xinet™

Xinet Guide to Portal Data Interchange Version 4.5

northplains™

Connecting your world. Visually.

Contents

Chapter 10

Introduction to the Xinet WebNative Portal data interchange: the *portalDI* CGI

This guide provides information about how to communicate with WebNative Suite using the *portalDI* CGI. The *portalDI* CGI is a query-based application that takes over for and streamlines the functionality of a set WebNative CGIs upon which WebNative Portal formerly depended for such things as:

- Constraining information displays
- Displaying *Browse* and *Search* results
- Displaying and manipulating *Shopping Baskets*
- Displaying and interacting with metadata

Xinet provides information in this *Guide* about the *portalDI* binary so that developers can customize and extend the WebNative Suite beyond those options which ship with the software. This CGI is platform- and program-independent and also faster than the CGIs WebNative Portal previously called upon.

Who should read this document?

The *WebNative Data Interchange Guide* has been written for programmers who want to integrate third party software with a WebNative Suite workflow and for those who want to create custom widgets that extend WebNative Suite functionality.

Document organization

This guide provides:

- An overview of the *portalDI* binary
- The *portalDI* vocabulary and syntax
- Further reading
- JSON and CGI RFCs

Conventions

We use several typographical conventions in this manual to help readers distinguish what must be typed as is and what is simply a parameter to be substituted. In particular, readers should type text set in a typewriter-like font, `like this`, *literally.* For example:

```
# more README          or c:\> type README
```

Readers should enter this command exactly as shown above (minus the shell prompt on UNIX systems, #, or the `c:\>` prompt on Windows systems). Readers should substitute their own applicable text for text set in slanted typewriter type, *`like this`*. For example, when one sees

```
% ls directory_name     or C:\> dir directory_name
```

one might type:

```
% ls doc                or C:\> dir doc
```

On UNIX systems

The two examples given above illustrate another convention: when a command must be issued by the superuser (*root*), you will see the prompt `#`; when a command may be issued by a normal user, you will see the prompt `%`. As in most documentation for UNIX software, text references to UNIX functions have the form *more*(1). The word in italic type is the name of the function; the parenthesized number is the section of the *UNIX Programmer's Manual* containing the function's manual page.

On Windows systems

When the manual discusses locations for files and programs, it uses default installation path names. If you install files and programs in other locations, the paths will not be the same as in this manual.

Where to look for technical information about Xinet products

Xinet technical information spans several volumes, with each software distribution having an appropriate guide. The guides are available for downloading in the *Maintenance* section of the Xinet Web site, *www.xinet.com.* The Web site also provides links for ordering printed copies.

To help you easily find what you need, the guides are extensively indexed and cross-referenced, and whenever possible, the index entries and cross references provide live links to the appropriate pages within the PDF.

Xinet technical documentation includes:

- The *WebNative Suite Administration Guide*

  This guide provides basic information about installation, licensing and WebNative Suite administration. It also provides information to consider before installation, such as optimal file-system organization and security issues. Chapters include details about administrative settings and basic ways to customize the look and feel of Web sites.

  The guide includes information about setting up everyday management of the WebNative Suite database, including ways to customize data fields and control their use by users and groups. It also contains information about automating production activities based on changes in the database and information about importing data.

  Within this guide, you'll also find information about file-sharing and print spooling optimizations for production workflows and details about the many options available when running WebNative Suite software in a production environment.

- The *Xinet Client Guide*

  This guide provides information to help end-users who are WebNative Suite server clients—with either direct access to the server or access through a Web browser. It explains the growing list of Xinet utilities that help users get the most out of a WebNative Suite server. It also includes tips for working within that environment when using Macintosh or Windows applications from Quark, Adobe, etc., as well as features available when sites make use of Video for WebNative Suite, Archive for WebNative Suite, and WebNative Portal.

- *Xinet PC Connectivity Guide*

  This guide provides information about installing, setting up and maintaining the Xinet modified Samba software which allows Microsoft Windows clients to interact with WebNative Suite.

- The *Xinet Guide to Development APIs*

  Not all sites where Xinet software is installed will need this volume. It presents information about WebNative Suite that will interest programmers customizing the product beyond ways made available through the GUIs which ship with it. Turn here for command-line customization and diagnostics and more significant changes to WebNative Suite interfaces.

- The *WebNative Portal Guide*

  This guide explains installation, licensing and administration for WebNative Portal sites. It also provides an overview of using WebNative Portal to customize the look and feel of user sites and gives examples of building new functionality to even further extend the product.

- The *Video for Xinet WebNative Suite Administration Guide*

  This guide provides information about installing and configuring the WebNative Suite Video module, including information about its various options, settings, metadata capabilities and interactions with the WebNative Suite database.

- The *Archive for Xinet WebNative Suite Administration Guide*

  This guide provides details about using WebNative Archive to allow WebNative users to communicate over the Web with *Symantic Backup Exec*™ archiving software. The guide explains installation and administration and provides guidelines for user interaction over the Web.

- On-line manual pages

  UNIX-style on-line *man*(1) pages are included with UNIX distributions. While not intended for the lay reader, they provide handy reference for more technically-advanced administrators and for programmers who want to work with Xinet programs from the command line. The *man*(1) pages are also included as part of each PDF manual.

- WebNative Suite on-line help

  Context-sensitive Help buttons in the WebNative Suite administration interfaces provide information about options in GUIs. They open appropriate pages within the *WebNative Suite Administration Guide*.

- *Xinet TechNotes*

  Xinet TechNotes provide information about technical issues not included in manuals. Like the *Xinet Guide to Development APIs* and *man*(1) pages, the notes are often aimed at those extending Xinet products beyond functionality offered in Xinet GUIs. They also provide information on issues that change quickly or that are of a transient nature.

If you cannot solve your problems from information in the manuals, please contact your Authorized Xinet Integrator or Xinet technical support.)

Chapter 2

Data interchange overview

Prior to Version 15.05 of WebNative Suite, developers depended on a set of WebNative JavaScript CGIs for communication with its WebNative Suite server. These included:

- *listdir* for listing directories and volumes
- *searchengine* for searching
- *toplevel* to provide an entry point for WebNative
- *imageinfo* for getting detailed image info
- *getimage* for extracting Web-ready previews of images
- *filemgr* for file management
- *basketbuttons* for determining plug-in availability
- *basketcontrol* for giving access to a user's basket
- *mview* for showing QuarkXPress, PDF and InDesign file previews
- *showbasket* for providing information about basket contents
- *streamfile* for streaming files to a browser

With Version 15.05, a single program- and platform-independent application on the WebNative Suite server, called *portalDI,* could be used instead *(/usr/etc/webnative/portalDI* on UNIX systems and *C:\Program File\Xinet\WebNative\Bin\portalDI.exe* on Windows). The newer *portalDI* CGI, which, because of its RESTful architecture and leaner, highly-compliant JSON inquiry format, has proven to be much faster, and brings along the advantage of eliminating SOAP, which XML used to require. JSON compliance also means that the *portalDI* CGI can be used to exchange data between the WebNative Suite server and applications written in ActionScript, C, C#, ColdFusion, Common Lisp, E, Erlang, Java, JavaScript, Lua, Objective CAM, Perl, PHP, Python, Rebol, Ruby, and Scheme or any other language that is JSON compliant.

Interaction with the *portalDI* binary is quite straightforward: your application sends CGI-compliant[1] queries for data. In turn, the *portalDI* binary on the WebNative Suite server returns answers formatted according to the *The application/json Media Type for JavaScript*

[1] D. Robinson and K. Coar, The Common Gateway Interface (CGI), Version 1.1, The Apache Software Foundation, October 2004 © The Internet Society, 2004, http://www.ietf.org/rfc/rfc3875. (Reprinted as Appendix C in this guide.)

Object Notation (JSON), JSON[2] RFC, with only the exact data you requested, nothing extra. (Thus, faster responses that don't require filtering.)

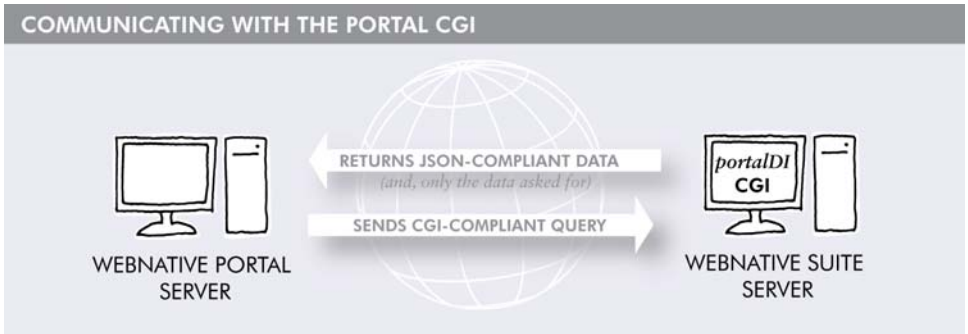

Figure 2-1 The *portalDI* CGI

For debugging use, you can use your browser's URL line to pass inquiries on to the portalDI binary and capture what the server returns using a plug-in or extension for your browser. An example follows:

Query: `portalDI?action=showbasket&showbaskbtns=true`

What it shows:

- Using `portalDI?` in the URL initiates a *portalDI* query.
- An `action=showbasket` query requests information from the `BASKET_INFO` array.
- The `showbaskbtns=true` lists available basket plug-ins and information along with the user's permission settings for them
- The reservered character `&` strings together mulitple queries.

Figure 2-2 shows the query in the URL.

[2] D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*, JSON.org, 2006 © The Internet Society, *http://tools.ietf.org/html/rfc4627.* (Reprinted as *Appendix D* in this guide.)

Query the portalDI CGI *on the WebNative Suite server.*

Show contents of the Shopping Basket (the BASKET_INFO array).

List Basket plug-ins and permissions for them

Figure 2-2 A query about the contents of the *Shopping Basket*

Figure 2-3 shows the results of the query looked at with Firebug

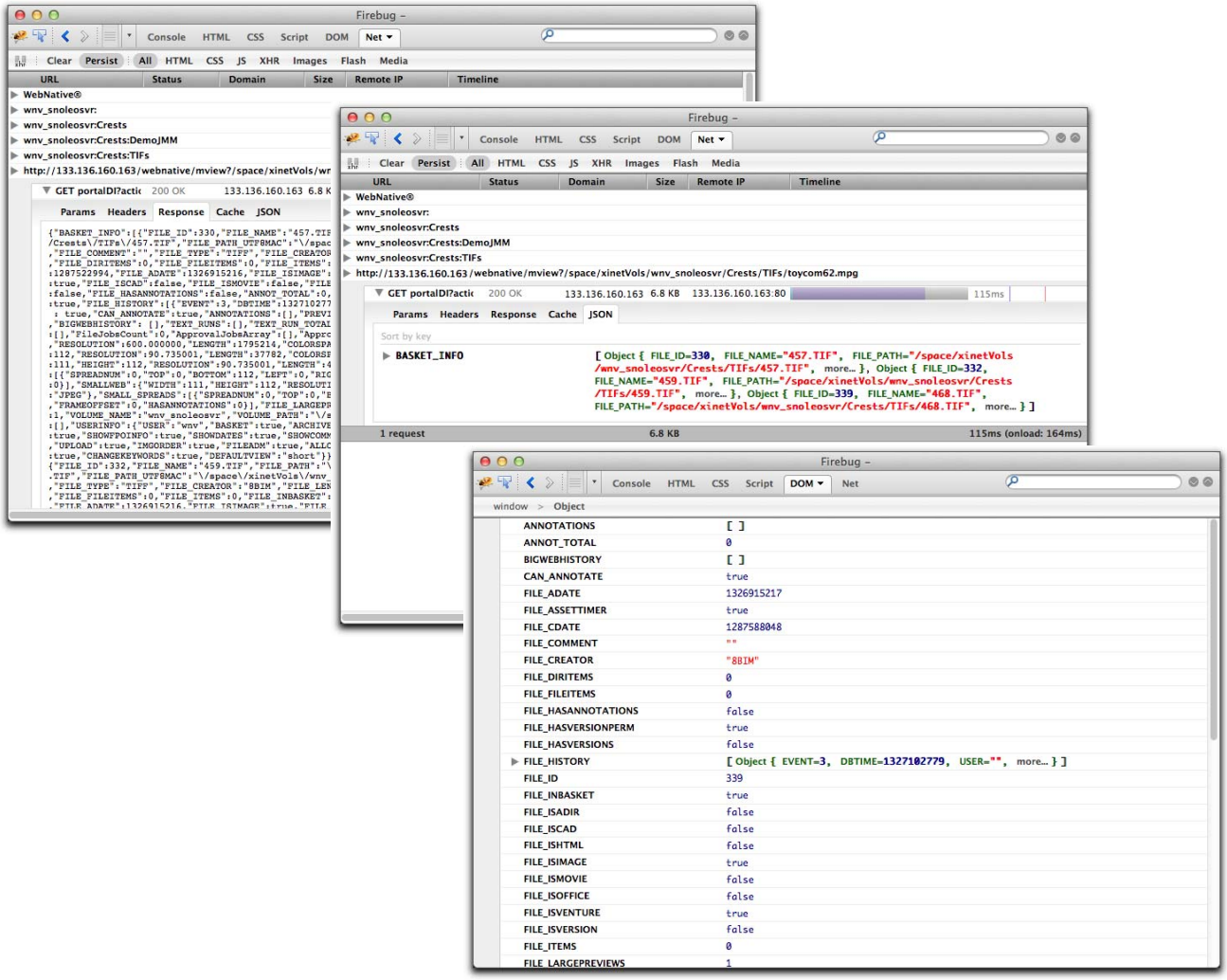

Figure 2-3 Examining the results of the query using Firebug

Chapter 3

Syntax

Command summary

portalDI? [action = version | showvols | showusersettings | showkywdperms | showbaskbtns | showiccusm | clearbasket] & [clientaddr = *ipaddress*] & [debug = true | false]

portalDI? [action = showdirinfo | fileinfo | showbasket | addbasket | removebasket] & [showkywds = true | false] & [clientaddr = *ipaddress*] & [debug = true | false] & ([basketname = *usersbasket*] | [basketfile = */path/to/basketfile*]) & ([fileid = *fileid_num*] | [path = */path/to/file*])

portalDI? [action = upload] & [dir = *path/to/target/directory*] & [newdir = *new_dir_name*] & [newname = *file_name*] & [clientaddr = *ipaddress*] &[overwrite = *true | false*] &[create_date = *unix_date*] & [modification_date = *unix_date*] & [dir_start = *value*] & [dir_end = *value*] &[filedata = *value*] & [keywordN = *metadata_value*]

portalDI? [action = getorderimage] & [archiveformat = *format*] & [backcolor = *value*] & [blackpoint = *true | false*] & [clipping = *value*] &[colorplate = *value*] &[colorspace = *value*] & [crop = *(cropw,croph)*] & [dpi = *output_resolution*] & [format = *value*] & [height = *value*] & [includemeta = *true | false*] & [inputicc = *true | false*] & [mergeclip = *true | false*] & [outputicc = *true | false*] & [overprint =*true | false*] & [preview = *true | false*] & [pctiff = *true | false*] & [pict = *true | false*] & [renderingintent = *value*] & [scale = *value*] & [spotcolor = *true | false*] & [spotoff = *true | false*] & [spreadnum = *true | false*] & [usm = *value*] & [watermark = *true | false*] & [webready = *true | false*] & [width = *value*]

portalDI? [action = streamfile] & [videoid = *video_num*] & [filetype = *value*] & [attach = true | false] & [clientaddr = *ipaddress*] & [debug = true | false] & ([fileid = *fileid_num*] | [path = */path/to/file*])

portalDI? [action = submitkywd] & [comment = *value*] & ([keyword*123* = *value*] & [keyword*789* = *value*]) & [clientaddr = *ipaddress*] & [debug = true | false] & ([fileid = *fileid_num*] | [path = */path/to/file*])

portalDI? [action = getimage] & [pixels = *num_pixels*] & [bgcolor_r = *background color red*] & [bgcolor_g = *background color green*] & [bgcolor_b = *background color blue*] & [spreadnum = *spread_num*] & [filetype = *value*] & [archtype = *value*] & [archname = *value*] &

```
[packerrors = true | false] & [clientaddr = ipaddress] & [debug = true
| false] & ( [fileid = fileid_num] | [path = /path/to/file] )

portalDI? [action = filemgr ] & [filemgraction = mkdir | move | copy |
delete | promote | version] & [newpath = /path/to/destination/folder]
& [newname = value] & [overwrite = true | false] & [clientaddr =
ipaddress] & [debug = true | false] &
( [fileid = fileid_num] | [filename = /path/to/original/file] )

portalDI? [action = annotations] & [page = value] & [grouponly = value]
& [clientaddr = ipaddress] & [debug = true | false] & ( [fileid =
fileid_num] | [path = /path/to/file] )

portalDI? [action = saveannotations] & [page = value] & [grouponly =
value] & [clientaddr = ipaddress] & [debug = true | false] & ( [fileid
= fileid_num] | [path = /path/to/file] )

portalDI? [action = browse] & [fileid = value | path = value] &
[itemsperpage = value & page = value] | [showfiles = false | [true &
filesperpage = value & filepage = value] & showdirs = false | [true &
dirsperpage = value & dirpage = value]] & [resultsetid = value] &
[filename_flag_i = -1 | value] & [filename_sort_i = value] &
[date_sort_n_i = value ] & [dbsearch_flag_n_i = -1 | value] &
[dbsearch_sort_n_i = value] & [showkywds = true | false]

portalDI? [action = navigator] & [fileid = value | path = value] &
[showfile = false | true]

portalDI? [action = presearch] & [searchid = value] & [sharesearches =
true] & [delsavedsearch = true & searchid = value]

portalDI? [action = search] &
( [subsearch_Y = value & subsearch_logic_Y = value] &
[searchall_X = value & searchall_flag_X = value & searchall_logic_X =
  value] &
[filename_X= value & filename_flag_X = value & filename_logic_X =
  value] &
[filetype_X = value & filetype_flag_X & filetype_logic_X = value] &
[date_Y_X = value & date_flag_Y_X = value & date_logic_Y_X = value] &
[comment_X = value & comment_flag_X = value & comment_logic_X = value] &
[searchallkeyword_X = value & searchallkeyword_flag_X = value &
  searchallkeyword_logic_X = value] &
[searchallfti_X = value & searchallfti_flag_X = value &
  searchallfti_logic_X = value] &
[dbsearch_keyword_Y_X = value & dbsearch_flag_Y_X = value &
  dbsearch_logic_Y_X = value] &
[annotations_X = value & annotations_flag_X = value &
  annotations_logic_X = value] &
[highresinfo_Y_X = value & highresinfo_flag_Y_X = value &
  highresinfo_logic_Y_X = value] &
[videoinfo_Y_X = value & videoinfo_flag_Y_X = value &
  videoinfo_logic_Y_X = value] &
[filecontent_X = value & filecontent_flag_X = value &
  filecontent_logic_X = value] &
[event_Y_X = value & event_flag_Y_X = value & event_logic_X = value] &
[stamps_X= value & stamps_flag_X = value & stamps_logic_X = value] &
```

```
[filedir_X = value & filedir_logic_X = value] &
[filesize_X = value & filesize_flag_X = value & filesize_logic_X =
  value] &
[assettimer_X = value & assettimer_flag_X = value & assettimer_logic_X
  = value] &
quicksearch_Y_X = value & customfile_X = value & customkw_X = value ) &
[searchname = value & searchdescription = value & savesearch = true] &
[showonlinearchive = value]
```

Some *portalDI* rules for queries

The following rules, in broad strokes, govern *portalDI* queries. For complete details, please refer to Appendix C and Appendix D.

