URL: http://cxc.harvard.edu/ciao3.4/xpaget.html
Last modified: December 2006

AHELP for CIAO 3.4

xpaget

Context: xpa

Jump to: Description Examples CHANGES IN CIAO 3.2 See Also

Synopsis

Retrieve data from one or more XPA servers.

Syntax

```
xpaget( String_Type dest )
xpaget( String_Type dest, String_Type cmd )

Return value: Array_Type

XPAGet( XPA_Type xpaHdl, String_Type dest, String_Type cmd )
XPAGet( XPA_Type xpaHdl, String_Type dest, String_Type cmd, String_Type
max_rec )
XPAGet( XPA_Type xpaHdl, String_Type dest, String_Type cmd, String_Type
max_rec, String_Type mode )

Return values: (Array_Type results, Array_Type names, Array_Type
messages)
```

Description

The xpaget() and XPAGet() functions are used to retrieve data from one or more XPA servers. An example would be to ask ds9 what file it was displaying. The xpaget() function provides a simple, easy to use, interface which is similar to the xpaget command−line tool from the XPA package, whilst the XPAGet() function gives the user full control.

The XPAGetToFile() function can be used to write the data straight to a file instead of having the data returned in a S−Lang variable. This can be useful when the volume of data is large, such as the pixel values of a large image. The XPAGetB() function may also be of interest in such situations.

The xpaget() function

When no XPA command is given the application(s) contacted typically return a list of the XPA commands that the application supports, although this is application−dependent. In the most common case only a single application server will be contacted by any single xpaget call, in which case the routine will return an array with only one element. If no servers are found then a zero−length array will be returned.

The XPAGet() function

This function is similar to xpaget(), but provides more user control. It uses an XPA_Type handle returned by XPAOpen(), and permits the user to explicitly limit the number of applications to contact (the max_rec parameter). The mode parameter is currently unused. The names and msgs arrays returned by the routine give the names of the servers that were accessed and an indication of any error condition from that server.

Example 1

```
chips> xpaget ("xpans")
String_Type[1]
chips> print (xpaget ("xpans"))
v2.0.ciaouser.machine.cfa.harvard.edu. chips gs
/tmp/.xpa/v2.0.ciaouser.machine.cfa.harvard.edu._chips.13264 ciaouser
v2.0.ciaouser.machine.cfa.harvard.edu. prism gs
/tmp/.xpa/v2.0.ciaouser.machine.cfa.harvard.edu._prism.2244 ciaouser
```

When an application with a XPA client is started, the "xpans" XPA server is created (or updated if it already exists) with information about that client. The "xpans" server can then be queried to find out what servers (other than itself) are available; this is done by the xpaget() call above. Since there is only one xpans server running then the return value of the function is a String_Type array with 1 element.

The information returned by the call tells us that there are two XPA access points available: "chips" and "prism" (the second elements on each line). The information returned by this call is the same as the xpans command−line tool from the XPA distribution.

For most XPA servers, using xpaget() with just the name of the server will return a list of the commands that that server understands. So

```
chips> print (xpaget ("prism"))
```

will return information on the XPA commands that prism understands. Note that you can not query the ChIPS XPA access point from the same ChIPS process − i.e. in the situation above

```
xpaget( "chips" )
```

will not return any information.

Example 2

```
chips> ns = xpaget ("xpans")
chips> ns = strtok (ns[0], "\n")
chips> length (ns)
2
chips> print (ns[1])
v2.0.ciaouser.machine.cfa.harvard.edu. prism gs
/tmp/.xpa/v2.0.ciaouser.machine.cfa.harvard.edu._prism.2244 ciaouser
```

In this example we store the return value from the xpaget() call in the variable ns. We convert the first element of ns into an array of strings using the strtok() function (from the S−Lang Run Time Library). Each element of the

array corresponds to one line of the output (since we split on the new−line character "\n").

We can use strtok() and the array−index capabilities of S−Lang to extract the names of each access point. We do this below by taking advantage of the fact that there are 5 elements per line and we want the second element of each line:

```
chips> ns = xpaget ("xpans")
chips> ns = strtok (ns[0])
chips> i = [1:length(ns):5]
chips> servers = ns[i]
chips> writeascii (stdout, i, servers)
1 chips
6 prism
```

Example 3

```
chips> fnames = xpaget ("prism", "file")
chips> fname = fnames[0]
```

Here we ask prism to tell us the name of the file that it is currently displaying. A shorter version would be to say:

```
chips> fname = xpaget ("prism", "file")[0]
```

CHANGES IN CIAO 3.2

The return values of xpaget() and XPAGet() have changed in CIAO 3.2. See the "Backwards Compatability" section below for a way of loading the module so that the old behavior is retained. The changes are:

- xpaget() now always returns an array of strings, even when only one server was contacted;
- XPAGet() now returns three arrays − results, names, and messages − rather than one variable (results) which could be a string or array of strings.

These changes mean that the xpaget() routine now cleanly handles the case of no server being contacted; it now returns an array with no elements rather than returning nothing, which would lead to "Stack Underflow" errors.

Backwards Compatability

The old behavior can be retained by declaring the variable

```
_CIAO3_XPA_COMPAT_
```

in the Global namespace prior to the first loading of the module. As an example, after:

```
if (0 != _featurep("xpa"))
  error("The XPA module has already been loaded");
public variable _CIAO3_XPA_COMPAT_;
require("xpa");
```

then the CIAO 3.1 versions of the commands will be used. The first statement, fearuting the _featurep() function, is not necessary, but added as a precaution to check that the XPA module has not previously been loaded.

See Also

modules

xpa

xpa

_slxpa_version, slxpa_errno, xpa_maxhosts, xpa_version, xpaaccess, xpaclose, xpagetb, xpagettofile, xpaopen, xpaset

---

URL: http://cxc.harvard.edu/ciao3.4/xpaget.html
Last modified: December 2006