NAME

gmautil – General Tcl Utility Routines

SYNOPSIS

(If package installed globally)
`package require gmautil`

(Otherwise)
`source gmautil.tcl`

`::gmautil::dassign` *dict key1 var1 ?key2 var2...?*

`::gmautil::is_git` *path* → `bool`

`::gmautil::lpop` *var index* → *element*

`::gmautil::my_arch` → *arch*

`::gmautil::my_os` → *os*

`::gmautil::rdist` *min max cmd arglist var ?var...?*

`::gmautil::upgrade` *destination tmp url file old new strip launch msg proxy curl*

`::gmautil::verify` *data signature* → *bool*

`::gmautil::version_compare` *v1 v2* → cmp

DESCRIPTION

This module provides a set of miscellaneous utility functions that could be useful to more than one part of the overall product suite.

`::gmautil::dassign` *dict key1 var1 ?key2 var2...?*

This is analogous to the Tcl built-in `lassign` command, but it operates on a dictionary value *dict* instead of a list. For each pair of arguments after the *dict* value, the named *key* is looked up in the dictionary and its value is assigned to the variable *var*.

If any *key* is a space-separated list of names, they are considered to be a list of nested sub-keys to be fetched from the dictionary in the same manner as `dict get`. For example, given the *key* `{foo bar}` it will retrieve the value stored under the key `foo`, which is itself a dictionary value, in which the key `bar` is looked up, and its value assigned to the corresponding *var* variable.

`::gmautil::is_git` *path*

Returns true if the directory *path* resides inside a Git working directory tree.

`::gmautil::lpop` *var index*

This removes the *index*th element from the list variable *var*, returning the removed element. This is intended to provide the basic functionality of the Tcl 8.7 built-in command `lpop`, and in fact if the Tcl interpreter is at least version 8.7 it simply calls the `lpop` provided by the language, which has additional capabilities not implemented by this version.

`::gmautil::my_arch`

Returns the name of the hardware architecture running the program, with naming convention standard to the GMA suite. Currently, it translates the Tcl runtime's machine names as obtained from `tcl_platform(machine)` with the following translation:

Tcl Name	**GMA Name**
x86_64	amd64

Any name not otherwise listed is returned as-is from the Tcl runtime. More names will likely be added to this translation list in the future.

`::gmautil::my_os`

Returns the name of the operating system running the program, with naming convention standard to the GMA suite. Currently, it translates the Tcl runtime's os names as obtained from `tcl_platform(os)` with the following translation:

Tcl Name	**GMA Name**
Darwin	darwin
Linux	linux
FreeBSD	freebsd

Any name not otherwise listed is returned as-is from the Tcl runtime. More names will likely be added to this translation list in the future.

`::gmautil::rdist` *min max cmd arglist var* ?*var...*?
This is a specialized version of `lassign`, which is intended to receive a number of parameters in *arglist* to a GMA command called *cmd*. It requires that *arglist* contain at least *min* ant at most *max* elements, generating an error if that condition is not met. The values in *arglist* are then distributed out into the list of named variables in the order given. If there are more *var* variable names than elements in *arglist*, the remaining variables are set to the empty string. Note that *cmd* is only used for forming error messages.

`::gmautil::upgrade` *destination tmp url file old new strip launch msg proxy curl*
This function facilitates automatic upgrades within an application. The application is downloaded using the CURL program whose pathname is given in *curl* (using a proxy server as specified by *proxy* if that value is non-empty). The file itself is obtained from the server directory indicated by the *url* value, with the base filename *file* and suffix `.tar.gz` and `.tar.gz.sig` (the latter being a cryptographic signature verifying the authenticity of the compressed tar file). These files are downloaded to the temporary directory *tmp*.

Once the files are downloaded, the signature is checked, and then the files are extracted into the destination diretory *destination*, although a leading *strip* (possibly prefixed with `./`) is removed from the names as they exist in the tar file.

The tar file is expected to contain a manifest file in its top-level directory, called `__checksums__`. Each line of this file consists of a SHA256 checksum in hex, whitespace, a type character (space for text files, `*` for binary files, `?` for portable-newline files, or `^` for BITS files), then the path of the file. Only binary files are supported at this time. Every file extracted from the archive must have an entry in this manifest file, and the installed file's checksum must match the manifest entry.

Once that is all accomplished, the downloaded files are removed and, if *launch* is non-empty, the program will attempt to start the program *launch* relative to *destination* and then the calling program is terminated.

old and *new* are the current and to-be-installed version numbers. Warnings will be issued before downgrading. The routine will refuse to continue if the versions are equal to each other.

`::gmautil::verify` *data signature*
Given a binary data string *data*, return true if the binary cryptographic signature in *signature* is valid for that data. This uses our built-in product public key.

`::gmautil::version_compare` *v1 v2*
Compares two version number strings to see which order they should go in, chronologically. Each string must consist of one or more integers (as ASCII digit sequences) separated by dots. The first such number is the most significant (major) version, the next is the next-most significant sub-version within the major one, and so forth. Thus, 1.2 comes before 1.10, 2.2.4 comes before 2.3, etc. Returns 0 if *v1* and *v2* are equal or equivalent, a number <0 if *v1*<*v2* (i.e., *v1* comes before *v2*), or a number >0 if *v1*>*v2*.

DIAGNOSTICS

An exception is thrown if a serious error is encountered.

SEE ALSO

`openssl`(1).

AUTHOR

Steve Willoughby / steve@madscience.zone.

HISTORY

This document describes version 1.1 of the `gmautil` package, released in December 2022.

Initial version created 17 July 2020.

COPYRIGHT