- Blank spaces

  A blank space between strings will be converted to %20, which then, becomes insignificant.
  Examples: `showvols=` returns the same output as `showvols =` and, `KV Photos` resolves to `KV%20Photos`, with both returning the same information.

  Other forms of blank space (horizontal tabs, Line feed or New line, and Carriage return) are also ignored when they occur before or after JSON structural characters ([ ], { }, colons, and commas).

- Reserved characters:
  “;” | “/” | “\” | “?” | “:” | “@” | “&” | “=” | “+” | “$” | “,” | “[” | “]”

  Each character string, above has special meaning and should be replaced by the corresponding “%” escape sequences when needed as a common character. The resulting string can then be used in assembling a URI.

  See RFC 4627 (Appendix D) for details and alternatives.

- Multiple query attributes and order

  Use `&` to separate assembled query attributes. Order doesn’t matter. The query `portalDI?fileid=1234&action=addbasket` produces the same results as the query `portalDI?action=addbasket&fileid=1234.`

- Case sensitivity

  Case sometimes matters, for example in UNIX path names it matters. Metadata *variables* are case insensitive, while metadata *values* are not, etc. Look for *case sensitivity* in the index of this volume or search electronically for *case* in Appendix C for full details.

- Only one `action` is allowed per query.

- Mistakes/Error handling

  In some cases an error is output. Otherwise, *toplevel* errors will be printed in the *venture* log (the *nativeadmin Logging, Database* page).

- Inquiry length

  Apache limits inquiries to 4000 characters. (Users with older Windows Internet Explorer browsers may have limitations of 2000 characters.)

Common argumentsDeprecated argumentsAs of WebNative Portal 4.0, the following are no longer in

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
display controls	
`debug= true | false | value`	When set to *true, portalDI* will send WebNative Suite debugging information to the *venture* log which can be viewed through *nativeadmin* GUI. The location of the actual log file varies according to platform. On Unix systems, it's in */var/adm/apple-talk/venture.log*; on Windows systems, *C:\Program Files\Xinet\FullPress\venture.log.*
`clientaddr = ipaddress`	The *portalDI* binary will use the address specified in the WebNative Suite event logs when populated. Otherwise, the WebNative Portal Server's address is added to the logs.
`fileid= fileid_num`	An alternative to specifying a file or directory using `path =`. Used to specify a WebNative Venture *fileid* number when you want to interact with a specific file or directory. You can determine the *fileid* by searching for it in 1) the source code for a browser page displaying the asset, or 2) the source code for a WebNative *Image Info* display of the asset. Scan the code for *"FILE_ID:"*.
`path=/path/to/some/file`	An alternative to specifying a file or directory using `fileid =`. Used to identify a file or folder by where it resides within the WebNative Suite file structure. Case sensitive. On both UNIX and Windows systems, use the forward slash or solidus (/) to indicate separate levels of the file system. On Windows, no need to indicate the drive (as in *C:*); Apache will determine it from configuration files. Take care when special characters are used in file names. However, as stated in RFC 4627 for JSON, any character may be escaped: "If the character is in the Basic Multilingual Plane (U+0000 through U+FFFF), then it may be represented as a six-character sequence: a reverse solidus, followed by the lowercase letter *u,* followed by four hexadecimal digits that encode the character's code point. The hexadecimal letters *A* through *F* can be upper or lowercase." There are also some two-character sequence escape representations of some frequently used characters. See *Appendix D* for details.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	**What it does**
portalDI actions	
`action = version`	When used, prints the current working version of the *portalDI* CGI. This action will also block other replies from the *portalDI* binary.
`action = showvols`	This is a bit like *toplevel*. When used, *showvols* provides a list of volumes and the current user's permissions for those volumes. This is the default action whenever the *portalDI* CGI is used without *path* or *fileid,* or whenever the *path* or *fileid* can't be found.
`action = showusersettings`	When invoked, provides basic information about the current user's assigned, non-volume-specific settings. These include *Language, Password Permissions, E-mail* address and *Group* information.
`action = showkywdperms`	When invoked, lists available keyword fields along with the user's permission settings for them.
`action = showbaskbtns`	When invoked, lists available basket plug-ins and information along with the user's permission settings for them.
`action = showiccusm`	When invoked, lists available ICC profiles and Unsharp Mask options available from the WebNative Suite.

`action = clearbasket`
Used in conjunction with:
`basketname =`
	When invoked, clears basket of its contents.Does not work if you specify a particular file for removal. Use `action = removebasket` for that.
	Adding a `basketname` will invoke the action on a saved basket for the current user.

`action = showbasket`
Used in conjunction with:
`basketname =` or `basketfile =`
`showkywds = true | false`
	When invoked, lists contents of the basket and their attributes.
	Adding a `basketname` will list the contents of a saved basket for the current user, or using `basketfile` with the full path to any user's basket will display that basket's contents.
	When combined with `showkywds = true` keyword values are included in output.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`action = streamfile`
Used in conjunction with:
`videoid = video_num`
`filetype = value`
`attach = true | false`
`fileid = or path =`
	When invoked, and used in conjunction with `fileid` or `path`, will stream the content to a user's Web browser. Content-type header is assigned when combined with `filetype`. Content-disposition set to attached when combined with `attach = true`. This will force a download action. Videos stored in the WebNative Suite database are streamed when combined with `videoid`. Video IDs are available in a the `fileinfo` output. An ID of *0* always corresponds to the original file.

`action = fileinfo`
Used in conjunction with:
`showkywds = true | false`
`fileid = or path =`
	When invoked, and used in conjunction with `fileid` or `path`, will output information for a file or folder. This option is only required when detailed information is needed about folders but not necessary about files. Adding `showkywds` will add more detailed information.

`action = showdirinfo`
Used in conjunction with:
`fileid = or path =`
	When invoked, and used in conjunction with `fileid` or `path`, will output information about the parent folder to the requested file/folder.

`action = addbasket`
Used in conjunction with:
`basketname =`
`fileid = or path =`
	When invoked, and used in conjunction with `fileid` or `path` will add the requested file/folder to the user's basket.
	Adding a `basketname` will invoke the action on a basket of the current user. If the basket does not exist it will be created.

`action = removebasket`
Used in conjunction with:
`basketname =`
`fileid = or path =`
	When invoked, and used in conjunction with `fileid` or `path` will remove the requested file/folder to the user's basket.
	Adding a `basketname` will invoke the action on a basket of the current user.

`action = submitkywd`
Used in conjunction with:
`keyword123 = value`
`fileid = or path =`
	When invoked, and used in conjunction with `comment = some comment` and a `fileid` or `path`, will change information in the file's *Show Details* field. When invoked, and used in conjunction with `keyword123 = value` and a `fileid` or `path`, will change keyword assignments.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`action = getimage`
 	
Used in conjunction with:
`pixels = num_pixels | [ bgcolor_r = background color red & bgcolor_g = background color green & bgcolor_b = background color blue ] | spreadnum = spread_num filetype = small | large | high | fpo archtype = sit | zip | uzip | maczip | umaczip archname = value packerrors = true | false fileid =` or `path =`
	When invoked, and used in conjunction with `fileid` or `path` and `filetype = small` or `large` or `high` or `fpo`, will send output to the user's Web browser. Small or large requests provide previews in JPEG or GIF formats depending on how they are stored in the WebNative Suite. High or FPO requests will be sent for download as file archives. Specifying `archtype = sit` or `zip` or `uzip` or `maczip` or `umaczip` will define the archive format. Additionally an alternative name for the archive can be requested with `archname = value`.
	
	Specifying `pixels = num pixels` will display a preview of the selected image at a specific size. Padding of the image is automatic. The backgound color of the pad is defined with `bgcolor_r`, `bgcolor_g`, and `bgcolor_b` using values *0* to *255*.

`action = filemgr`
 	
Used in conjunction with:
`filemgraction = mkdir | move | copy | delete | promote | version newpath = /path/to/destination/folder newname = value overwrite = true | false fileid =` or `filename =`
	When invoked, and used in conjunction with `fileid` or `filename` and `filemgraction = option,` will preform the requested action on the WebNative Suite server. The `delete` option only requires a `fileid` or `filename` be given for the action. The `mkdir` action will create a new folder on the WebNative Suite server in the `newpath` with the name specified as `newname`. Actions `move`, `copy` and `promote` require an originating `fileid` or `filename` and a destination `newpath` and `newname`. `Overwrite = true` will cause the action to replace any existing file or folder in the `newpath` with matching `newname`.

`action = annotations`
 	
Used in conjunction with:
`page = value grouponly = value fileid =` or `path =`
	When invoked, lists all active annotations for the specified page in the given file. A `page` number and `path` or `fileid` must be specified. Page indexing begins at zero.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`action = saveannotations`
 	
Used in conjunction with:
`fileid =` or `path =`
	When invoked, saves the supplied arguments as an annotation for the specified page in the given file. A page number and path must be specified. Annotation arguments must be supplied in a POST request.
	
	Annotations are specified with the following arguments:
	
	`save_annotations =` A comma-delimited list of annotation IDs. Each entry in the list should provide annotation details to save, as described below.
	
	NOTE: The *N* in each argument name represents the annotation ID of the annotation to which the argument corresponds. For a new annotation, use any unique string.
	
	• *N*`page =` Page to which the annotation applies
	• *N*`position`*X* `=` The *X* position of the left side of the annotation, in pixels.
	• *N*`position`*Y* `=` The *Y* position of the top side of the annotation, in pixels.
	• *N*`dimension`*H* `=` The height of the annotation, in pixels.
	• *N*`dimension`*W* = The width of the annotation, in pixels.
	• *N*`notes =` Notes for the annotation.
	• *N*`visible =` Indicates whether the annotation is visible. Can be `true` or `false`.
	• *N*`z`*Order* `=` The *z*-index of the annotation. Can be any integer.
	• *N*`zoom =` The zoom level at which the annotation was made. Can be any real number.
	• *N*`type =` The type of the annotation. Can be `Rect`, `Text`, `Stamp`, or `Sketch`.
	If the annotation is type `Text`, the following arguments should be supplied:
	
	• *N*`contents =` The body of the text annotation.
	• *N*`size =` The size of the text, in points.
	• *N*`font =` The font family used to display the annotation. Can be `Sans-Serif`, `Serif`, `Monospace`, `Cursive`, or `Fantasy`.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`action = saveannotations` (continued)	• *N*`color =` The color used to display the annotation. Should be an integer. Hex values in *0xN* notation are fine. • *N*`italic =` Indicates whether to use italics for the annotation. Can be `true` or `false`. • *N*`bold =` Indicates whether the annotation should be bold. Can be `true` or `false`. • *N*`justification =` How to justify the annotation. Can be `left`, `right`, or `center`. If the annotation is type `Rect`, the following arguments should be supplied: • *N*`width =` Border width for the annotation, in pixels. • *N*`style =` Border style for the annotation. Can be `solid`, `dashed`, `beveled`, or `inset`. • *N*`fill =` Fill color for the annotation. Should be an integer. Hex values in *0xN* notation are fine. • *N*`stroke =` Stroke color for the annotation. Should be an integer. Hex values in *0xN* notation are fine. If the annotation is type `Stamp`, the following arguments should be supplied: *N*`stamp =` Stamp index for the annotation. Should be an integer. Indices correspond as follows: `0` = *Approved* `1` = *Confidential* `2` = *Departmental* `3` = *Draft* `4` = *Experimental* `5` = *Expired* `6` = *Final* `7` = *For Comment* `8` = *For Public Release* `9` = *Not Approved* `10` = *Not For Public Release* `11` = *Sold* `12` = *Top Secret* If the annotation is type `Sketch`, the following arguments should be supplied: • *N*`width =` Stroke width for the annotation, in pixels. • *N*`color =` Stroke color for the annotation. Should be an integer. Hex values in *0xN* notation are fine. • *N*`path =` Stroke path for the annotation. Should be a colon-delimited list of *X,Y* pairs. For example, a five by five square with an origin at 1,1 could be encoded as `1,1:6,1:6,6:1,6:1,1.`

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`action = browse`
 	
Used in conjunction with:
 	
`[ itemsperpage = value page = value ] | [showfiles = false | [ true filesperpage = value filepage = value ] showdirs = false | [ true dirsperpage = value dirpage = value ]]`
 	
`resultsetid = value`
 	
`filename_flag_i = -1 | value`
`filename_sort_i = value`
`date_flag_n_i = -1 | value`
`date_sort_n_i = value`
`dbsearch_flag_n_i = -1 | value`
`dbsearch_sort_n_i = value`
`showkywds = true | false`
	When invoked with a folder's *fileid* or a folder's *full path,* will list the contents of the given folder. Some specific arguments:
	
	• `itemsperpage` = number of files and folders to list per request. When more items exist then defined for `itemserpage` the results will be paginated.
	• `page` = value of page number to be shown.
	
	Not to be used with above arguments:
	
	• `showfiles` = request specific output of files.
	• `filesperpage` = number of files to list per request. When more items exist then defined for `filesperpage` the results will be paginated.
	• `filepage` = value of file page number to be shown.
	• `showdirs` = request specific output of folders.
	• `dirsperpage` = number of folders to list per request. When more items exist then defined for `dirsperpage`, the results will be paginated.
	• `dirpage` = value of directory page number to be shown.
	
	Additional arguments:
	
	• `resultsetid` = pervious request result set from the Search Engine.
	• `filename_flag_i` = when submitted with `-1` allows the Search Engine to list results with a specified sort order.
	• `filename_sort_i` = sets Search Engine sort flag for `filename` and its sort order.
	• `date_flag_n_i` = when submitted with `-1` allows the Search Engine to list results with a specified sort order. `n` defines date type. `n` = `4` for *access date* and `n` = `3` for *modified date*.
	• `date_sort_n_i` = sets the Search Engine sort flag for dates and its sort order. `n` defines *date type*. `n` = `4` for access date and `n` = `3` for *modified date*.
	• `dbsearch_flag_n_i` = when submitted with `-1` allows the Search Engine to list results with a specified sort order. *n* defines *keyword field id*.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`action = browse` (continued)	• `dbsearch_sort_n_i`: Sets the Search Engine sort flag for keyword field n and its sort order. n defines keyword field id. • `showkywds` = when *true* retrieves *keyword data* for each file and or folder listed.

`action = navigator`
Used in conjunction with:
`fileid = value | path = value`
`showfiles = false | true`
	When invoked with a list of folders' *fileids* or folders' *full paths*, will output an ordered directory tree.
	Some specific arguments:
	`showfiles` = when `true`, will include files at each folder depth requested.
	Example: `http://172.16.0.20/webnative/portalDI?action=navigator&fileid=6047&fileid=6044&showfiles=true`

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`action = upload`
 	
Used in conjunction with:
 	
`path = /path/to/target/dir`
	To transfer a file to the Xinet server, a client should POST requests to a URI of the following form:
	
	`portalDI?action=upload&path=/path/to/target/directory`
	
	Note: portalDI expects arguments to be encoded in application/x-www-form-urlencoded. When uploading files, however, the client must encode the request arguments in multipart/form-data. Most developers should never have to worry about the details of the encoding format; they should be handled by whatever HTTP library or utility is being used. Should it become necessary to encode the MIME stream by hand, details can be found at the following address: http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.2
	
	The cURL command-line utility is a particularly easy way to transfer files to the Xinet server. A sample invocation is show below. Elements in italics should be replaced with values appropriate to the server.
	
	`curl --verbose -u "username:password" \`
	`-F 'overwrite=true' \`
	`-F 'action=upload' \`
	`-F 'keywordXX=cURLed' \`
	`-F 'dir=/server/path/to/target/dir' \`
	`-F 'filedata=@/Local/path/file/to/upload.jpg' \`
	`'http://hostname/webnative/portalDI?action=upload&path==/server/path/to/target/dir' \`
	
	Some specific arguments:
	
	• `dir = /path/to/target/directory`
	The target directory in which uploaded files should be placed on the Xinet server.
	
	• `newdir = new_directory_name`
	If this argument is present in the MIME stream, then portalDI will create a new directory in the target directory `dir`. Its name will be the same as the argument's value.
	
	• `newname = new_file_name`
	This value will be used to name the file, instead of the filename hints made by the browser in the MIME stream.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`action = upload` (continued)	• `clientaddr = ipaddress` Host address of the client where the file originated. • `overwrite = true | false` If this argument has a value of "true" or "1," and a file with the same name already exists on the server, then the version on the server will be overwritten by the version being uploaded. • `create_date = unix_date` If present, then portalDI will adjust the uploaded file's creation date to match the value. The value should be a valid unix-style timestamp. • `modification_date = unix_date` If present, then portalDI will adjust the uploaded file's modification date to match the value. The value should be a valid unix-style timestamp. • `dir_start = true | false` If given, portalDI will instruct dblogd to suspend upload event processing until the dir_end argument is given in a later transaction. This argument is useful when uploading directory contents, as uploads are often moved by triggers/actions from a dropbox to a more appropriate location. With dir_start/dir_end, triggers won't fire until the client has transferred every file. • `dir_end = true | false` When found, portalDI instructs dblogd to resume upload event processing. If dir_start was never provided by the client, it has no effect. • `filedata = value` The actual file data to be stored on the server. • `keywordN = true | false` If present, the value of this argument will be applied to the metadata field with ID "N" for the uploaded file. The user must have permission to access to the given field for the value to take effect.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`action= getorderimage`
 	
Used in conjunction with:
 	
`path = /path/to/target/dir`
	Some specific arguments:
	
	• `archiveformat = value`
	If the `webready` argument is not supplied, the resulting image will be stored in an archive file. This argument determines which format portalDI should use. If it's omitted, an appropriate default will be used. The following values are accepted:
	
	zip = Plain zip file
	unzip = Uncompressed zip file
	maczip = Zip file with finder info and resource fork
	umaczip = Uncompressed zip file with finder info and resource fork.
	
	• `backcolor = value`
	Sets the background color, when a Masked or Clipped image is output in a format that does not support masks or Clipping paths to the given color.
	
	• `blackpoint = true | false`
	Enables black point compensation during ICC color correction.
	
	• `clipping = value`
	Determines how clipping paths and masks are handled during conversion. The following values are accepted:
	
	[empty] = Ignore clipping paths and masks
	2 = Ignore only clipping paths
	3 = Causes all non-pixel data to be omitted from the output image.
	4 = Forces pixels to be clipped to a selected clipping path even if the output format supports clipping paths.
	5 = Same effect as empty string.
	
