NAME

appstreamcli − Handle AppStream metadata and the AppStream index

SYNOPSIS

appstreamcli [**COMMAND**]

DESCRIPTION

This manual page documents briefly the **appstreamcli** command.

appstreamcli is a small helper tool to work with AppStream metadata and access the AppStream component index from the command−line. The AppStream component index contains a list of all available software components for your distribution, matched to their package names. It is generated using AppStream XML or Debian DEP−11 data, which is provided by your distributor.

For more information about the AppStream project and the other components which are part of it, take a look at the AppStream pages at **Freedesktop.org**[1].

OPTIONS

status

Display various information about the installed metadata and the metadata cache.

get *ID*

Get a component by it's identifier.

s, **search** *TERM*

Search the AppStream component database for a given search term.

what−provides *TYPE TERM*

Return components which provide a given item. An item type can be specified using the *TYPE* parameter, a value to search for has to be supplied using the *TERM* parameter.

Examples:

Get components which handle the "text/xml" mediatype.

appstreamcli what−provides mediatype "text/xml"

Get component which provides the "libfoo.so.2" library.

appstreamcli what−provides lib libfoo.so.2

refresh, **refresh−cache**

Trigger a database refresh, if necessary. In case you want to force the database to be rebuilt, supply the **−−force** flag.

This command must be executed with root permission.

dump *ID*

Dump the XML description of a component with the given ID.

validate *FILES*

Validate AppStream XML metadata for compliance with the specification.

Both XML metadata types, upstream and distro XML, are handled. The filetype which should be validated is determined automatically.

The **−−pedantic** flag triggers a more pedantic validation of the file, including minor and style issues in the report.

validate−tree *DIRECTORY*

Validate AppStream XML metadata found in a file−tree.

This performs a standard validation of all found metadata, but also checks for additional errors, like the presence of .desktop files and validity of other additional metadata.

install *ID*
Install a software component by its ID.

This resolves the AppStream component ID to the package name which provides it and then calls the distribution's package manager to install the software.

This feature is experimental.

put *FILE*
Install a metadata file into the right directory on the current machine.

compare−versions *VER1 [CMP] VER2*
Compare two version numbers. If two version numbers are given as parameters, the versions will be compared and the comparison result will be printed to stdout.

If a version number, a comparison operator and another version number are passed in as parameter, the result of the comparison operation will be printed to stdout, and **appstreamcli** will exit with a non−zero exit status in case the comparison failed. The comparison operator can be one of the following:

- eq − Equal to

- ne − Not equal to

- lt − Lesser than

- gt − Greater than

- le − Lesser than or equal to

- ge − Greater than or equal to

new−template *TYPE FILE*
Create a metainfo file template to be used by software projects. The **−−from−desktop** option can be used to use a .desktop file as template for generating the example file.

The generated files contain example entries which need to be filed in with the actual desired values by the project author.

The first *TYPE* parameter is the name of an AppStream component type. For a complete list check out **the documentation**[2] or the help output of **appstreamcli** for this subcommand.

make−desktop−file *MI_FILE DESKTOP_FILE*
Create a XDG desktop−entry file from a metainfo file. If the desktop−entry file specified in *DESKTOP_FILE* already exists, it will get extended with the new information extracted from the metainfo file. Otherwise a new file will be created.

This command will use the first binary mentioned in a provides tag of the component for the Exec= field of the new desktop−entry file. If this is not the desired behavior, the **−−exec** flag can be used to explicitly define a binary to launch. Other methods of launching the application are currently not supported.

In order to generate a proper desktop−entry, this command assumes that not only the minimally required tags for an AppStream component are set, but also that it has an <icon/> tag of type "stock" to describe the stock icon that should be used as well as a <categories/> tag containing the categories the application should be placed in.

news−to−metainfo *NEWS_FILE MI_FILE [OUT_FILE]*
This command converts a NEWS file as used by many open source projects into the XML used by

AppStream. Since NEWS files are free text, a lot of heuristics will be applied to get reasonable results. The converter can also read a YAML version of the AppStream release description and convert it to XML as well. If the metainfo file *MI_FILE* already exists, it will be augmented with the new release entries, otherwise the release entries will be written without any wrapping component. If *[OUT_FILE]* is specified, instead of acting on *MI_FILE* the changed data will be written to the particular file. If any of the output filenames is set to "−", the output will instead be written to stdout.

The **−−format** option can be used to enforce reading the input file in a specific format ("text" or "yaml") in case the format autodetection fails. The **−−limit** option is used to limit the amount of release entries written (the newest entries will always be first).

metainfo−to−news *MI_FILE NEWS_FILE*
> This command reverses the **news−to−metainfo** command and writes a NEWS file as text or YAML using the XML contained in a metainfo file. If *NEWS_FILE* is set to "−", the resulting data will be written to stdout instead of to a file.

> The **−−format** option can be used to explicitly specify the output format ("yaml" or "text"). If it is not set, **appstreamcli** will guess which format is most suitable.

−−details
> Print out more information about a found component.

−−no−color
> Don't print colored output.

−−no−net
> Do not access the network when validating metadata.

> The same effect can be achieved by setting the **AS_VALIDATE_NONET** environment variable before running **appstreamcli**.

−−version
> Display the version number of appstreamcli

SEE ALSO
pkcon (1).

AUTHOR
This manual page was written by Matthias Klumpp <matthias@tenstral.net>.

COPYRIGHT
Copyright © 2012-2019 Matthias Klumpp

NOTES
1. Freedesktop.org
   http://www.freedesktop.org/wiki/Distributions/AppStream/

2. the documentation
   https://www.freedesktop.org/software/appstream/docs/chap-Metadata.html