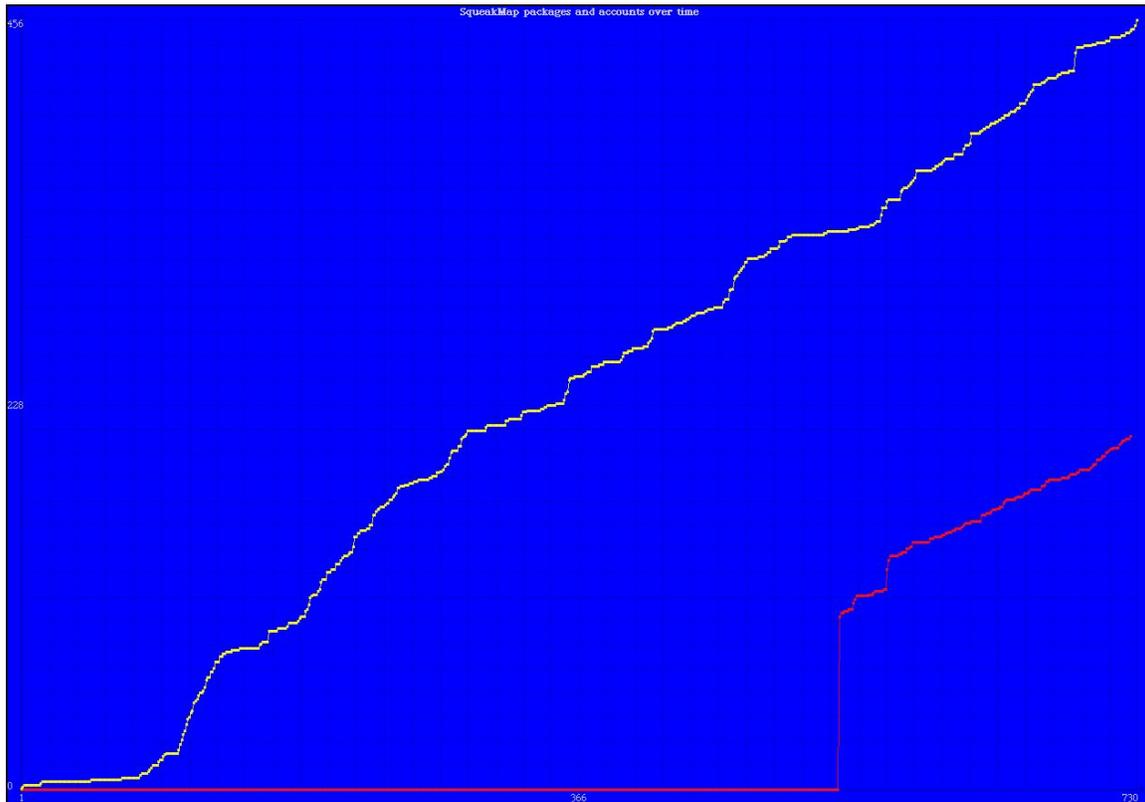
Isn't Smalltalk and Squeak great! :)

As can be seen in the plot the registration of packages has almost been linear, as has the registration of accounts. Accounts weren't represented as objects with creation dates in the beginning – that is why that graph starts at 103 accounts.

SqueakMap has turned into a key part of the collaboration architecture around Squeak and is enabling us to start making the image more modular and to distribute the responsibilities of maintaining it.

SqueakMap is used daily by Squeakers all over the world and today it is painful to think back on how awkward it was to find and install Squeak packages before it existed.

Graph showing packages and accounts over time



Future

SqueakMap keeps evolving and the next step is to make it capable of handling dependencies and through that offer automatic upgrades and installations of interdependent packages, very similar to how apt-get works in Debian Linux.

The goal is to move Squeak over to a model where it is evolved in parallel as a group of packages where each has its own release cycle.

The task of releasing Squeak itself then simply becomes a task of selecting and combining package releases to form a tested unity.

Since the beginning SqueakMap has been built with the intention that it should cooperate with all other projects and tools within our community. It is primarily a catalog of people and packages and will try to keep that focus.

The dependency mechanisms are built as a separate layer on top of SqueakMap and hopefully the generic mechanisms added during this work for attaching meta data to objects in SqueakMap will also enable other further experiments.