	• `colorplate = value`
	Select a color plate from the input image to render in greyscale. The following values are accepted:
	
	C = Cyan Plate
	M = Magenta Plate
	Y = Yellow Plate
	K = Black Plate

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`action = getorderimage` (continued)	• `colorspace = value` Process the output to a colorspace. The following values are accepted: *Grey* *RGB* *LAB* *CMYK* • `crop = (cropw, croph)` The argument specifies a rectangle on the source image, in pixels, numbered from (0,0), which is the pixel in the upper-left corner of the image. The coordinates cropx (horizontal) and cropy (vertical) provide the coordinates of the first pixel that will be placed in the upper-left corner of the output image. The pair, cropw (width) and croph (height) give the number of pixels of the original image to include in the cropped image. • `dpi = value` Output resolution to DPI (dots per inch) • `format = value` File format for output. Those formats marked with * all take an optional number following the fomat (e.g., -x eps83). This number sets a JPEG compression factor (normally off for EPS format). The JPEG compression factor is a number with a useful range between 5 and 95, where 5 creates a highly compressed image that doesn't look very much like the original, and 95 produces a less-compressed image, which will look very much (but not exactly) like the original. When JPEG output format is requested, the default compression factor is 75. Requesting EPS format with a compression value of 0 will cause the output to be HEX encoded (instead of the default BINARY); but not JPEG compressed.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`action = getorderimage` (continued)	• `format = value` (continued) The following values are accepted: *eps* - Encapsulated PostScript* *tif - Tag Image File Format* *gif - Composure's Graphics Interchange Format* *jpg* - Native JPEG, "lossy" compression format* *web* - GIF (if image is masked/clipped) or JPEG* *bmp - MS Windows BitMaP (aka DIB) format* *png - Portable Network Graphic Format* • `height = value` Scale to a specific height. • `includemeta = true | false` Write the file's XMP data to the output image, if the output format supports it. • `mergeclip = true | false` Forces pixels to be clipped to a selected clipping path even if the output format supports clipping paths. • `outputicc = true | false` Sets an output ICC profile pathname. This option enables ICC color correction if one of the following conditions is met (precedence in the order listed): *- An input profile has been supplied with the inputicc option* *- The image has an embedded profile* *- A default profile has been specified on the Xinet server for the image format and colorspace in the file* `/var/adm/appletalk/coloropts` *[UNIX] or* `C:\Program Files\Xinet\Full-Press\Admin\coloropts` *[WINDOWS]. Note: Xinet creates/updates the coloropts file whenever the administrator saves default ICC profiles in a Print Queue for various formats).* *- The input image is in LAB colorspace.* • `overprint = true | false` Causes any file that must be rendered (PDFs and vector EPS, for example) to be rendered in CMYK, regardless of what output colorspace was requested. This option guarantees that Overprints in the image are preserved.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`action = getorderimage` (continued)	• `preview = true | false` This option causes imageorder to search for a "preview" image before reading in the "full-scale" image. • `pctiff = true | false` If an EPS image has a PC-style TIFF preview, this option causes the TIFF to be read instead of the EPS. Alternatively, the argument can be 1 to 4 comma-separated numbers giving the parameters for the filter. The sharpening parameters include: The radius, in pixels, of the blurring Aperture, which must be greater than 0.5 (and has no default) A percentage strength of the sharpening values to apply (default is 100), where anything 0 or less disables the Unsharp Mask filter A lower threshold, in pixel intensity (0-255), below which no sharpening will occur (default is 0). An upper limit on the sharpening that will be applied to the image (default is 255, and any number out of the 1-255 range will be ignored). A value of 255 means no limit, a value of 10 means the maximum amount of change to a pixel to 10 (in component intensity value units, which are 0-255). Note that the Unsharp Mask parameters differ from Photoshop in the radius only: add one to a Photoshop Unsharp Mask Radius to get the equivalent sharpening. • `watermark = true | false` Turns on watermarking for the output image. This turns on automatically if the Xinet installation has not been licensed. By default, a watermark is tiled, pixel-for-pixel across and down the image as many times as it will fit. • `webready = true | false` Output is ready for display in a web page from an <img> HTML tag, rather than stored in an archive format. • `width = value` Scale to a specific width.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`action = presearch`	Returns the following arrays: *SAVEDSEARCHES_INFO*: Contains the list of saved Searches for this user *STAMPS_INFO*: Contains the stamps that can be searched on. *TYPEGROUPS_INFO*: Contains the list of file types used to generate the *File Type* Search filter. Optionally used with: • `searchid = value` Used to load the filters of a saved Search. • `sharesearches = true` Also loads saved group Searches, which are prefixed with an asterix • `delsavedsearh = true & searchid = value` Used to delete a saved Search.
`action = search`	For some examples of *portalDI Search* filters, see "Search filter examples" on page 39.
About `filter_logic`	Each *Search* filter can include a logic value, e.g., `searchall_logic_X = value` and `filename_-logic_X = value`. Possible values include: `1` = *AND* `2` = *OR* `3` = *XOR* If no logic value is given, then an *AND* is assumed, which is typically what most people will use.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
About `filter_flags`	`_flag` can take on the following values: `0` = *Is null* `1` = *Contains* `2` = *Start With (text) / After (dates) / Greater than (numbers)* `3` = *Ends With (text) / Before (dates) / Lesser than (numbers)* `4` = *Is Exactly (text) / On (dates)* `5` = *Has Any* `6` = *Has Each* `7` = *Contains the word* `8` = *Regular Expression* `9` = *Aggregate Has Each* `10` = *In the last (dates)* `11` = *In the next (dates)* `100` = *Is not null* `101` = *Does not contain* `102` = *Does not start with (text) / Not before (dates) / Not greater than (numbers)* `103` = *Does not end with (text) / Not after (dates) / Not lesser than (numbers)* `104` = *Is not exactly (text) / Not on (dates)* `105` = *Does not have any* `106` = *Does not have each* `107` = *Does not contain the word* `108` = *Not regular expression* A few `filter_flags` have special requirements.
About `_X` *and* `_Y_X`	The `_X` is the index of the filter. It is used to differenciate multiple filters of the same type in a *Search*. For example, if one wanted to search for *filename contains mango*, but also *filename ends with pdf*, one would construct the query as: `filename_0=mango & filename_flag_0=1 & filename_1=pdf & filename_flag_1=3` The `_Y` is different. Certain types of filters have an extra value that needs to be set. For example, single *keyword Searches* need to pass the *keyword id* of the *metadata Field* upon which to perform the *Search*. *Date* filters need to specify which of the 4 *Date Fields* to *Search* upon when the *Search* is executed. The `Y` value is used for these purposes.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
Search Filters:	
`subsearch_Y=X`	where `Y` is the *SearchID* of the previous executed Search, and `X` is the *ResultSetID* of the previous executed Search. This filter reques a `subsearch_logic_Y` value. Subsearching is meant to be used with the WebNative Portal *Quick Search* filter functionality. Essentially, you provide the Search Engine with the *SearchID* and *ResultSetID* of a previously-executed Search filter, and then add other Search filters to the Search. The Search Engine will use the previously cached Search results and apply the new filters to those results. Cached results are refreshed automatically when needed.

`searchall_X`
`searchall_flag_X`
`searchall_logic_X`
	Searches file name, Comments, text content, Annotations and text-based metadata fields assigned to the user's template. An optional `searchall_flag_X` value can be passed. If missing, a value of `1` (*contains*) is assumed.

`filename_X`
`filename_flag_X`
`filename_logic_X`
	Adds a file name filter. Requires a `filename_flag_X` value to be passed as well.

`filetype_X`
`filetype_flag_X`
`filetype_logic_X`
	Adds a file type filter. The value of this parameter corresponds to the *TypeGroupID* column of the *searchtypegroups* table.
	A `filetype_flag_X` should always be `4` (*Is Exactly*). Even if set to another value, it will be changed to `4`. Still, it is important to provide a value because the absence of a `filtype_flag_X` value tells the *Search Engine* that it has encountered a legacy file type *Search,* which won't work with the new values. So, always pass `filetype_flag_X=4` along with `filetype_X` filters.

`date_Y_X`
`date_flag_Y_X`
`date_logic_Y_X`
	Adds a date filter. Requires a `date_flag_Y_X` value to be passed as well. The value of `Y` determines which date field is searched:
	`1` = *Create date*
	`2` = *Backup date*
	`3` = *Modify date*
	`Anything else` = *Access date*
	The value of this parameter should be in UNIX timestamp format.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`comment_X`
`comment_flag_x`
`comment_logic_X`
	Adds a *Finder Comment* filter. Requires a `comment_-flag_X` value to be passed as well.

`searchallkeyword_X`
`searchallkeyword_flag_X`
`searchallkeyword_logic_X`
	Searches all text-based metadata Fields assigned to the user's template but not necessarily indexed in the *wnvfti* indexes. Requires a `searchallkeyword_flag_X` value to be passed as well.

`searchallfti_X`
`searchallfti_flag_X`
`searchallfti_logic_X`
	Searches all text-based metadata Fields assigned to the user's template and indexed in *wnvfti* indexes. Requires a `searchallfti_flag_X` value to be passed as well.

`dbsearch_keyword_Y_X`
`dbsearch_flag_Y_X`
`dbsearch_logic_Y_X`
	Adds a filter for *Keyword ID* `Y`. Requires a `dbsearch_flag_Y_X` value to be passed as well.

`annotations_X`
`annotations_flag_X`
`annotations_logic_X`
	Adds a filter for *Annotations* Notes and Comments. Requires a `annotations_flag_X` value to be passed as well.

`highresinfo_Y_X`
`highresinfo_flag_Y_X`
`highresinfo_logic_Y_X`
	Adds a *highresinfo* filter, where `Y` is:
	`1` for *width*
	`2` for *height*
	`3` for *resolution*
	`4` for *colorspace*
	`5` for *width* OR *height*
	`6` for *width* AND *height*
	`7` for *ICC profile*
	Requires a `highresinfo_flag_Y_X` value to be passed as well. For *colorspace*, the value can be:
	`1` for *mask*
	`2` for *1-bit*
	`4` for *grayscale*
	`8` for *indexed colors*
	`16` for *RGB colors*
	`32` for *CMYK colors*
	`64` for *Lab colors*
	`128` for *RGB high*
	You can add `1` to `4`, `16`, `32` and `64` to combine a *colorspace* and the *has mask* option.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`videoinfo_Y_X`
`videoinfo_flag_Y_X`
`videoinfo_logic_Y_X`
	Adds a *videoinfo* filter, where `Y` is:
	`2` for *video format*
	`3` for *codec name*
	`4` for *width*
	`5` for *height*
	`6` for *duration*
	`7` for *bitrate*
	`8` for *width* OR *height*
	`9` for *width* AND *height*
	Requires a `video_flag_Y_X` value to be passed as well.

`filecontent_X`
`filecontent_flag_X`
`filecontent_logic_X`
	Adds a *file content* filter. Requires a `filecontent_-flag_X` value to be passed as well.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`event_Y_X`
`event_flag_Y_X`
`event_logic_Y_X`
	Adds an event filter, where `Y` is:
	`3` for *CREATE*
	`4` for *DELETE*
	`5` for *RENAME*
	`6` for *READ*
	`7` for *WRITE*
	`8` for *COPY*
	`9` for *MKDIR*
	`10` for *RMDIR*
	`11` for *ADDCMT*
	`12` for *SETPARAM*
	`13` for *FPO*
	`14` for *WEBIMAGE*
	`15` for *QUARK*
	`16` for *PDF*
	`18` for *DOWNHIRES*
	`19` for *DOWNFPO*
	`20` for *DOWNPREV*
	`21` for *UPLOAD*
	`22` for *IMAGEORDER*
	`23` for *BACKEDUP*
	`24` for *ONLINE*
	`25` for *MEDIACHG*
	`26` for *TAPEDE*
	`27` for *PRINTED*
	`28` for *ARCHIVED*
	`29` for *DOSYNC*
	`30` for *TRIGGER*
	`31` for *METACHG*
	`32` for *WNA*
	`33` for *IMGREPL*
	`34` for *VIDEO*
	`35` for *DOWNVIDEO*
	`36` for *VIDVIEWED*
	`37` for *ANNOTNEW*
	`38` for *ANNOTEDIT*
	`39` for *ANNOTVIEW*
	`40` for *SETMETA*
	`41` for *DOXMPSYN*
	`43` for *ASSETSTART*
	`44` for *ASSETEXPIRED*
	`45` for *ASSETLOCKED*
	`46` for *ASSETUNLOCKE*D
	The value should be in UNIX timestamp format.
	Requires an `event_flag_Y_X` value to be passed as well.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does

`stamps_X`
`stamps_flag_X`
`stamps_logic_X`
	Adds an Annotation stamp filter with the `value` being the stamp's text. Requires a `stamps_flag_X` value to be passed as well.

`filedir_X` `filedir_logic_X`	Adds a file/folder filter. A value of `0` for files only, or `1` for folders only.

`filesize_X`
`filesize_flag_X`
`filesize_logic_X`
	Adds a file size filter. Requires a `filesize_flag_X` value to be passed as well.

`assettimer_X`
`assettimer_flag_X`
`assettimer_logic_X`
	Adds a filter for Asset Timer status. For `assettimer_X`, possible values include:
	`1` = *Available*
	`2` = *Unavailable*
	`3` = *Unclassified*
	`assettimer_flag_X` should be set to `4`. Even if set to another value, it will be changed to `4`.
	`assettimer_logic_X` is optional. Defaults to `1` (*AND*).

`quicksearch_Y_X`	Runs a Quick Search with Quick Search setting `Y`. The *searchenginesettings* table should contain a `quicksearchfield_Y` and `quicksearchtype_Y` value. Otherwise the Quick Search defaults to *searchall contains*.
`customfile_X`	Adds a custom MySQL *file* table filter. No validity check is done on the content of the custom filter.
`customkw_X`	Adds a custom MySQL *keyword1* table filter. No validity check is done on the content of the custom filter.

`searchname`
`searchdescription`
`savesearch`
	• Set the name of the saved *Search*.
	• Set the description of the saved *Search*.
	• Save the *Search* being executed.

Table 3-1 WebNative *portalDI* CGI arguments

Argument	What it does
`showonlinearchive`	Set the online/archived status of files that should be returned. The `value` should be a number, using the following value bitmap: `1` Use *AND* instead of *OR* between *Online* and *Archived* status `2` *Online = 0* (excludes *4*) `4` *Online = 1* `8` *Archived = 0* (exclued *16, 32,* and *64*) `16` *Archived = 1* `32` *Nearline = 1* `64` *Offline* = 1 For example, if you wanted to return files that were both *Online* and *Archived Nearline* (but not *Archived Offline*), you would use a value of `37` (`1` + `4` + `32`).
Additional file information	
`showkywds = true | false`	When *true,* includes keyword values with output of files and folders in both listings and detailed output.

`action = annotations`
Used in conjunction with:
`page = value`
`grouponly = value`
`fileid = or path =`
	When invoked, lists all active annotations for the specified page in the given file. A `page` number and `path` or `fileid` must be specified. Page indexing begins at zero.

use:

`showastxt = true	false`
`showlinks`, `showhistory` and `showversions` arguments, previously used with `action = fileinfo`
The following, used in conjunction with `action=search`

```
searchaction = search | more | less | clear
pathtype = value
pathtypelogic = value
selectedvol = value
selectedvollogic = value
comment = value
commentlogic = value
commenthasany = value
filenamelogic_i = value
filetypelogic_i = value
date_i = value
datelogic_i = value
dbsearchalllogic = value
custquery = value
dbsearchkeyword123_i = value
dbsearchlogic123_i = value
dbsearchflag123_i = value
dbsearchtype123_i = value
maxmatches = value
maxtotalmatches = value
pathsearchtype_i = value
skipcount = value
rowoffset = value
rowoffsetcount = value
rowoffsetarray = value
dateon_i = value
datewhence_i = value
year_i = value
month_i = value
day_i = value
exactmatch = value
nocasematch = value
casematch = value
skiphidden = true | false
dbsearchallkeyword = value
dbsearchallwithfile = value
dbsearchallwithcomment = value
filecount=integer
```

The following, used in conjunction with `action=search` (*continued*)

```
filestart=integer
dircount=integer
dirstart=integer
joindirfile= true | false
treedepth=integer
showfiles = true | false
showlinks = true | false
showhistory = true | false
showversions = true | false
```

Search filter examples

These examples show examples of *portalDI Search* filters.

- [*Search All*] [*Contains*] [*mango*]

  ```
  http://127.0.0.1/webnative/portalDI?action=search&
  searchall_0=mango&searchall_flag_0=1
  ```

  In this simple example, *portalDI* passes the *Search* value in the *searchall* variable with index `0` and the *searchall* flag for that same index. Here, `searchall_flag_0=1` could be left out since the default flag value for a searchall is `1` (*contains*).

- [*Filename*] [*Starts With*] [*mango*]

  ```
  http://127.0.0.1/webnative/portalDI?action=search&
  filename_0=mango&filename_flag_0=2
  ```

  This example uses a *filename* filter and sets the flag to `2` for [*starts with*].

- [*Create Date*] [*Before*] [*2012-02-25 13:24:56*] [*And*] [*Filename*] [*Has Each*] [*mango pdf*]

  ```
  http://127.0.0.1/webnative/portalDI?action=search&
  date_1_0=1330194296&date_flag_1_0=3&filename_0=mango pdf&
  filename_flag_0=6&filename_logic_0=1
  ```

  A little more complex, this example uses two filters. The first one is on the *Create Date* where the value, `1330194296`, is the UNIX timestamp that represents *2012-02-25 13:24:56*. The second filter is on the filename and is a token *Search* looking for both *mango* and *pdf* to be part of the filename. Note that the `filename_logic_0=1` part could have been left out since the default logic value between filters is `1` (*and*).

- [*Field 128*] [*Is Greater Than*] [*0*]

  ```
  http://127.0.0.1/webnative/portalDI?action=search&
  keyword_128_0=0&keyword_flag_128_0=2&maxfiles=10&keyword_sort_128_0=-1
  ```

  This example shows a keyword *Search* on a metadata field for values that are higher than `0`, but with the results sorted in reverse order of those values (`keyword_sort_128_0=-1`) and it also

limits the results to the first 10 (`maxfiles=10`). Essentially this *Search* finds the files that have the 10 highest values for *keyword ID 128*.

Chapter 3

PHP Example

Retrieving information and assigning it to an array

These two examples, the first using GET, the second POST, show how to use *portalDI* via PHP to retrieve information and assign it to an array.

Using GET

```
$get_string = "action=showkywdperms";

$POST = "GET /webnative/portalDI?$get_string HTTP/1.0\r\n";
#must be HTTP/1.0; can't use HTTP/1.1

$POST .= "HOST:" . $host . ":" . $port . "\r\n"
   . "Authorization: Basic " . base64_encode("$username:$password") . "\r\n"

   #. "Content-type: application/x-www-form-urlencoded\r\n"
# Content-type only needed for post

   #. "Content-length: " . strlen($request) . "\r\n"
# Content-length only needed for post

   . "Connection: close\r\n\r\n";
# must always have two line endings at the end

$fp = @fsockopen($host,$port,$errno,$errstr,300);

# read buffer, excluding any headers that we don't need
if ($fp) {
   fwrite($fp, $POST . $request);
$unneeded = array('www-authenticate','http','server','date', 'connection' );
   do {
$skipped = false;
$header = fgets ( $fp );
foreach ( $unneeded as $skip ) {
   if (($skipped = stristr ( $header, $skip )) == true) {break;}
}
if ($skipped === false) {
       $hds .= $header;
       # header ( $header );
}
} while ( strrpos ( $hds, "\r\n\r\n" ) == false );
    while (!feof($fp)) {
```

```
        $buffer .= fgets($fp);
    }
}
fclose($fp);

# translate results from json to php array
$results_array = json_decode($buffer, true);
```

Using POST

```
<?php
$url_string = "action=showusersettings";
$form_paramaters = "field1=value1&field2=value2&field3=value3";
$POST = "POST /webnative/portalDI?$url_string HTTP/1.0\r\n";
$POST .= "HOST:" . $_SESSION[CONFIG]['WNHOSTNAME'] . "\r\n"
   . "Authorization: Basic " . $_SESSION["USER"] . "\r\n"
   . "Content-type: application/x-www-form-urlencoded\r\n"
   . "Content-length: " . strlen($form_paramaters) . "\r\n"
   . "Connection: close\r\n\r\n";
$fp = @fsockopen($host, $port, $errno, $errstr, 300);
if ($fp)
{
   fwrite($fp, $POST . $form_paramaters);
   $unneeded = array('www-authenticate', 'http', 'server', 'date',
'connection' );
   do
   {
      $skipped = false;
      $header = fgets( $fp );
      foreach ( $unneeded as $skip )
      {
         if (($skipped = stristr( $header, $skip )) == true)
         {break;}
      }
      if ($skipped === false)
      {
         $hds .= $header;
      }
   } while ( strrpos( $hds, "\r\n\r\n" ) == false );
   while (!feof($fp))
   {
      $buffer .= fgets($fp);
   }
}
fclose($fp);

$results_array = json_decode($buffer, true);
?>
```

Further reading

- Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)," JSON.org, © The Internet Society, http://tools.ietf.org/html/rfc4627, 2006. (Included as *Appendix D*.)
- Gregorio, Joe, "How to create a REST Protocol," *http://bitworking.org/news/How_to_create_a_REST_Protocol*, 2006.
- Robinson, D., Coar, K., "The Common Gateway Interface (CGI) Version 1.1," The Apache Software Foundation, © The Internet Society, 2004, *http://www.ietf.org/rfc/rfc3875*, October 2004. (Included as *Appendix C*.)
- Udell, Jon, "The Beauty of REST," *http://www.xml.com/pub/a/2004/03/17/udell.html,* 2004.
- Udell, Jon, "Tangled in Threads: The power of the URL-line," *http://207.22.26.166/bytecols/2001-08-15.html,* 2001.
- Xinet, *Xinet Guide to Development APIs*, Berkeley, CA, 2008, *http://www.xinet.com*.

Xinet End User Software License Agreement

THIS XINET END USER SOFTWARE LICENSE AGREEMENT ("AGREEMENT") IS THE LEGAL AGREEMENT THAT GOVERNS YOUR USE OF THE ASSET MANAGEMENT SOFTWARE MADE AVAILABLE BY XINET INC. (TOGETHER WITH ITS ACCOMPANYING DOCUMENTATION, THE "SOFTWARE"). THIS AGREEMENT IS BETWEEN YOU, THE CUSTOMER WHO HAS ACQUIRED THE SOFTWARE ("YOU"), AND XINET INC. ("XINET"). PLEASE READ THIS AGREEMENT CAREFULLY.

XINET IS ONLY WILLING TO PROVIDE THE SOFTWARE TO YOU ON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS AGREEMENT. YOU ACCEPT THIS AGREEMENT BY INSTALLING OR USING THE SOFTWARE OR CLICKING ON THE BUTTON INDICATING ACCEPTANCE OF THE AGREEMENT WHICH IS PRESENTED UPON SOFTWARE INSTALLATION. BY CLICKING ON THE BUTTON TO ACCEPT THIS AGREEMENT OR BY INSTALLING THE SOFTWARE, YOU REPRESENT AND WARRANT THAT YOU HAVE THE AUTHORITY TO ENTER INTO THIS AGREEMENT, PERSONALLY OR IF YOU HAVE NAMED A COMPANY AS CUSTOMER, ON BEHALF OF THE COMPANY NAMED AS CUSTOMER, AND TO BIND EITHER YOURSELF OR SUCH COMPANY TO THE TERMS OF THIS AGREEMENT.

IF YOU DID NOT ACQUIRE THE SOFTWARE FROM XINET OR FROM AN AUTHORIZED XINET INTEGRATOR OR A XINET AFFILIATE (EACH, AN "AUTHORIZED INTEGRATOR"), THEN YOU MAY NOT ENTER INTO THIS AGREEMENT OR USE THE SOFTWARE. NO OTHER PARTY HAS THE RIGHT TO TRANSFER A COPY OF THE SOFTWARE TO YOU.

IF YOU ARE UNWILLING TO ACCEPT THIS AGREEMENT, DO NOT USE THE SOFTWARE. IF YOU HAVE ALREADY PAID FOR THE SOFTWARE WITHOUT HAVING A PRIOR OPPORTUNITY TO REVIEW THIS AGREEMENT AND ARE NOW UNWILLING TO AGREE TO THESE TERMS, YOU MAY, WITHIN TEN (10) DAYS AFTER THE DATE ON WHICH YOU ACQUIRED THE SOFTWARE, RETURN IT TO XINET OR THE AUTHORIZED INTEGRATOR FROM WHOM YOU ACQUIRED IT, ALONG WITH ITS ORIGINAL PACKAGING AND PROOF-OF-PURCHASE, FOR A FULL REFUND.

NOTWITHSTANDING ANYTHING HEREIN TO THE CONTRARY, NO AUTHORIZED XINET INTEGRATOR ACTS AS AN AGENT OF XINET, AND NO SUCH PARTY MAY ENTER INTO ANY CONTRACTS ON BEHALF OF XINET. NO AUTHORIZED INTEGRATOR HAS THE AUTHORITY TO MODIFY THE TERMS OF THIS AGREEMENT.

1. Grant of License. The Software consists of (a) certain components that are designated by Xinet as server software ("Server Software") that permit you to manage creative asset files stored on the one or more internal Xinet servers for which you have purchased a license (each, a "Xinet Server"), (b) certain client-side plug-in and extension utility components that permit the number of your internal employees for whom you have purchased a use license concurrently to access and use the Server Software and the files stored on the Xinet Server ("Client Software"), and (c) certain client-side utility components that provide desktop computer users, which may be external to your organization, with limited access to, and use of, your Xinet Server via the internet, solely to enable such users to upload, download, and search the creative asset files stored on your Xinet Server ("Utility Software"). The Server Software, Client Software and the Utility Software are collectively referred to herein as the "Software". Depending on the specific set of Xinet Software components for which you have paid the corresponding license fees, certain components may not be available to you in your Software installation and are not covered by the licenses granted in this Agreement.

1.1 Server Software. Subject to the terms and conditions of this Agreement, Xinet grants you a non-exclusive, non-transferable, worldwide, revocable license, during the term of this Agreement, to (a) install and execute, in each case on the Xinet Server in your possession, one (1) copy of the Server Software for which you have paid the corresponding license fees, in executable code form solely in accordance with the Documentation and solely to establish and use, on an internal basis and by the number of concurrent users for whom you have paid the corresponding license fees, one or more volumes of creative asset files to be stored on your Xinet Server and (b) to make one (1) copy of the Server Software for which you have paid the corresponding license fees, solely for backup or archival purposes.

1.2 Client Software. Subject to the terms and conditions of this Agreement, Xinet grants you a non-exclusive, non-transferable, worldwide, revocable license, during the term of this Agreement, to make a reasonable number of copies of the Client Software and to install and execute such copies of the Client Software on end-user computers

in your possession, in each case solely in executable code form and in accordance with the Documentation, solely for your internal business purposes, and to enable the number of concurrent users for whom you have paid the corresponding license fees to access and use the Xinet Server and upload and download files from the Xinet Server.

1.3 Utility Software. Subject to the terms and conditions of this Agreement, Xinet grants you a non-exclusive, non-transferable, worldwide, revocable license, during the term of this Agreement, to (a) permit each third party to whom you are providing consulting services related to the management of creative asset files (each, a "Utility User") to install and execute the Utility Software, solely in executable code and solely (i) in accordance with the Documentation and (ii) to enable the Utility User to access, via the internet, the Xinet Server and upload and download files from the Xinet Server, in each case in connection with the consulting services provided by you. You will be solely responsible for each Utility User's use of the Utility Software and any services provided by you to any Utility User. Xinet and its suppliers make no warranty, and have no obligation to provide support or other services to any Utility User. You will defend and indemnify Xinet and its suppliers from any claims or liabilities arising from or related to any Utility User to whom you provide access to the Software. You must enter into transactions with each Utility User on your own account and not on behalf of Xinet.

2. Restrictions on Use. You acknowledge, and will notify each Utility User, that the Software and its structure, organization, and source code constitute valuable trade secrets of Xinet and its suppliers. Accordingly, you agree not to (a) modify, adapt, alter, translate, or create derivative works from the Software or any copy, in whole or in part; (b) merge the Software with other software; (c) except as expressly permitted in Section 1.3, sublicense, lease, rent, loan, or otherwise transfer the Software to any third party, (d) use the Software in any service bureau or time-sharing arrangement, (e) **REVERSE ENGINEER, DECOMPILE, DISASSEMBLE, OR OTHERWISE ATTEMPT TO DERIVE THE SOURCE CODE FOR THE SOFTWARE**, or (f) otherwise use or copy the Software except as expressly allowed under Section 1. You will not remove, alter, or obscure any proprietary rights notices (including copyright notices) of Xinet or its suppliers on the Software, and you shall reproduce the same proprietary rights notices (including copyright notices) on any copy of the Software as appears on the original. You will ensure that each copy of the Utility Software that is distributed or made available by you is conspicuously marked with Xinet's proprietary rights notices.

3. Delivery and Acceptance. The Software is licensed and not sold, and the nonexclusive license set forth in this Agreement is not a sale of the Software or any copy. Title to the media on which the Software is recorded will transfer to you upon the delivery to you of such media, and the Software shall be deemed irrevocably and unconditionally accepted upon such delivery. Title to the Software remains with Xinet.

4. Proprietary Rights. **THE SOFTWARE, AND ALL WORLDWIDE INTELLECTUAL PROPERTY RIGHTS THEREIN, ARE THE EXCLUSIVE PROPERTY OF XINET AND ITS SUPPLIERS.** Except as stated above, this Agreement does not grant you any rights to any patents, copyrights, trade secrets, trade names, trademarks (whether registered or unregistered), or any other rights, franchises, or license in respect of the Software. All rights in and to the Software not expressly granted to you in this Agreement are reserved by Xinet and its suppliers.

5. Term and Termination; Effects of Termination. The term of this Agreement will begin on the date you accept this Agreement as provided above and will continue indefinitely. You may terminate your license at any time by destroying all copies of the Software in your possession or control. Xinet will have the right to terminate the license(s) granted to you by this Agreement immediately upon notice to you if you fail to comply with any term or condition of this Agreement. Upon any termination of this Agreement and/or the licenses for any reason, any amounts owed to Xinet under this Agreement before such termination or expiration will be immediately due and payable, all license rights granted in this Agreement will immediately cease to exist, and you must promptly discontinue all use of the Software, erase all copies of the Software from your computers, and return to Xinet, at your expense, all copies of the Software on tangible media in your possession or control.

6. Limited Warranty.

6.1 Xinet warrants to you that, for a period of thirty (30) days after delivery to you ("Warranty Period") the Software, when used as permitted under this Agreement and in accordance with the instructions in the Documentation (including use on a computer hardware and operating system platform supported by Xinet), will be free from Errors. As used herein, "Error" means a reproducible failure of the Software to perform substantially in accordance with the Documentation provided to you together with the Software. Xinet does not warrant that your use of the Software will be completely error-free or uninterrupted. If you received the Software from an Authorized Integrator, you may be required to obtain services pursuant to the warranty in this Section 6.1 directly from your Authorized Integrator.

6.2 If you notify Xinet or the Authorized Integrator, as applicable, during the Warranty Period that the Software fails to comply with the limited warranty in Section 6.1 above, Xinet will, at its option and expense and as its sole obligation and your exclusive remedy for any breach of such warranty, use commercially reasonable efforts to correct any reproducible error in the Software reported by you in writing during the Warranty Period, or, if Xinet determines that it is unable to correct the error, Xinet may, at its option, refund to you or the Authorized Integrator, on your behalf, the license fees actually paid by you to Xinet or the Authorized Integrator, as applicable, in which case the License and your right to use the Software will be terminated. Any Error correction provided to you will not extend the original Warranty Period. Notwithstanding anything herein to the contrary, if you received the Software from an integrator, distributor, reseller, or other third party you shall only be entitled to make a warranty claim to, and receive warranty remedies from, such third party.

6.3 **XINET AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE. YOU UNDERSTAND THAT, EXCEPT FOR THE EXPRESS WARRANTY SET FORTH IN SECTION 6.1, XINET AND ITS SUPPLIERS MAKE NO WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, WITH RESPECT TO THE SOFTWARE, INCLUDING ANY WARRANTIES AS TO PERFORMANCE, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. EXCEPT FOR THE EXPRESS WARRANTY STATED IN SECTION 6.1, THE SOFTWARE IS PROVIDED "AS IS" WITH ALL FAULTS, AND THE ENTIRE RISK AS TO SATISFACTORY QUALITY, ACCURACY, AND EFFORT IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XINET OR ANY AUTHORIZED INTEGRATOR, AGENTS OR EMPLOYEES.**

6.4 Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

7. Limit of Liability.

7.1 **IN NO EVENT WILL XINET OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, OR DAMAGES FOR ANY LOST DATA OR LOST PROFITS, ARISING FROM OR RELATING TO THIS AGREEMENT, EVEN IF XINET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. XINET'S TOTAL CUMULATIVE LIABILITY IN CONNECTION WITH THIS AGREEMENT AND THE SOFTWARE, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL NOT EXCEED THE AMOUNT OF LICENSE FEES PAID TO XINET OR YOUR AUTHORIZED INTEGRATOR, AS APPLICABLE, HEREUNDER. YOU ACKNOWLEDGE THAT THE LICENSE FEES REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XINET WOULD NOT ENTER INTO THIS AGREEMENT WITHOUT THESE LIMITATIONS ON ITS LIABILITY. IN ADDITION, XINET DISCLAIMS ALL LIABILITY OF ANY KIND OF XINET'S SUPPLIERS.**

7.2 Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

8. Export. You acknowledge that the laws and regulations of the United States restrict the export and re-export of the Software. You agree that you will not export or re-export the Software or media in any form without the appropriate United States and foreign government approval.

9. Choice of Law. This Agreement will be governed by the laws of the State of California without regard to any conflict of laws provisions that would require the applications of the laws of a different jurisdiction. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement. Any action or proceeding arising from or relating to this Agreement may be brought in a federal court in the Northern District of California or in state court in Alameda County, California, and each party irrevocably submits to the jurisdiction and venue of any such court in any such action or proceeding.

10. U.S. Government Restricted Rights Legend. The Software is comprised of "commercial computer software" and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 and are provided to the Government (i) for acquisition by or on behalf of civilian agencies, consistent with the policy set forth in 48 C.F.R. 12.212 or (ii) for acquisition by or on behalf of units of the Department of Defense, consistent with the policies set forth in 48 C.F.R. 227.7202-1 and 227.7202-3.

11. Notices. All notices, consents, and approvals under this Agreement must be delivered in writing by courier, by electronic facsimile (fax), or by certified or registered mail, (postage prepaid and return receipt requested) to the other party, for Xinet at the address set out on the Xinet website or for you, at the address provided to Xinet when the Software was licensed by you, and such notice will be effective upon receipt or three (3) business days after being deposited in the mail as required above. You may change your address by giving notice of the new address to Xinet.

12. Assignment. You may not assign or transfer, by operation of law or otherwise, any of your rights under this Agreement (including your licenses with respect to the Software) to any third party without Xinet's prior written consent. Any attempted assignment or transfer in violation of the foregoing will be void.

13. Remedies. Except as provided in Sections 6 and 7, the parties' rights and remedies under this Agreement are cumulative. You acknowledge that the Software contains valuable trade secrets and proprietary information of Xinet, that any actual or threatened breach of Section 2 will constitute immediate, irreparable harm to Xinet for which monetary damages would be an inadequate remedy, and that injunctive relief is an appropriate remedy for such breach.

14. Third Party Software. Certain items of software that may be provided to Xinet and distributed by Xinet together with the Software are owned by third parties and/or are subject to "open source" or "free software" licenses ("Third Party Software"). The Third Party Software is not subject to the terms and conditions in this Agreement other than Section 6.3 and Section 7. Instead, each item of Third Party Software is licensed under the terms of the end-user license that accompanies such Third Party Software. Nothing in this Agreement limits your rights under, or grants you rights that supersede, the terms and conditions of any applicable end-user license for the Third Party Software. If required by any license for particular Third Party Software, Xinet provides the information to access such Third Party Software and its accompanying end-user license in the Documentation.

15. Third Party Beneficiary. Artifex Software Inc. is an express third-party beneficiary of this Agreement. Except as expressly set forth in this section, there are no intended third-party beneficiaries of this Agreement.

16. Waivers. All waivers must be in writing. Any waiver or failure to enforce any provision of this Agreement on one occasion will not be deemed a waiver of any other provision or of such provision on any other occasion.

17. Severability. If any provision of this Agreement is unenforceable, such provision will be changed and interpreted to accomplish the objectives of such provision to the greatest extent possible under applicable law and the remaining provisions will continue in full force and effect. Without limiting the generality of the foregoing, you agree that Section 7 will remain in effect notwithstanding the unenforceability of any provision in Section 6.

18. Construction. The headings of Sections of this Agreement are for convenience and are not to be used in interpreting this Agreement. As used in this Agreement, the word "including" means "including but not limited to".

19. Integration. You acknowledge that you have read this Agreement, understand it, and that by installing the Software you agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between Xinet and you, which supersedes any proposal or prior agreement, oral or written, and any other communications between Xinet and you relating to the subject matter of this Agreement. No variation of the terms of the Agreement or any different terms will be enforceable against Xinet unless Xinet gives its express consent, including an express waiver of the terms of this Agreement, in writing signed by an officer of Xinet.

Xinet Trademark Acknowledgements, Attributions, and Credit Notices

Xinet, the Xinet logo, WebNative, Picture Wrangler, FullPress, WebVolume, Queue Master, Color Verite, Server to the Macs, FPO Exporter, K-Talk, K-Time, K-FS, K-Spool, and K-AShare are either trademarks or registered trademarks of Xinet, registered in the United States and other countries.

• • • • •

Xinet acknowledges the following trademarks for company names, brands, and product names mentioned within its published materials:

Adobe, Adobe Acrobat, Adobe PDF and the Adobe PDF logo design, Adobe Premiere, Adobe Type Manager, Creative Suite, Distiller, Flash and the Flash logo design, FrameMaker, Freehand, GoLive, InDesign, Illustrator, Macromedia, PageMaker, Photoshop and the Photoshop logo design, PostScript and the PostScript logo design, XMP and the XMP logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apache and the Apache feather logo are trademarks of The Apache Software Foundation.

Apple, AppleScript, AppleShare, AppleTalk, Bonjour, Ethertalk, Extensions Manager, Final Cut, Finder, ImageWriter, LaserWriter, LocalTalk, Macintosh, Mac OS, QuickTime, Safari, TeachText, TrueType, Xsan, Xserve and other Apple related marks are either trademarks or registered trademarks of Apple Inc., registered in the United States and/or other countries.

Artifex is a registered trademark of Artifex Software Inc. in the United States and/or other countries.

Autodesk and AutoCAD are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affilates in the USA and/or other countries. DWG is the native file format for Autodesk's AutoCAD software and is a trademark of Autodesk, Inc.

FlashNet, FlashWeb, and SGL are trademarks of Software Generation Limited in the United Kingdom, United States and/or other countries.

Intel is a registered trademark of Intel Corporation in the United States and other countries.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Microsoft, Active Directory, Excel and the Microsoft Excel logo design, Internet Explorer and the Microsoft Internet Explorer logo design, Microsoft Office and the Office logo design, PowerPoint, Windows, Windows NT, Microsoft Windows Server, Windows XP, Windows 2000, Windows 2003 and the Windows logo design are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Mozilla, Firefox and the Firefox logo design, Camino, and the Camino logo design are registered trademarks of the Mozilla Foundation in the United States and other countries.

MultiAd and MultiAd Creator are registered trademarks of Multi-Ad Services, Inc., in the United States and other countries.

OpenEXR, Industrial Light & Magic and ILM are trademarks and service marks of Lucasfilm Ltd.; all associated intellectual property is protected by the laws of the United States and other countries. All rights reserved.

Quark, the Quark logo, QuarkXPress, QuarkXPress Passport, Quark Publishing System, XTensions, the XTensions logo, and other Quark related marks are trademarks or registered trademarks of Quark, Inc. and its affiliates in the U.S. and/or other countries.

RED®, RED CAMERA®, RED EPIC®, SCARLET™, RED ONE®, REDRAY™, MYSTERIUM®, MYSTERIUM X®, R3D®, REDCODE®, REDMOTE® and all other trademarks, service marks, trade names logos, and product names of RED used in Xinet product materials and documentation are trademarks or registered trademarks of Red.com, Inc. in the USA and other countries.

Red Hat, Red Hat Enterprise Linux, and the Red Hat "Shadowman" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

Symantec, NetBackup, Backup Exec, and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the United States and other countries.

Silicon Graphics, SGI and the SGI logo design, Altix, IRIX, Octane, Origin, Rackable Systems and the Rackable logo design, OpenGL and XFS are either registered trademarks or trademarks of Silicon Graphics International in the United States and/or other countries worldwide.

Sun, Sun Microsystems, the Sun logo, Java and all Java-based trademarks and logo designs, MySQL and the MySQL logo, NFS, Solaris and the Solaris logo are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and/or other countries.

SPARC is a registered trademark of SPARC International, Inc. in the United States and other countries.

TEIGHA, the TEIGHA logo, Open Design Alliance, and the Open Design Alliance logo are trademarks of the Open Design Alliance in the United States and/or other countries. DWG is the native and proprietary file format for AutoCAD® and a trademark of Autodesk, Inc. The Open Design Alliance is not associated with Autodesk.

Telestream and FlipFactory are registered trademarks of Telestream, Inc. in the United States and/or other countries.

UNIX, X/Open, Motif, OSF/1, and The Open Group are registered trademarks of The Open Group in the United States and other countries.

Zend and Zend Technologies, and/or the logos and icons associated with each of these names are trademarks of Zend Technologies Ltd. and may be registered in certain jurisdictions.

All other product names, trade names, brands, company names, registered trademarks, or unregistered trademarks which are referenced in materials published by Xinet are used for identification purposes only and are the property of their respective owners.

• • • • •

Depending on the Xinet software products licensed to you under the *Xinet End User Software License Agreement*, some or all of the copyright and warranty notices which follow may pertain to your installation.

— Portions of the software © Copyright 1999–2003 Apple Computer, Inc.

APPLE COMPUTER, INC. ("APPLE") MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THAT THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE REGARDING THE APPLE SOFTWARE. APPLE DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE APPLE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE APPLE SOFTWARE

IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. IN NO EVENT WILL APPLE, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE APPLE SOFTWARE EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. APPLE'S LIABILITY TO YOU FOR ACTUAL DAMAGES FROM ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), WILL BE LIMITED TO $50.

— Portions Copyright © 2000 Artifex Software Inc. All rights reserved. Portions Copyright © 1998, 2000 Aladdin Enterprises. All rights reserved. Portions Copyright © Soft Horizons. All rights reserved.

— Portions of this software include software whose copyright is owned by, or licensed from, MySQL AB and Sun Microsystems. © Copyright 1995-2008 MySQL AB, © Copyright 2000, MySQL Finland AB, © Copyright 2008-2009 Sun Microsystems, Inc.

— Portions of this software use OpenEXR, Copyright © 2002-2012, Industrial Light & Magic, a division of Lucasfilm Entertainment Company Ltd. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Industrial Light & Magic nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— QUARK IS NOT THE MANUFACTURER OF THIRD-PARTY SOFTWARE OR OTHER THIRD-PARTY HARDWARE (HEREINAFTER "THIRD-PARTY-PRODUCTS") AND SUCH THIRD-PARTY PRODUCTS HAVE NOT BEEN CREATED, REVIEWED, OR TESTED BY QUARK, THE QUARK AFFILIATED COMPANIES OR THEIR LICENSORS. (QUARK AFFILIATED COMPANIES SHALL MEAN ANY PERSON, BRANCH, OR ENTITY CONTROLLING, CONTROLLED BY OR UNDER COMMON CONTROL WITH QUARK OR ITS PARENT OR A MAJORITY OF THE QUARK SHAREHOLDERS, WHETHER NOW EXISTING OR FORMED IN THE FUTURE, TOGETHER WITH ANY PERSON, BRANCH, OR ENTITY WHICH MAY ACQUIRE SUCH STATUS IN THE FUTURE.) QUARK, THE QUARK AFFILIATED COMPANIES AND/OR THEIR LICENSORS MAKE NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE QUARK PRODUCTS/SERVICES AND/OR THIRD-PARTY PRODUCTS/SERVICES, THEIR MERCHANTABILITY, OR THEIR FITNESS FOR A PARTICULAR PURPOSE. QUARK, THE QUARK AFFILIATED COMPANIES AND THEIR LICENSORS DISCLAIM ALL WARRANTIES RELATING TO THE QUARK PRODUCTS/SERVICES AND ANY THIRD-PARTY PRODUCTS/SERVICES. ALL OTHER WARRANTIES AND CONDITIONS, WHETHER EXPRESS, IMPLIED, OR COLLATERAL, AND WHETHER OR NOT, MADE BY DISTRIBUTORS, RETAILERS, XTENSIONS DEVELOPERS OR OTHER THIRD-PARTIES ARE DISCLAIMED BY QUARK, THE QUARK AFFILIATED COMPANIES AND THEIR LICENSORS, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF NON-INFRINGEMENT, COMPATIBILITY, OR THAT THE SOFTWARE IS ERROR-FREE OR THAT ERRORS CAN OR WILL BE CORRECTED. THIRD-PARTIES MAY PROVIDE LIMITED WARRANTIES AS TO THEIR OWN PRODUCTS AND/OR SERVICES, AND USERS MUST LOOK TO SAID THIRD-PARTIES FOR SUCH WARRANTIES, IF ANY. SOME JURISDICTIONS, STATES OR PROVINCES DO NOT ALLOW LIMITATIONS ON IMPLIED WARRANTIES, SO THE ABOVE LIMITATION MAY NOT APPLY TO PARTICULAR USERS. IN NO EVENT SHALL QUARK, THE QUARK AFFILIATED COMPANIES, AND/OR THEIR LICENSORS BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES, INCLUDING, BUT NOT LIMITED TO, ANY LOST PROFITS, LOST TIME, LOST SAVINGS, LOST DATA, LOST FEES, OR EXPENSES OF ANY KIND ARISING FROM INSTALLATION OR USE OF THE QUARK PRODUCTS/SERVICES, IN ANY MATTER, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY. IF, NOTWITHSTANDING THE FOREGOING, QUARK, THE QUARK AFFILIATED COMPANIES AND/OR THEIR LICENSORS ARE FOUND TO HAVE LIABILITY RELATING TO THE QUARK PRODUCTS/SERVICES OR THIRD-PARTY PRODUCTS/SERVICES, SUCH LIABILITY SHALL BE LIMITED TO THE AMOUNT PAID BY THE USER TO QUARK FOR THE SOFTWARE/SERVICES AT ISSUE (EXCLUDING THIRD-PARTY PRODUCTS/SERVICES), IF ANY, OR THE LOWEST AMOUNT UNDER APPLICABLE LAW, WHICHEVER IS LESS. THESE LIMITATIONS WILL APPLY EVEN IF QUARK, THE QUARK AFFILIATED COMPANIES, THEIR LICENSORS AND/OR THEIR AGENTS HAVE BEEN ADVISED OF SUCH POSSIBLE DAMAGES. SOME JURISDICTIONS, STATES OR PROVINCES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION OR EXCLUSION MAY NOT APPLY. ALL OTHER LIMITATIONS PROVIDED UNDER APPLICABLE LAW,

INCLUDING STATUTES OF LIMITATION, SHALL CONTINUE TO APPLY. IN THE EVENT ANY OF THESE PROVISIONS ARE OR BECOME UNENFORCEABLE UNDER APPLICABLE LAW, SUCH PROVISION SHALL BE MODIFIED OR LIMITED IN ITS EFFECT TO THE EXTENT NECESSARY TO CAUSE IT TO BE ENFORCEABLE. USE OF THE QUARK PRODUCTS IS SUBJECT TO THE TERMS OF THE END USER LICENSE AGREEMENT OR OTHER APPLICABLE AGREEMENTS FOR SUCH PRODUCT/SERVICE. IN THE EVENT OF A CONFLICT BETWEEN SUCH AGREEMENTS AND THESE PROVISIONS THE RELEVANT AGREEMENTS SHALL CONTROL.

— Tag Image File Format (TIFF) Library © Copyright 1988-1997 Sam Leffler. © Copyright 1991-1997 Silicon Graphics, Inc. Permission to use, copy, modify, distribute and sell the Tag Image File Format Library software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics. THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

— XMP portions Copyright © 1999-2008, Adobe Systems Incorporated. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Adobe Systems Incorporated, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— This software is based in part on the work of the Independent JPEG Group.

— Portions of the software, including *ndbm(3)*, Copyright © 1983, Regents of the University of California. All rights reserved. The Berkeley software License Agreement specifies the terms and conditions for redistribution.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Regents of the University of California nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS OF THE UNIVERSITY OF CALIFORNIA "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

— This software includes portions of the Darwin open source project.

— Portions of the K-Time utility software for Mac OS 9 clients were developed by The University of Melbourne and redistributed by Xinet as a component of Xinet's K-Talk AppleTalk stack. Copyright © 1992, The University of Melbourne.

— The *curl* and *libcurl* portions of this software © Copyright 2000, Daniel Stenberg, daniel@haxx.se. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of *curl* and *libcurl* and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

— Portions of this software © Copyright 2003 Joe Penn.

— PHP and the PHP logo designs are copyright © 2001-2009 The PHP Group. All rights reserved. Portions of this software © Copyright 2001–2003 PHP Application Tools. Portions of this software © Copyright 1999–2009 The PHP Group. This product includes PHP software, freely available from http://www.php.net/software/. Use of the PHP software provided in the Xinet WebNative Portal software distribution is governed by *The PHP License, version 3.01*, below:

The PHP License, version 3.01

Copyright (c) 1999 - 2009 The PHP Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact group@php.net.

4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo"

5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.

6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes PHP software, freely available from <http://www.php.net/software/>".

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PHP DEVELOPMENT TEAM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the PHP Group. The PHP Group can be contacted via Email at group@php.net. For more information on the PHP Group and the PHP project, please see <http://www.php.net>. PHP includes the Zend Engine, freely available at <http://www.zend.com>.

— Portions of this software make use of the Zend Optimizer, Copyright © 1998-2009, by Zend Technologies Ltd. Use of the Zend software provided in the Xinet WebNative Portal software distribution is governed by the license agreement below:

ZEND OPTIMIZER - ZEND LICENSE AGREEMENT

ZEND TECHNOLOGIES LTD. (“ZEND”) SOFTWARE LICENSE AGREEMENT (“AGREEMENT”)

IMPORTANT: READ THESE TERMS CAREFULLY BEFORE INSTALLING THE SOFTWARE KNOWN AS THE “ZEND OPTIMIZER,” AS INSTALLED BY THIS INSTALLATION PROCESS, IN MACHINE-EXECUTABLE FORM ONLY, AND ANY RELATED DOCUMENTATION (COLLECTIVELY, THE “SOFTWARE”) BY INSTALLING, OR OTHERWISE USING THIS SOFTWARE, YOU (THE “LICENSEE”) ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, AND THAT YOU AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU ARE NOT AN AUTHORIZED USER OF THE SOFTWARE AND IT IS YOUR RESPONSIBILITY TO EXIT THIS INSTALLATION PROGRAM WITHOUT INSTALLING THE SOFTWARE, OR TO DELETE THE SOFTWARE FROM YOUR COMPUTER.

1. License. Subject to the terms and conditions of this Agreement, including, without limitation, Section 2 hereof, Zend hereby grants to Licensee, during the Term (as defined below), a limited, a non-exclusive license (the “License”) to: (i) install and operate the Software on a computer or a computer network owned or operated by Licensee; (ii) make copies of the Software; and (iii) sublicense and distribute a limited, non-exclusive sublicense to install, use and sublicense such copies of the Software, provided that any sub-license granted hereunder shall be subject to the limitations and restrictions set forth in this Agreement.

2. Restrictions. Except as otherwise expressly set forth herein, Licensee or any of its sub-licensees shall not: (a) translate or decompile, or create or attempt to create, by reverse engineering or otherwise, the source code form from the object code supplied hereunder; (b) modify, adapt, translate or create a derivative work from the Software; (c) remove any proprietary notices, labels, or marks on the Software.

3. Termination. This Agreement and the License hereunder shall be in effect from and after the date Licensee installs the Software on a computer in accordance with the terms and conditions hereof and shall continue perpetually unless terminated in accordance with this Section 3. This Agreement shall be automatically terminated upon any breach by Licensee of any term or condition of this Agreement. Such period shall be referred to herein as the “Term”. Within five (5) business days of any such termination, Licensee shall return the Software to Zend (or, at Zend's sole discretion and only at Zend's direction, destroy the Software and certify in writing to Zend that said Software has been destroyed). Upon return of the Software or upon receipt of notice of the destruction of the Software, as appropriate, this Agreement shall terminate and Zend shall have no further obligations to Licensee. Articles 2, 4, 5, 6, 7, 8 and 10 hereof shall survive the expiration or termination of this Agreement for any reason.

4. Intellectual Property Rights. Licensee hereby acknowledges and agrees that Zend or its licensors own and retain all rights, title, and interest in and to the Software, regardless of the form or media in or on which the original or other copies may subsequently exist including, without limitation, all copyrights, trademarks, patents and trade secret rights inherent therein or appurtenant thereto. This Agreement shall not constitute a sale of the Software and no title or proprietary rights to the Software are transferred to the Licensee hereby. Licensee acknowledges that the Software is a unique, confidential and valuable asset and trade secret of Zend or its licensors, and Zend or its licensors shall have the right to obtain all equitable and legal redress which may be available to it for the breach or threatened breach of this Agreement including, without limitation, injunctive relief.

5. Warranty; Disclaimer. THE SOFTWARE IS BEING LICENSED HEREUNDER WITH NO WARRANTY WHATSOEVER. LICENSEE ACKNOWLEDGES THAT ITS USE OF THE SOFTWARE IS AT ITS OWN RISK. THE SOFTWARE IS PROVIDED SOLELY ON AN “AS-IS” BASIS. ZEND AND ITS LICENSORS MAKE, AND LICENSEE RECEIVES, NO WARRANTIES, EXPRESS, IMPLIED, OR OTHERWISE. ZEND EXPRESSLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND/OR NONINFRINGEMENT. ZEND DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE SHALL BE OPERABLE, UNINTERRUPTED OR ERROR FREE OR THAT IT WILL FUNCTION OR OPERATE IN CONJUNCTION WITH ANY OTHER PRODUCT.

6. No Liability. IN NO EVENT SHALL ZEND OR ITS LICENSORS BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, EXEMPLARY, SPECIAL, INDIRECT, INCIDENTAL OR PUNITIVE DAMAGES OR LOST PROFITS (EVEN IF THEY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES) ARISING FROM OR RELATING TO THIS AGREEMENT OR THE SUBJECT MATTER HEREOF, INCLUDING WITHOUT LIMITATION, DAMAGES ARISING FROM LOSS OF DATA, OR ANY OTHER DAMAGES, HOWEVER CAUSED, AND UNDER ANY THEORY OF LIABILITY. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, AND IN SUCH CASE AND ONLY TO THE EXTENT PROHIBITED BY APPLICABLE LAW, THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO LICENSEE. WITHOUT LIMITATION OF THE FOREGOING, UNDER NO CIRCUMSTANCES SHALL LICENSOR'S TOTAL AGGREGATE LIABILITY TO LICENSEE AND ANY OF ITS SUB-LICENSEES (REGARDLESS OF THE LENGTH OF THE TRAIL OF SUBLICENSES) FOR ANY AND ALL CLAIMS ARISING HEREUNDER, INCLUDING WITHOUT LIMITATION ARISING OUT OF YOUR USE OF THE SOFTWARE, EXCEED THE SUM OF $100.

7. Indemnity. Licensee will, at its own expense, defend any action brought by a third party against Zend to the extent that such action is based on a claim arising from or relating to: (a) Licensee's use of the Software, (ii) any distribution of the Software by Licensee or by any of or by any sub-licensee, regardless of privity of

contract and regardless of the length of the trail of sublicenses, (iii) any claims based upon warranties, guarantees or representations made by Licensee or any of its employees, agents or sub-licensees; or (iv) any use of the Software by any of the foregoing sub-licensees. Zend shall have the exclusive right to control such defense. In no event shall Licensee settle any such claim, lawsuit or proceeding without Zend's prior written approval.

8. U.s. Government restricted rights. The Software provided hereunder is a “commercial item,” as that term is defined in 48 C.F.R. 2.101, consisting of “commercial computer software” and “commercial computer software documentation” as such terms are used in 48 C.F.R. 12.212. Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4, the Software made available to the United States of America, its agencies and/or instrumentalities, is provided with only those rights set forth in this Agreement. Use, duplication or disclosure of the Software by the government is subject to the restrictions as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software-Restricted Rights clause at 48 C.F.R. 52.227-19, as amended, or any successor regulations thereto.

9. Export and Import Restrictions. Licensee shall be solely responsible for obtaining any required regulatory approvals for exportation, importation and/or distribution of the Software.

10. Miscellaneous. This Agreement is made in and shall be governed by the laws of the State of Israel, excluding choice of law principles. Venue for all proceedings shall be Tel Aviv, Israel. Notwithstanding the foregoing, Zend shall have the right to apply to any court of competent jurisdiction for injunctive or other relief. The United Nations Convention for the International Sale of Goods shall not apply. There is no relationship of agency, partnership, joint venture, employment or franchise between the parties. Licensee or its sub-licensees do not have the authority to bind Zend or to incur any obligation its behalf or to represent itself as Zend's agent or in any way that might result in confusion as to the fact that the parties are separate and distinct entities. The section headings therein are provided for convenience only and have no substantive effect on the construction of this Agreement. If any provision of this Agreement is held to be unenforceable, this Agreement shall be construed without such provision. The failure by Zend to exercise any right hereunder shall not operate as a waiver of Zend's right to exercise such right or any other right in the future. This Agreement may be amended only by a written document executed by a duly authorized representative of each of the parties. This Agreement constitutes the complete and exclusive statement of the terms and agreement between Zend and Licensee and supersedes all prior representations, understandings and communications, oral or written, between the parties with respect thereto, including memoranda of agreement. In the event legal action is taken by Zend or its licensors to enforce any provision of this Agreement, all costs and expenses, including reasonable attorneys' fees and expenses, incurred by Zend or its licensors shall be paid by Licensee, in addition to other damages to which Zend or its licensors may be lawfully entitled. Licensee agrees that it will not refer to the Software or to the existence of this Agreement nor will it use Zend's name in any press releases, advertising, marketing or other materials without Zend's advanced written consent in each instance. Any notice provided by Licensee to Zend pursuant to this Agreement shall be in writing to Zend Technologies Ltd. at P.O. Box 3619, Ramat Gan 52136, Israel, and shall be deemed given (i) if by hand delivery, upon receipt thereof; (ii) if mailed, three (3) days after deposit in the U.S. mails, postage prepaid, certified mail, return receipt requested; or (iii) if by next day delivery service, upon such delivery. Any notice provided by Zend to Licensee pursuant to this Agreement shall be sent to the e-mail or mailing address provided by Licensee upon registering on Zend's Web site, and shall be deemed given (i) if by hand delivery, upon receipt thereof; (ii) if mailed, three (3) days after deposit in the U.S. mails, postage prepaid, certified mail, return receipt requested; or (iii) if by next day delivery service, upon such delivery. Either party may change its address by giving written notice to the other party.

— Portions of this software use libraries from the Open Design Alliance: “Teigha® for .dwg files” © 2010 Open Design Alliance. All rights reserved. As part of its Open Design Alliance Membership Agreement, Xinet is furnished with Alliance Tools and Libraries. Xinet must provide notice as follows regarding applications it has developed with these Alliance Tools:

“The Tools are provided to Member ‘AS IS’ AND WITH ALL DEFECTS AND ERRORS. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, THE ALLIANCE HEREBY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IMPLIED WARRANTY ARISING FROM ANY COURSE OF PERFORMANCE OR DEALING OR USUAGE OF TRADE, IMPLIED WARRANTY OF NONINFRINGEMENT OR IMPLIED WARRANTY OF QUIET ENJOYMENT. THE ALLIANCE IS NOT LIABLE FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT, EXEMPLARY OR OTHER SIMILAR DAMAGES ARISING FROM BREACH OF THIS AGREEMENT OR OF ANY WARRANTY CONTAINED HEREIN, WHETHER ARISING IN CONTRACT, TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY, EQUITY OR OTHERWISE, EVEN IF THE ALLIANCE WAS ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

— Portions of this software include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). Copyright © 1998–2008 The OpenSSL Project. All rights reserved. The OpenSSL

Toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).”

4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).”

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright © 1995–1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young’s, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: “This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).” The word “cryptographic” can be left out if the rouines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: “This product includes software written by Tim Hudson (tjh@cryptsoft.com)”

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

— Portions of this software use the Mesa 3-D graphics library: the main Mesa code and 'omesa' driver. To the extent that Mesa utilizes the OpenGL™ command syntax or state machine, it is being used with authorization from Silicon Graphics, Inc. (SGI). However, the author of the Mesa software does not possess an OpenGL license from SGI, and makes no claim that Mesa is in any way a compatible replacement for OpenGL or associated with SGI. The core Mesa library and 'omesa' driver are licensed according to the terms of the MIT license, as follows:

Copyright © 1999-2007 Brian Paul. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (The "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRES OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLIAM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

— Xinet distributes FlashNet backup and recovery software for installation on certain server platform OSs in conjunction with Xinet WebNative Archive software and/or other Xinet software products. FlashNet is developed by Software Generation Limited and use of FlashNet software is governed by the license agreement below:

FlashNet End User Software Licence Agreement

General Description

By order to preserve and protect its rights under applicable laws, SGL does not sell any rights in SGL Software. Rather SGL grants the right to use SGL software by means of a software licence. SGL specifically retains title to all SGL software. Important note: DO NOT install the software from the supplied CD, disk or file until you have carefully read this Agreement. BY INSTALLING THE SOFTWARE, (OR AGREEING FOR AN APPROVED INSTALLER TO DO SO), AND SUBSEQUENT COMMERCIAL USE OF THE SOFTWARE, YOU AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT. If you do not agree with the terms and conditions set forth in this Agreement, return the software and all documentation to your supplier.

Grant of a Licence

In consideration of payment of the licence fee which is part of the price set out in the schedule and your agreement to abide by the terms and conditions of this licence and limited warranty, SGL as the licensor grants to you the licencee, a non-exclusive right to use and display the copy of an SGL software product (hereinafter called the "software") on the computer and at the location listed in the schedules, so long as you comply with the terms of this licence. SGL reserves all rights not expressly granted to licensee.

Ownership of Software

As the licensee you own the magnetic or other physical media on which the software is originally or subsequently recorded or fixed, but it is an express condition of this licence that SGL retains title and ownership of the software and all subsequent copy(ies) of the software regardless of the form or media on which the software or other copies may exist. This licence is not a sale of the original software or of any copy.

Copy Restrictions

This software and the accompanying or subsequently provided written materials are the subject of copyright. Unauthorised copying of the software, including software that has been modified, merged or included with other software or of the written materials is expressly forbidden. You maybe held legally responsible for any copyright infringement which is caused or encouraged by your failure to abide by the terms of this licence.

Subject to the restrictions above you may make one copy of the software solely for backup purposes. You must reproduce and include copyright notices on the backup copy.

User Restrictions

As the licensee you may use the software on the computer(s) and the location(s) stated in the schedule. You may not electronically transfer the software from one computer to another over a network unless authorised by prior written consent of SGL to do so, save and except that, Licensee may transfer software over a network solely for the purposes of making back-up copies or re-installing from the backup following any system failure. You may not distribute copies of the software or accompanying materials to others. You may not modify, adapt, translate or create derivative work based on the written materials without the prior consent of SGL.

Transfer Restrictions

This software is licensed only to you, the licensee and may not be transferred to anyone without the prior written consent of SGL. Any authorized transfer of the software shall be bound by the terms and conditions of this licence. In no event may you transfer, assign, rent, lease, sell or otherwise dispose of the software on a temporary or permanent basis except as expressly provided herein.

Termination

This licence is effective unless terminated in accordance with the provisions of this clause. This licence will terminate automatically without notice from SGL if you fail to comply with any provision of this licence. If either party shall become bankrupt or make arrangement with its creditors or go into liquidation or be subject to reconstruction this licence will terminate. If SGL shall become bankrupt or make arrangement with its creditors or go into liquidation a copy of software is available through ESCROW and the source code will be made available to the licensee for their own internal use only. The licensee shall within 30 days of the events detailed in this clause provide to SGL a properly executed statutory declaration stating that it has destroyed or returned the software to SGL and permanently deleted and rendered useless all copies of the software and destroyed all supplied written materials.

Patents, Trademarks, Copyright and Intellectual Property Rights

The licensee acknowledges that any and all of the trademarks, copyrights, patents and other intellectual property rights used or embodied in or in connection with the software or its maintenance or support shall remain the sole property of SGL. The licensee agrees not to in anyway question or dispute ownership by SGL of any such rights, during or at any time after the expiry or termination of this agreement (neither in part nor in whole).

Limited Warranty

The licensor warrants the supplied software will substantially conform to the specifications established by SGL. SGL does not warrant that the software is free from all bugs, errors and omissions. Further SGL does not warrant nor guarantee nor make any representations regarding the use or accuracy, reliability, correctness or otherwise. The above is the only warranty of any kind either expressed or implied, including but not limited to fitness of uses or purposes. No oral or written information by SGL, its employees or authorized associates shall increase the scope of this warranty and you may not rely on such information. SGL nor anyone else connected with creation, production or delivery of the software shall be liable to the licensee for any loss or damage, however caused whether directly or indirectly in connection with or arising out of the use or inability to use software.

SGL shall expressly exclude liability for but not limited to damage to equipment, loss of business profits, loss of business information, good will, anticipated savings and the like. SGL does not exclude liability for death or personal injury to the extent only that the same arises as the result of negligence of SGL, its employees, agents or authorized representative.

Indemnity

Notwithstanding anything contained in the Limited Warranty clause, SGL warrants that it has title or authority to provide the licence granted therein, and at its own cost will defend and hold licensee harmless from any action against the licensee based on a claim that the licensee's use of the software in accordance with this licence agreement infringes any patent, copyright or any other intellectual property of any third party.

Entire Agreement

The agreement shall be governed by and conducted in accordance with English law and shall benefit SGL, its successors and assigns.

Acknowledgement

You acknowledge that you have read this licence and limited warranty, understand them and agree to be bound by their terms and conditions. You also agree that the licence and limited warranty are the complete and exclusive statement between the parties.

Definitions

SGL means:

Software Generation Limited
12, Fulcrum 2
Solent Way, Whiteley,
Fareham
Hants, England. PO15 7FN
Tel: +44 (0)1489 88 99 30
Fax: + 44 (0)1489 56 58 25
Email: *info@sgluk.com*
http://www.sgluk.com

Software means:

The computer program(s) supplied together with all codes, technique, software tools, formats, designs, concepts, methods and ideas associated with the computer program(s). The term also includes all copies of any part of the software as well as manual(s) and other printed materials.

FlashNet End User Software Licence Agreement - NF0310

— Portions of this software are governed by the GNU *Lesser General Public License* available from *http://www.gnu.org/licenses/lgpl.html*.

— Portions of this software were developed through the RED R3D Software Developer's Kit (SDK), made available by RED.COM, Inc. If a Xinet customer requires any other RED software in order to use the licensed Xinet software, the Xinet customer must obtain such other RED software via a valid license, and the customer's use of such other RED software must be in accordance with the terms and conditions of the End User License Agreement that accompanies such other RED software.

The R3D SDK and all included materials (including header files, libraries, sample code & documentation) are Copyright © 2008-2011 RED Digital Cinema. All rights reserved. All trademarks are the property of their respective owners. This software was developed using KAKADU software.

— Portions of this software use the Lynx text-based Web browser to load and extract text from Web pages. Lynx was released under the *GNU General Public License version 2* in 1995 by the University of Kansas; since then it has been maintained through the work of developers on the *lynx-dev* mailing list (*lynx-dev@nongnu.org*), who have agreed to distribute their work under the terms and conditions of the GNU GPL. The original copyright notice for Lynx 2.4.2 follows; a copy of the full text of the latest GNU GPL is also presented below.

Copyright © 1995 University of Kansas.
This file is part of Lynx, a text-based WWW browser.

Lynx is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License, version 2, as published by the Free Software Foundation. Lynx is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Gerneral Public License along with Lynx; see the file 'COPYING'. If not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. Any licensing or usage questions should be directed to Michael Grobe <*grobe@ukans.edu*>.

— Use of a modified '*ffmpeg*' library with the Xinet Video software is governed by the GNU *General Public License version 2 or later*. Running */usr/etc/venture/bin/ffmpeg -L* from the command prompt on UNIX systems and *C:\Program Files\Xinet\Venture\ffmpeg.exe -L* on Windows systems provides *FFmpeg* copyright information. As Xinet's PC Connectivity software package contains modified software from *samba.org*, Xinet Samba modifications and VFS Module are released and distributed as open-source under the GNU *GPL* terms as well. Xinet will make the full source code of its modifications available to any interested parties under the terms of the GNU *GPL*; the complete GPL License appears below.

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

* b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

* c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

* d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

* b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

* c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

* d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

* e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

* b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

* c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

* d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

* e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

* f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Down Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to down recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license,

your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

Xinet Trademark Acknowledgements, Attributions, and Credit Notices

Revision Date: March 23, 2012

#####

ENDBENUTZER-LIZENZVERTRAG FÜR XINET-SOFTWARE

Gültig nur für Produkte, die in Deutschland lizensiert werden.

This applies only to products licensed in Germany.

Wichtig bitte sorgfältig lesen: Dieser XINET-Endbenutzer-Lizenzvertrag ist ein rechtsgültiger Vertrag zwischen Ihnen (entweder als natürliche oder juristischer Person) und der XINET Inc. für ein XINET-Software-Produkt. Das Software-Produkt umfaßt Computer-Software sowie möglicherweise dazugehörige Medien, gedruckte Materialien und „Online“ oder elektronische Dokumentation („Software-Produkt“). Indem Sie das Software-Produkt installieren, kopieren oder anderweitig verwenden, erklären Sie sich einverstanden, durch die Bestimmungen dieses Lizenzvertrages gebunden zu sein. Falls Sie diesen Bestimmungen nicht zustimmen, sind Sie nicht berechtigt, das Software-Produkt zu installieren oder zu verwenden; Sie können es jedoch gegen Rückerstattung des Kaufpreises an den Händler zurückgeben, von dem Sie es erworben haben.

Das Software-Produkt wird sowohl durch Urheberrechtsgesetze und internationale Urheberrechtsverträge geschützt, als auch durch andere Gesetze und Vereinbarungen über geistiges Eigentum. Das Software-Produkt wird lizenziert, nicht verkauft.

1. Lizenzeinräumung. XINET räumt Ihnen unter der Einschränkung der Regelung in Ziff. 2.f) dieses Vertrages das zeitlich unbegrenzte, nicht ausschließliche und nicht übertragbare Recht ein, das Software-Produkt im vertragsgemäßen Umfang zu nutzen. Sie sind berechtigt, eine Kopie des Software-Produkts auf einem einzelnen Computer zu installieren und zu verwenden. Ferner dürfen Sie eine Sicherungskopie anfertigen.

2. Weitere Rechte und Einschränkungen

a) Nicht zum Weiterverkauf bestimmte Software. Falls Ihnen das Software-Produkt als Demonstrationsmaterial überlassen wird, sind Sie ungeachtet anderer Abschnitte dieses Lizenzvertrages, nicht berechtigt, das Software-Produkt weiterzuverkaufen oder auf andere Weise gegen einen Gegenwert oder unentgeltlich zu übertragen. Sollten Sie die Hardware auf welcher die Software installiert ist veräußern, sind Sie verpflichtet, sämtliche Kopien des Softwareprodukts zu zerstören bzw. an XINET zurückzugeben.

b) Beschränkung im Hinblick auf Zurückentwicklung (Reverse Engineering), Dekompilierung und Disassemblierung. Sie sind nicht berechtigt, das Software-Produkt zurückzuentwickeln (Reverse Engineering), zu dekompilieren oder zu disassemblieren, es sei denn und nur insoweit wie das anwendbare Recht, ungeachtet dieser Beschränkung, dieses ausdrücklich gestattet.

c) Trennung von Komponenten. Das Software-Produkt wird als einzelnes Produkt lizenziert. Sie sind nicht berechtigt, dessen Komponenten für die Verwertung auf mehr als einem Computer zu trennen.

d) Vermietung. Sie sind nicht berechtigt, das Software-Produkt zu vermieten, zu verleasen oder zu verleihen.

e) Serviceleistungen. XINET bietet Ihnen möglicherweise Serviceleistungen in Verbindung mit dem Software-Produkt („Serviceleistungen“). Die Serviceleistungen können entsprechend den XINET-Bestimmungen und -Programmen, die im Benutzerhandbuch, der „Online“-Dokumentation und/oder anderen von XINET zur Verfügung gestellten Materialien beschrieben sind, genutzt werden. Jeder ergänzende Software-Code, der Ihnen als Teil der Serviceleistungen zur Verfügung gestellt wird, wird als Bestandteil des Software-Produkts betrachtet und unterliegt den Bestimmungen und Bedingungen dieses Lizenzvertrages. XINET ist berechtigt, die technischen

Daten, die Sie XINET als Teil der Serviceleistungen zur Verfügung stellen, für geschäftliche Zwecke, einschließlich Produktunterstützung und -entwicklung, zu verwenden. XINET verpflichtet sich, solche technische Daten ausschließlich anonym im Sinne des Datenschutzes zu verwenden.

f) Kündigung. Unbeschadet sonstiger Rechte ist XINET berechtigt, diesen Lizenzvertrag zu kündigen, sofern Sie gegen die Bestimmungen und Bedingungen dieses Lizenzvertrages verstoßen. In einem solchen Fall sind Sie verpflichtet, sämtliche Kopien des Software-Produkts und alle seine Komponenten zu vernichten. Durch Zerstörung der Software einschließlich seiner Kopie und damit verbundenen Dokumentation kündigen Sie selbst den Lizenzvertrag.

3. Urheberrecht. Eigentum und Urheberrecht an dem Software-Produkt (einschließlich, aber nicht beschränkt auf Bilder, Fotografien, Animationen, Video, Audio, Musik, Text und „Applets“, die in dem Software-Produkt enthalten sind), den gedruckten Begleitmaterialien und jeder Kopie des Software-Produkts liegen bei XINET. Das Software-Produkt ist durch Urheberrechtsgesetze und internationale Urheberrechtsbestimmungen geschützt. Aus diesem Grund sind Sie verpflichtet, das Software-Produkt wie jedes andere durch das Urheberrecht geschützte Material zu behandeln, mit der Ausnahme, daß Sie berechtigt sind, das Software-Produkt auf einem einzelnen Computer zu installieren, vorausgesetzt, Sie bewahren das Original ausschließlich für Sicherungs- und Archivierungszwecke auf und schützen es vor unerlaubter Offenbarung. Sie sind nicht berechtigt, dass das Software-Produkt begleitende gedruckte Material zu vervielfältigen.

4. Gewährleistungen

a) XINET und ihre Vertragshändler weisen darauf hin, daß es nicht möglich ist, Software so zu entwickeln, dass sie für alle Anwendungsbedingungen fehlerfrei ist. Dafür übernimmt XINET für einen Zeitraum von 30 Tagen ab Auslieferung an Sie die Garantie, dass das Software-Produkt im wesentlichen gem. den beiliegenden bedruckten Materialien arbeitet.

b) Darüber hinaus stehen XINET und ihre Vertragshändler für Mängel, die bei der Übergabe des Software-Produkts vorhanden sind, während einer Gewährleistungsfrist von 6 Monaten gem. folgenden Regeln ein.

c) Als Mängel im Sinn der Ziffer 4.b) gelten Abweichungen des Software-Produkts von den beiliegenden bedruckten Materialien, soweit diese Abweichungen die Tauglichkeit des Software-Produkts beeinträchtigen. Eine unerhebliche Minderung der Brauchbarkeit bleibt außer Betracht.

d) Erweist sich das Software-Produkt als nicht brauchbar oder fehlerhaft, werden XINET und ihre Vertragshändler diese Mängel binnen angemessener Frist beheben. Gelingt ein derartiger Nachbesserungsversuch nicht innerhalb angemessener Frist und schlägt er auch innerhalb einer weiteren, von Ihnen angemessen gesetzten Frist fehl, und stellen XINET und ihre Vertragshändler keine Umgehungslösung gem. Ziff. 4.e) zur Verfügung, so stehen Ihnen die gesetzlichen Gewährleistungsrechte zu. Sie können dann den Vergütungsanspruch herabsetzen (mindern) oder den Vertrag rückgängig machen (wandeln).

e) XINET und ihre Vertragshändler sind berechtigt, einen eventuell auftretenden Fehler zu umgehen, wenn der Fehler selbst nur durch unverhältnissemäßigen Aufwand zu beseitigen ist und dadurch der Einsatz des Software-Produkts nicht erheblich leidet.

f) XINET und ihre Vertragshändler haften nicht, wenn der Fehler unseres Produkts auf einem Unfall, Mißbrauch oder fehlerhaften Anwendung beruht.

g) XINET und ihre Vertragshändler haften nicht für Folgeschäden und im übrigen nur nach den Bestimmungen der Ziff. 5. Weitergehende Gewährleistungsansprüche sind ausgeschlossen. Darüber hinausgehende Leistungen bieten XINET und ihre Vertragshändler im Zusammenhang mit Support-Leistungen an, soweit nicht ein zusätzlicher Vertrag über Support-Leistungen abgeschlossen wurde.

5. Haftung

a) XINET und ihre Vertragshändler haften für anfängliches Unvermögen, Verzug und Unmöglichkeit auf solche Schäden begrenzt, die aufgrund der vertraglichen Verwendung typisch und vorhersehbar sind. In einem derartigen Fall ist die Haftung auf den Betrag, den Sie für das Software-Produkt bezahlt haben, begrenzt.

b) Im übrigen haften XINET und ihre Vertragshändler unbeschränkt nur für Schäden, die durch Vorsatz oder grobe Fahrlässigkeit ihrer gesetzlichen Vertreter und/oder ihrer leitenden Angestellten verursacht wurden. Für Schäden, die

durch ein Verschulden sonstiger Mitarbeiter und Erfüllungsgehilfen verursacht worden sind, haften XINET und ihre Vertragshändler nur im Umfang der Haftung nach Ziff. 5.a).

c) Für leichte Fahrlässigkeit haften XINET und ihre Vertragshändler nur, sofern eine Pflicht verletzt wird, deren Einhaltung für die Erreichung des Vertragszwecks von besonderer Bedeutung ist (Kardinalpflicht). Bei Verletzung einer Kardinalpflicht gilt Ziff. 5.a) entsprechend.

d) Den besonderen Risiken der Überlassung des Software-Produkts entsprechend wird die Haftung für Folgeschäden und entgangenen Gewinn ausgeschlossen, soweit ein derartiger Schaden nicht auf Rechtsmängel, das Fehlen zugesicherter Eigenschaften, Vorsatz oder grobe Fahrlässigkeit der gesetzlichen Vertreter oder leitenden Angestellten zurückzuführen ist. XINET und ihre Vertragshändler haften nur für die Wiederherstellung von Daten, soweit Sie regelmäßig und gefahr-entsprechend Sicherungskopien angefertigt und sichergestellt haben, daß die Daten aus diesen Sicherungskopien mit vertretbaren Aufwand rekonstruiert werden können. Eine darüber hinausgehende Haftung für Datenverluste ist ausgeschlossen.

e) Die vorstehenden Regelungen gelten auch zugunsten der Mitarbeiter und Erfüllungsgehilfen von XINET und ihren Vertragshändlern. Die Haftungsbeschränkungen gelten nicht für Ansprüche nach dem Produkthaftungsgesetz.

6. Anwendbares Recht. Wenn Sie unser Produkt in Deutschland gekauft haben, unterliegt der Lizenzvertrag deutschem Recht.

History of this document

2 September 2009	Issued in draft form with limited distribution
18 December 2009	Version 3.02: added syntax for `action = search`, introductory material, CGI and JSON RFCs, and extensive index.
26 January 2010	Version 3.03: added details about `basketname =`, `basketfile =`, and `skiphidden =`
8 March 2010	Version 3.03, 2nd Edition: Added details about `pathsearchtype_i`
14 June 2011 29 July 2011	Version 4.0: added PHP Example; added details about `show-links=true`.
14 March 2012	Version 4.01: revised for compliance with WebNative Suite 17.01 and WebNative Portal 4.01
	Extensive updates to `action = browse` and `action = search`; removed and summarized deprecated arguments. (Note: information about Annotations has not been reviewed for 4.01 compliance.)
17 April 2012	Version 4.01, 2nd Edition: removed [`searchid = value`] from *browse* syntax. Corrected *search* syntax for `dbsearch_*` filters.
25 April 2012	Version 4.02: added Asset Timer Status filter for `action=search`, compliant with WebNative Suite 17.02.
10 September 2012	Version 4.02, 3rd Edition: Corrected action = search syntax errors on pages 16 and explanation on page 26.
21 June 2013	Version 4.5: revised to add “upload” and “getorderimage” actions

The Common Gateway Interface (CGI) Version 1.1

Included, courtesy of:
Network Working Group
Request for Comments: 3875
Category: Informational

D. Robinson
K. Coar
The Apache Software Foundation
October 2004

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

IESG Note

This document is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this document for any purpose, and in particular notes that it has not had IETF review for such things as security, congestion control or inappropriate interaction with deployed protocols. The RFC Editor has chosen to publish this document at its discretion. Readers of this document should exercise caution in evaluating its value for implementation and deployment.

Abstract

The Common Gateway Interface (CGI) is a simple interface for running external programs, software or gateways under an information server in a platform-independent manner. Currently, the supported information servers are HTTP servers.

The interface has been in use by the World-Wide Web (WWW) since 1993. This specification defines the 'current practice' parameters of the CGI/1.1' interface developed and documented at the U.S. National Centre for Supercomputing Applications. This document also defines the use of the CGI/1.1 interface on UNIX® and other, similar systems.

1. Introduction

1.1. Purpose

The Common Gateway Interface (CGI) [22] allows an HTTP [1], [4] server and a CGI script to share responsibility for responding to client requests. The client request comprises a Uniform Resource Identifier (URI) [11], a request method and various information about the request provided by the transport protocol.

The CGI defines the abstract parameters, known as meta-variables, which describe a client's request. Together with a concrete programmer interface this specifies a platform-independent interface between the script and the HTTP server.

The server is responsible for managing connection, data transfer, transport and network issues related to the client request, whereas the CGI script handles the application issues, such as data access and document processing.

1.2. Requirements

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY' and 'OPTIONAL' in this document are to be interpreted as described in BCP 14, RFC 2119 [3].

An implementation is not compliant if it fails to satisfy one or more of the 'must' requirements for the protocols it implements. An implementation that satisfies all of the 'must' and all of the 'should' requirements for its features is said to be 'unconditionally compliant'; one that satisfies all of the 'must' requirements but not all of the 'should' requirements for its features is said to be 'conditionally compliant'.

1.3. Specifications

Not all of the functions and features of the CGI are defined in the main part of this specification. The following phrases are used to describe the features that are not specified:

'system-defined'
The feature may differ between systems, but must be the same for different implementations using the same system. A system will usually identify a class of operating systems. Some systems are defined in section 7 of this document. New systems may be defined by new specifications without revision of this document.

'implementation-defined'
The behavior of the feature may vary from implementation to implementation; a particular implementation must document its behavior.

1.4. Terminology

This specification uses many terms defined in the HTTP/1.1 specification [4]; however, the following terms are used here in a sense which may not accord with their definitions in that document, or with their common meaning.

'meta-variable'
A named parameter which carries information from the server to the script. It is not necessarily a variable in the operating system's environment, although that is the most common implementation.

'script'
The software that is invoked by the server according to this interface. It need not be a standalone program, but could be a dynamically-loaded or shared library, or even a subroutine in the server. It might be a set of statements interpreted at run-time, as the term 'script' is frequently understood, but that is not a requirement and within the context of this specification the term has the broader definition stated.

'server'
The application program that invokes the script in order to service requests from the client.

2. Notational Conventions and Generic Grammar

2.1. Augmented BNF

All of the mechanisms specified in this document are described in both prose and an augmented Backus-Naur Form (BNF) similar to that used by RFC 822 [13]. Unless stated otherwise, the elements are case-sensitive. This augmented BNF contains the following constructs:

name = definition
The name of a rule and its definition are separated by the equals character ('='). White space is only significant in that continuation lines of a definition are indented.

"literal"
Double quotation marks (") surround literal text, except for a literal quotation mark, which is surrounded by angle-brackets ('<' and '>').

rule1 | rule2
Alternative rules are separated by a vertical bar ('|').

(*rule1 rule2 rule3*)
Elements enclosed in parentheses are treated as a single element.

**rule*
A rule preceded by an asterisk ('*') may have zero or more occurrences. The full form is 'n*m rule' indicating at least n and at most m occurrences of the rule. n and m are optional decimal values with default values of 0 and infinity respectively.

[*rule*]
An element enclosed in square brackets (‘[‘ and ‘]’) is optional, and is equivalent to ‘*1 rule’.

N rule
A rule preceded by a decimal number represents exactly *N* occurrences of the rule. It is equivalent to ‘*N*N rule*’.

2.2. Basic Rules

This specification uses a BNF-like grammar defined in terms of characters. Unlike many specifications which define the bytes allowed by a protocol, here each literal in the grammar corresponds to the character it represents. How these characters are represented in terms of bits and bytes within a system are either system-defined or specified in the particular context. The single exception is the rule ‘OCTET’, defined below.

The following rules are used throughout this specification to describe basic parsing constructs.

alpha = lowalpha | hialpha

lowalpha = “a” | “b” | “c” | “d” | “e” | “f” | “g” | “h” | “i” | “j” | “k” | “l” | “m” | “n” | “o” | “p” | “q” | “r” | “s” | “t” | “u” | “v” | “w” | “x” | “y” | “z”

hialpha = “A” | “B” | “C” | “D” | “E” | “F” | “G” | “H” | “I” | “J” | “K” | “L” | “M” | “N” | “O” | “P” | “Q” | “R” | “S” | “T” | “U” | “V” | “W” | “X” | “Y” | “Z”

digit = “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”

alphanum = alpha | digit OCTET = <any 8-bit byte>

CHAR = a lpha | digit | separator | “!” | “#” | “$” | “%” | “&” | “’” | “*” | “+” | “-” | “.” | “`” | “^” | “_” | “{“ | “|” | “}” | “~” |

CTL CTL = <any control character>

SP = <space character>

HT = <horizontal tab character>

NL = <newline>

LWSP = SP | HT | NL

separator = “(“ | “)” | “<“ | “>” | “@” | “,” | “;” | “:” | “\” | <“> | “/” | “[“ | “]” | “?” | “=” | “{“ | “}” | SP | HT

token = 1*<any CHAR except CTLs or separators>

quoted-string = <“> *qdtext <“>

qdtext = <any CHAR except <“> and CTLs but including LWSP>

TEXT = <any printable character>

Note that newline (NL) need not be a single control character, but can be a sequence of control characters. A system MAY define TEXT to be a larger set of characters than <any CHAR excluding CTLs but including LWSP>.

2.3. URL Encoding

Some variables and constructs used here are described as being 'URL-encoded'. This encoding is described in section 2 of RFC 2396 [2]. In a URL-encoded string an escape sequence consists of a percent character ("%") followed by two hexadecimal digits, where the two hexadecimal digits form an octet. An escape sequence represents the graphic character that has the octet as its code within the US-ASCII [9] coded character set, if it exists. Currently there is no provision within the URI syntax to identify which character set non-ASCII codes represent, so CGI handles this issue on an ad-hoc basis.

Note that some unsafe (reserved) characters may have different semantics when encoded. The definition of which characters are unsafe depends on the context; see section 2 of RFC 2396 [2], updated by RFC 2732 [7], for an authoritative treatment. These reserved characters are generally used to provide syntactic structure to the character string, for example as field separators. In all cases, the string is first processed with regard to any reserved characters present, and then the resulting data can be URL-decoded by replacing "%" escape sequences by their character values.

To encode a character string, all reserved and forbidden characters are replaced by the corresponding "%" escape sequences. The string can then be used in assembling a URI. The reserved characters will vary from context to context, but will always be drawn from this set:

reserved = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" | "$" | "," | "[" | "]"

The last two characters were added by RFC 2732 [7]. In any particular context, a sub-set of these characters will be reserved; the other characters from this set MUST NOT be encoded when a string is URL-encoded in that context. Other basic rules used to describe URI syntax are:

hex = digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"
escaped = "%" hex hex
unreserved = alpha | digit | mark
mark = "-" | "_" | "." | "!" | "~" | "*" | "'" | "(" | ")"

3. Invoking the Script

3.1. Server Responsibilities

The server acts as an application gateway. It receives the request from the client, selects a CGI script to handle the request, converts the client request to a CGI request, executes the script and converts the CGI response into a response for the client. When processing the client request, it is responsible for implementing any protocol or transport level authentication and security. The

server MAY also function in a 'non-transparent' manner, modifying the request or response in order to provide some additional service, such as media type transformation or protocol reduction.

The server MUST perform translations and protocol conversions on the client request data required by this specification. Furthermore, the server retains its responsibility to the client to conform to the relevant network protocol even if the CGI script fails to conform to this specification.

If the server is applying authentication to the request, then it MUST NOT execute the script unless the request passes all defined access controls.

3.2. Script Selection

The server determines which CGI script is to be executed based on a generic-form URI supplied by the client. This URI includes a hierarchical path with components separated by "/". For any particular request, the server will identify all or a leading part of this path with an individual script, thus placing the script at a particular point in the path hierarchy. The remainder of the path, if any, is a resource or sub-resource identifier to be interpreted by the script.

Information about this split of the path is available to the script in the meta-variables, described below. Support for non-hierarchical URI schemes is outside the scope of this specification.

3.3. The Script-URI

The mapping from client request URI to choice of script is defined by the particular server implementation and its configuration. The server may allow the script to be identified with a set of several different URI path hierarchies, and therefore is permitted to replace the URI by other members of this set during processing and generation of the meta-variables. The server

1. MAY preserve the URI in the particular client request; or
2. it MAY select a canonical URI from the set of possible values for each script; or
3. it can implement any other selection of URI from the set.

From the meta-variables thus generated, a URI, the 'Script-URI', can be constructed. This MUST have the property that if the client had accessed this URI instead, then the script would have been executed with the same values for the SCRIPT_NAME, PATH_INFO and QUERY_STRING meta-variables. The Script-URI has the structure of a generic URI as defined in section 3 of RFC 2396 [2], with the exception that object parameters and fragment identifiers are not permitted. The various components of the Script-URI are defined by some of the meta-variables (see below);

```
script-URI =   <scheme> "://" <server-name> ":" <server-port> <script-path> <extra-path>
               "?" <query-string>
```

where <scheme> is found from SERVER_PROTOCOL, <server-name>, <server-port> and <query-string> are the values of the respective meta-variables. The SCRIPT_NAME and PATH_INFO values, URL-encoded with ";", "=" and "?" reserved, give <script-path> and <extra-path>.

See "4.1.5. PATH_INFO" on page 82 for more information about the PATH_INFO meta-variable.

The scheme and the protocol are not identical as the scheme identifies the access method in addition to the application protocol. For example, a resource accessed using Transport Layer Security (TLS) [14] would have a request URI with a scheme of *https* when using the HTTP protocol [19]. CGI/1.1 provides no generic means for the script to reconstruct this, and therefore the Script-URI as defined includes the base protocol used. However, a script MAY make use of scheme-specific meta-variables to better deduce the URI scheme.

Note that this definition also allows URIs to be constructed which would invoke the script with any permitted values for the path-info or query-string, by modifying the appropriate components.

3.4. Execution

The script is invoked in a system-defined manner. Unless specified otherwise, the file containing the script will be invoked as an executable program. The server prepares the CGI request as described in section 4; this comprises the request meta-variables (immediately available to the script on execution) and request message data. The request data need not be immediately available to the script; the script can be executed before all this data has been received by the server from the client. The response from the script is returned to the server as described in sections 5 and 6.

In the event of an error condition, the server can interrupt or terminate script execution at any time and without warning. That could occur, for example, in the event of a transport failure between the server and the client; so the script SHOULD be prepared to handle abnormal termination.

4. The CGI Request

Information about a request comes from two different sources; the request meta-variables and any associated message-body.

4.1. Request Meta-Variables

Meta-variables contain data about the request passed from the server to the script, and are accessed by the script in a system-defined manner. Meta-variables are identified by case-insensitive names; there cannot be two different variables whose names differ in case only. Here they are shown using a canonical representation of capitals plus underscore ("_"). A particular system can define a different representation.

```
meta-variable-name   =   "AUTH_TYPE" | "CONTENT_LENGTH" |
                         "CONTENT_TYPE" | "GATEWAY_INTERFACE" |
                         "PATH_INFO" | "PATH_TRANSLATED" |
                         "QUERY_STRING" |"REMOTE_ADDR" |
                         "REMOTE_HOST" | "REMOTE_IDENT" |
                         "REMOTE_USER" | "REQUEST_METHOD" |
                         "SCRIPT_NAME" | "SERVER_NAME" |
                         "SERVER_PORT" | "SERVER_PROTOCOL" |
                         "SERVER_SOFTWARE" | scheme |
                         protocol-var-name | extension-var-name
protocol-var-name    =   ( protocol | scheme ) "_" var-name
scheme               =   alpha *( alpha | digit | "+" | "-" | "." )
var-name             =   token
extension-var-name   =   token
```

Meta-variables with the same name as a scheme, and names beginning with the name of a protocol or scheme (e.g., HTTP_ACCEPT) are also defined. The number and meaning of these variables may change independently of this specification. (See also section 4.1.18.)

The server MAY set additional implementation-defined extension meta-variables, whose names SHOULD be prefixed with "X_".

This specification does not distinguish between zero-length (NULL) values and missing values. For example, a script cannot distinguish between the two requests http://host/script and http://host/script? as in both cases the QUERY_STRING meta-variable would be NULL.

```
meta-variable-value = "" | 1*<TEXT, CHAR or tokens of value>
```

An optional meta-variable may be omitted (left unset) if its value is NULL. Meta-variable values MUST be considered case-sensitive except as noted otherwise. The representation of the characters in the meta-variables is system-defined; the server MUST convert values to that representation.

4.1.1. AUTH_TYPE

The AUTH_TYPE variable identifies any mechanism used by the server to authenticate the user. It contains a case-insensitive value defined by the client protocol or server implementation.

For HTTP, if the client request required authentication for external access, then the server MUST set the value of this variable from the 'auth-scheme' token in the request Authorization header field.

```
AUTH_TYPE        =   "" | auth-scheme
auth-scheme      =   "Basic" | "Digest" | extension-auth
extension-auth   =   token
```

HTTP access authentication schemes are described in RFC 2617 [5].

4.1.2. CONTENT_LENGTH

The CONTENT_LENGTH variable contains the size of the message-body attached to the request, if any, in decimal number of octets. If no data is attached, then NULL (or unset).

CONTENT_LENGTH = “” | 1*digit

The server MUST set this meta-variable if and only if the request is accompanied by a message-body entity. The CONTENT_LENGTH value must reflect the length of the message-body after the server has removed any transfer-codings or content-codings.

4.1.3. CONTENT_TYPE

If the request includes a message-body, the CONTENT_TYPE variable is set to the Internet Media Type [6] of the message-body.

```
CONTENT_TYPE    =    “” | media-type
media-type      =    type “/” subtype *( “;” parameter )
type            =    token
subtype         =    token
parameter       =    attribute “=” value
attribute       =    token
value           =    token | quoted-string
```

The type, subtype and parameter attribute names are not case-sensitive. Parameter values may be case sensitive. Media types and their use in HTTP are described section 3.7 of the HTTP/1.1 specification [4].

There is no default value for this variable. If and only if it is unset, then the script MAY attempt to determine the media type from the data received. If the type remains unknown, then the script MAY choose to assume a type of application/octet-stream or it may reject the request with an error (as described in section 6.3.3).

Each media-type defines a set of optional and mandatory parameters. This may include a charset parameter with a case-insensitive value defining the coded character set for the message-body. If the charset parameter is omitted, then the default value should be derived according to whichever of the following rules is the first to apply:

1. There MAY be a system-defined default charset for some media-types.
2. The default for media-types of type “text” is ISO-8859-1 [4].
3. Any default defined in the media-type specification.
4. The default is US-ASCII.

The server MUST set this meta-variable if an HTTP Content-Type field is present in the client request header. If the server receives a request with an attached entity but no Content-Type header field, it MAY attempt to determine the correct content type, otherwise it should omit this meta-variable.

4.1.4. GATEWAY_INTERFACE

The GATEWAY_INTERFACE variable MUST be set to the dialect of CGI being used by the server to communicate with the script. Syntax:

GATEWAY_INTERFACE = “CGI” “/” 1*digit “.” 1*digit

Note that the major and minor numbers are treated as separate integers and hence each may be incremented higher than a single digit. Thus CGI/2.4 is a lower version than CGI/2.13 which in turn is lower than CGI/12.3. Leading zeros MUST be ignored by the script and MUST NOT be generated by the server.

This document defines the 1.1 version of the CGI interface.

4.1.5. PATH_INFO

The PATH_INFO variable specifies a path to be interpreted by the CGI script. It identifies the resource or sub-resource to be returned by the CGI script, and is derived from the portion of the URI path hierarchy following the part that identifies the script itself. Unlike a URI path, the PATH_INFO is not URL-encoded, and cannot contain path-segment parameters. A PATH_INFO of “/” represents a single void path segment.

```
PATH_INFO     =   “” | ( “/” path )
path          =   lsegment *( “/” lsegment )
lsegment      =   *lchar
lchar         =   <any TEXT or CTL except “/”>
```

The value is considered case-sensitive and the server MUST preserve the case of the path as presented in the request URI. The server MAY impose restrictions and limitations on what values it permits for PATH_INFO, and MAY reject the request with an error if it encounters any values considered objectionable. That MAY include any requests that would result in an encoded “/” being decoded into PATH_INFO, as this might represent a loss of information to the script. Similarly, treatment of non US-ASCII characters in the path is system-defined.

URL-encoded, the PATH_INFO string forms the extra-path component of the Script-URI (see section 3.3) which follows the SCRIPT_NAME part of that path.

4.1.6. PATH_TRANSLATED

The PATH_TRANSLATED variable is derived by taking the PATH_INFO value, parsing it as a local URI in its own right, and performing any virtual-to-physical translation appropriate to map it onto the server’s document repository structure. The set of characters permitted in the result is system-defined.

PATH_TRANSLATED = *<any character>

This is the file location that would be accessed by a request for

<scheme> “://” <server-name> “:” <server-port> <extra-path>

where <scheme> is the scheme for the original client request and <extra-path> is a URL-encoded version of PATH_INFO, with “;”, “=” and “?” reserved. For example, a request such as the following:

http://somehost.com/cgi-bin/somescript/this%2eis%2epath%3binfo

would result in a PATH_INFO value of

/this.is.the.path;info

An internal URI is constructed from the scheme, server location and the URL-encoded PATH_INFO:

http://somehost.com/this.is.the.path%3binfo

This would then be translated to a location in the server's document repository, perhaps a filesystem path something like this:

/usr/local/www/htdocs/this.is.the.path;info

The value of PATH_TRANSLATED is the result of the translation.

The value is derived in this way irrespective of whether it maps to a valid repository location. The server MUST preserve the case of the extra-path segment unless the underlying repository supports case-insensitive names. If the repository is only case-aware, case-preserving, or case-blind with regard to document names, the server is not required to preserve the case of the original segment through the translation.

The translation algorithm the server uses to derive PATH_TRANSLATED is implementation-defined; CGI scripts which use this variable may suffer limited portability.

The server SHOULD set this meta-variable if the request URI includes a path-info component. If PATH_INFO is NULL, then the PATH_TRANSLATED variable MUST be set to NULL (or unset).

4.1.7. QUERY_STRING

The QUERY_STRING variable contains a URL-encoded search or parameter string; it provides information to the CGI script to affect or refine the document to be returned by the script.

The URL syntax for a search string is described in section 3 of RFC 2396 [2]. The QUERY_STRING value is case-sensitive.

```
QUERY_STRING    =    query-string
query-string    =    *uric
uric            =    reserved | unreserved | escaped
```

When parsing and decoding the query string, the details of the parsing, reserved characters and support for non US-ASCII characters depends on the context. For example, form submission from an HTML document [18] uses application/x-www-form-urlencoded encoding, in which the characters "+", "&" and "=" are reserved, and the ISO 8859-1 encoding may be used for non US-ASCII characters.

The QUERY_STRING value provides the query-string part of the Script-URI. (See "3.3. The Script-URI" on page 78).

The server MUST set this variable; if the Script-URI does not include a query component, the QUERY_STRING MUST be defined as an empty string (“”).

4.1.8. REMOTE_ADDR

The REMOTE_ADDR variable MUST be set to the network address of the client sending the request to the server.

```
REMOTE_ADDR     =   hostnumber
hostnumber      =   ipv4-address | ipv6-address
ipv4-address    =   1*3digit “.” 1*3digit “.” 1*3digit “.” 1*3digit
ipv6-address    =   hexpart [ “:” ipv4-address ]
hexpart         =   hexseq | ( [ hexseq ] “::” [ hexseq ] )
hexseq          =   1*4hex *( “:” 1*4hex )
```

The format of an IPv6 address is described in RFC 3513 [15].

4.1.9. REMOTE_HOST

The REMOTE_HOST variable contains the fully qualified domain name of the client sending the request to the server, if available, otherwise NULL. Fully qualified domain names take the form as described in section 3.5 of RFC 1034 [17] and section 2.1 of RFC 1123 [12]. Domain names are not case sensitive.

```
REMOTE_HOST     =   “” | hostname | hostnumber
hostname        =   *( domainlabel “.” ) toplabel [ “.” ]
domainlabel     =   alphanum [ *alphahypdigit alphanum ]
toplabel        =   alpha [ *alphahypdigit alphanum ]
alphahypdigit   =   alphanum | “-”
```

The server SHOULD set this variable. If the hostname is not available for performance reasons or otherwise, the server MAY substitute the REMOTE_ADDR value.

4.1.10. REMOTE_IDENT

The REMOTE_IDENT variable MAY be used to provide identity information reported about the connection by an RFC 1413 [20] request to the remote agent, if available. The server may choose not to support this feature, or not to request the data for efficiency reasons, or not to return available identity data.

```
REMOTE_IDENT = *TEXT
```

The data returned may be used for authentication purposes, but the level of trust reposed in it should be minimal.

4.1.11. REMOTE_USER

The REMOTE_USER variable provides a user identification string supplied by client as part of user authentication.

```
REMOTE_USER = *TEXT
```

If the client request required HTTP Authentication [5] (e.g., the AUTH_TYPE meta-variable is set to “Basic” or “Digest”), then the value of the REMOTE_USER meta-variable MUST be set to the user-ID supplied.

4.1.12. REQUEST_METHOD

The REQUEST_METHOD meta-variable MUST be set to the method which should be used by the script to process the request, as described in section 4.3.

```
REQUEST_METHOD   =   method
method           =   “GET” | “POST” | “HEAD” | extension-method
extension-method =   “PUT” | “DELETE” | token
```

The method is case sensitive. The HTTP methods are described in section 5.1.1 of the HTTP/1.0 specification [1] and section 5.1.1 of the HTTP/1.1 specification [4].

4.1.13. SCRIPT_NAME

The SCRIPT_NAME variable MUST be set to a URI path (not URL-encoded) which could identify the CGI script (rather than the script’s output). The syntax is the same as for PATH_INFO (section 4.1.5)

```
SCRIPT_NAME = “” | ( “/” path )
```

The leading “/” is not part of the path. It is optional if the path is NULL; however, the variable MUST still be set in that case.

The SCRIPT_NAME string forms some leading part of the path component of the Script-URI derived in some implementation-defined manner. No PATH_INFO segment (see section 4.1.5) is included in the SCRIPT_NAME value.

4.1.14. SERVER_NAME

The SERVER_NAME variable MUST be set to the name of the server host to which the client request is directed. It is a case-insensitive hostname or network address. It forms the host part of the Script-URI.

```
SERVER_NAME   =   server-name
server-name   =   hostname | ipv4-address | ( “[“ ipv6-address “]” )
```

A deployed server can have more than one possible value for this variable, where several HTTP virtual hosts share the same IP address. In that case, the server would use the contents of the request’s Host header field to select the correct virtual host.

4.1.15. SERVER_PORT

The SERVER_PORT variable MUST be set to the TCP/IP port number on which this request is received from the client. This value is used in the port part of the Script-URI.

```
SERVER_PORT   =   server-port
server-port   =   1*digit
```

Note that this variable MUST be set, even if the port is the default port for the scheme and could otherwise be omitted from a URI.

4.1.16. SERVER_PROTOCOL

The SERVER_PROTOCOL variable MUST be set to the name and version of the application protocol used for this CGI request. This MAY differ from the protocol version used by the server in its communication with the client.

```
SERVER_PROTOCOL   =   HTTP-Version | "INCLUDED" | extension-version
HTTP-Version      =   "HTTP" "/" 1*digit "." 1*digit
extension-version =   protocol [ "/" 1*digit "." 1*digit ]
protocol          =   token
```

Here, 'protocol' defines the syntax of some of the information passing between the server and the script (the 'protocol-specific' features). It is not case sensitive and is usually presented in upper case. The protocol is not the same as the scheme part of the script URI, which defines the overall access mechanism used by the client to communicate with the server. For example, a request that reaches the script with a protocol of "HTTP" may have used an "https" scheme.

A well-known value for SERVER_PROTOCOL which the server MAY use is "INCLUDED", which signals that the current document is being included as part of a composite document, rather than being the direct target of the client request. The script should treat this as an HTTP/1.0 request.

4.1.17. SERVER_SOFTWARE

The SERVER_SOFTWARE meta-variable MUST be set to the name and version of the information server software making the CGI request (and running the gateway). It SHOULD be the same as the server description reported to the client, if any.

```
SERVER_SOFTWARE   =   1*( product | comment )
product           =   token [ "/" product-version ]
product-version   =   token
comment           =   "(" *( ctext | comment ) ")"
ctext             =   <any TEXT excluding "(" and ")">
```

4.1.18. Protocol-Specific Meta-Variables

The server SHOULD set meta-variables specific to the protocol and scheme for the request. Interpretation of protocol-specific variables depends on the protocol version in SERVER_PROTOCOL. The server MAY set a meta-variable with the name of the scheme to a non-NULL value if the scheme is not the same as the protocol. The presence of such a variable indicates to a script which scheme is used by the request.

Meta-variables with names beginning with "HTTP_" contain values read from the client request header fields, if the protocol used is HTTP. The HTTP header field name is converted to upper case, has all occurrences of "-" replaced with "_" and has "HTTP_" prepended to give the meta-variable name. The header data can be presented as sent by the client, or can be rewritten in ways which do not change its semantics. If multiple header fields with the same field-name are received then the server MUST rewrite them as a single value having the same

semantics. Similarly, a header field that spans multiple lines MUST be merged onto a single line. The server MUST, if necessary, change the representation of the data (for example, the character set) to be appropriate for a CGI meta-variable.

The server is not required to create meta-variables for all the header fields that it receives. In particular, it SHOULD remove any header fields carrying authentication information, such as 'Authorization'; or that are available to the script in other variables, such as 'Content-Length' and 'Content-Type'. The server MAY remove header fields that relate solely to client-side communication issues, such as 'Connection'.

4.2. Request Message-Body

Request data is accessed by the script in a system-defined method; unless defined otherwise, this will be by reading the 'standard input' file descriptor or file handle.

```
Request-Data      =    [ request-body ] [ extension-data ]
request-body      =    <CONTENT_LENGTH>OCTET
extension-data    =    *OCTET
```

A request-body is supplied with the request if the CONTENT_LENGTH is not NULL. The server MUST make at least that many bytes available for the script to read. The server MAY signal an end-of-file condition after CONTENT_LENGTH bytes have been read or it MAY supply extension data. Therefore, the script MUST NOT attempt to read more than CONTENT_LENGTH bytes, even if more data is available. However, it is not obliged to read any of the data.

For non-parsed header (NPH) scripts (section 5), the server SHOULD attempt to ensure that the data supplied to the script is precisely as supplied by the client and is unaltered by the server.

As transfer-codings are not supported on the request-body, the server MUST remove any such codings from the message-body, and recalculate the CONTENT_LENGTH. If this is not possible (for example, because of large buffering requirements), the server SHOULD reject the client request. It MAY also remove content-codings from the message-body.

4.3. Request Methods

The Request Method, as supplied in the REQUEST_METHOD meta-variable, identifies the processing method to be applied by the script in producing a response. The script author can choose to implement the methods most appropriate for the particular application. If the script receives a request with a method it does not support it SHOULD reject it with an error (see section 6.3.3).

4.3.1. GET

The GET method indicates that the script should produce a document based on the meta-variable values. By convention, the GET method is 'safe' and 'idempotent' and SHOULD NOT have the significance of taking an action other than producing a document.

The meaning of the GET method may be modified and refined by protocol-specific meta-variables.

4.3.2. POST

The POST method is used to request the script perform processing and produce a document based on the data in the request message-body, inaddition to meta-variable values. A common use is form submission in HTML [18], intended to initiate processing by the script that has a permanent affect, such a change in a database.

The script MUST check the value of the CONTENT_LENGTH variable before reading the attached message-body, and SHOULD check the CONTENT_TYPE value before processing it.

4.3.3. HEAD

The HEAD method requests the script to do sufficient processing to return the response header fields, without providing a response message-body. The script MUST NOT provide a response message-body for a HEAD request. If it does, then the server MUST discard the message-body when reading the response from the script.

4.3.4. Protocol-Specific Methods

The script MAY implement any protocol-specific method, such as HTTP/1.1 PUT and DELETE; it SHOULD check the value of SERVER_PROTOCOL when doing so.

The server MAY decide that some methods are not appropriate or permitted for a script, and may handle the methods itself or return an error to the client.

4.4. The Script Command Line

Some systems support a method for supplying an array of strings to the CGI script. This is only used in the case of an ‘indexed’ HTTP query, which is identified by a ‘GET’ or ‘HEAD’ request with a URI query string that does not contain any unencoded “=” characters. For such a request, the server SHOULD treat the query-string as a search-string and parse it into words, using the rules

```
search-string    =    search-word *( "+" search-word )
search-word      =    1*schar
schar            =    unreserved | escaped | xreserved
xreserved        =    ";" | "/" | "?" | ":" | "@" | "&" | "=" | "," | "$"
```

After parsing, each search-word is URL-decoded, optionally encoded in a system-defined manner and then added to the command line argument list.

If the server cannot create any part of the argument list, then the server MUST NOT generate any command line information. For example, the number of arguments may be greater than operating system or server limits, or one of the words may not be representable as an argument.

The script SHOULD check to see if the QUERY_STRING value contains an unencoded "=" character, and SHOULD NOT use the command line arguments if it does.

5. NPH Scripts

5.1. Identification

The server MAY support NPH (Non-Parsed Header) scripts; these are scripts to which the server passes all responsibility for response processing.

This specification provides no mechanism for an NPH script to be identified on the basis of its output data alone. By convention, therefore, any particular script can only ever provide output of one type (NPH or CGI) and hence the script itself is described as an 'NPH script'. A server with NPH support MUST provide an implementation-defined mechanism for identifying NPH scripts, perhaps based on the name or location of the script.

5.2. NPH Response

There MUST be a system-defined method for the script to send data back to the server or client; a script MUST always return some data. Unless defined otherwise, this will be the same as for conventional CGI scripts.

Currently, NPH scripts are only defined for HTTP client requests. An (HTTP) NPH script MUST return a complete HTTP response message, currently described in section 6 of the HTTP specifications [1], [4]. The script MUST use the SERVER_PROTOCOL variable to determine the appropriate format for a response. It MUST also take account of any generic or protocol-specific meta-variables in the request as might be mandated by the particular protocol specification.

The server MUST ensure that the script output is sent to the client unmodified. Note that this requires the script to use the correct character set (US-ASCII [9] and ISO 8859-1 [10] for HTTP) in the header fields. The server SHOULD attempt to ensure that the script output is sent directly to the client, with minimal internal and no transport-visible buffering.

Unless the implementation defines otherwise, the script MUST NOT indicate in its response that the client can send further requests over the same connection.

6. CGI Response

6.1. Response Handling

A script MUST always provide a non-empty response, and so there is a system-defined method for it to send this data back to the server. Unless defined otherwise, this will be via the 'standard output' file descriptor.

The script MUST check the REQUEST_METHOD variable when processing the request and preparing its response.

The server MAY implement a timeout period within which data must be received from the script. If a server implementation defines such a timeout and receives no data from a script within the timeout period, the server MAY terminate the script process.

6.2. Response Types

The response comprises a message-header and a message-body, separated by a blank line. The message-header contains one or more header fields. The body may be NULL.

```
generic-response =      1*header-field NL [ response-body ]
```

The script MUST return one of either a document response, a local redirect response or a client redirect (with optional document) response. In the response definitions below, the order of header fields in a response is not significant (despite appearing so in the BNF). The header fields are defined in section 6.3.

```
CGI-Response =          document-response | local-redir-response |
                        client-redir-response | client-redirdoc-response
```

6.2.1. Document Response

The CGI script can return a document to the user in a document response, with an optional error code indicating the success status of the response.

```
document-response       =       Content-Type [ Status ] *other-field NL
                                response-body
```

The script MUST return a Content-Type header field. A Status header field is optional, and status 200 ‘OK’ is assumed if it is omitted. The server MUST make any appropriate modifications to the script’s output to ensure that the response to the client complies with the response protocol version.

6.2.2. Local Redirect Response

The CGI script can return a URI path and query-string (‘local-pathquery’) for a local resource in a Location header field. This indicates to the server that it should reprocess the request using the path specified.

```
local-redir-response = local-Location NL
```

The script MUST NOT return any other header fields or a message-body, and the server MUST generate the response that it would have produced in response to a request containing the URL

```
scheme “://” server-name “:” server-port local-pathquery
```

6.2.3. Client Redirect Response

The CGI script can return an absolute URI path in a Location header field, to indicate to the client that it should reprocess the request using the URI specified.

```
client-redir-response =     client-Location *extension-field NL
```

The script MUST not provide any other header fields, except for server-defined CGI extension fields. For an HTTP client request, the server MUST generate a 302 'Found' HTTP response message.

6.2.4. Client Redirect Response with Document

The CGI script can return an absolute URI path in a Location header field together with an attached document, to indicate to the client that it should reprocess the request using the URI specified.

```
client-redirdoc-response  =   client-Location Status Content-Type
                              *other-field NL response-body
```

The Status header field MUST be supplied and MUST contain a status value of 302 'Found', or it MAY contain an extension-code, that is, another valid status code that means client redirection. The server MUST make any appropriate modifications to the script's output to ensure that the response to the client complies with the response protocol version.

6.3. Response Header Fields

The response header fields are either CGI or extension header fields to be interpreted by the server, or protocol-specific header fields to be included in the response returned to the client. At least one CGI field MUST be supplied; each CGI field MUST NOT appear more than once in the response. The response header fields have the syntax:

```
header-field     =   CGI-field | other-field
CGI-field        =   Content-Type | Location | Status
other-field      =   protocol-field | extension-field
protocol-field   =   generic-field
extension-field  =   generic-field
generic-field    =   field-name ":" [ field-value ] NL
field-name       =   token
field-value      =   *( field-content | LWSP )
field-content    =   *( token | separator | quoted-string )
```

The field-name is not case sensitive. A NULL field value is equivalent to a field not being sent. Note that each header field in a CGI-Response MUST be specified on a single line; CGI/1.1 does not support continuation lines. Whitespace is permitted between the ":" and the field-value (but not between the field-name and the ":"), and also between tokens in the field-value.

6.3.1. Content-Type

The Content-Type response field sets the Internet Media Type [6] of the entity body.

```
Content-Type   =   "Content-Type:" media-type NL
```

If an entity body is returned, the script MUST supply a Content-Type field in the response. If it fails to do so, the server SHOULD NOT attempt to determine the correct content type. The value SHOULD be sent unmodified to the client, except for any charset parameter changes.

Unless it is otherwise system-defined, the default charset assumed by the client for text media-types is ISO-8859-1 if the protocol is HTTP and US-ASCII otherwise. Hence the script SHOULD include a charset parameter. See section 3.4.1 of the HTTP/1.1 specification [4] for a discussion of this issue.

6.3.2. Location

The Location header field is used to specify to the server that the script is returning a reference to a document rather than an actual document (see sections 6.2.3 and 6.2.4). It is either an absolute URI (optionally with a fragment identifier), indicating that the client is to fetch the referenced document, or a local URI path (optionally with a query string), indicating that the server is to fetch the referenced document and return it to the client as the response.

```
Location          =   local-Location | client-Location
client-Location   =   "Location:" fragment-URI NL
local-Location    =   "Location:" local-pathquery NL
fragment-URI      =   absoluteURI[ "#" fragment ]
fragment          =   *uric
local-pathquery   =   abs-path [ "?" query-string ]
abs-path          =   "/" path-segments
path-segments     =   segment *( "/" segment )
segment           =   *pchar
pchar             =   unreserved | escaped | extra
extra             =   ":" | "@" | "&" | "=" | "+" | "$" | ","
```

The syntax of an absoluteURI is incorporated into this document from that specified in RFC 2396 [2] and RFC 2732 [7]. A valid absoluteURI always starts with the name of scheme followed by ":"; scheme names start with a letter and continue with alphanumerics, "+", "-" or ".". The local URI path and query must be an absolute path, and not a relative path or NULL, and hence must start with a "/".

Note that any message-body attached to the request (such as for a POST request) may not be available to the resource that is the target of the redirect.

6.3.3. Status

The Status header field contains a 3-digit integer result code that indicates the level of success of the script's attempt to handle the request.

```
Status            =   "Status:" status-code SP reason-phrase NL
status-code       =   "200" | "302" | "400" | "501" | extension-code
extension-code    =   3digit
reason-phrase     =   *TEXT
```

Status code 200 'OK' indicates success, and is the default value assumed for a document response. Status code 302 'Found' is used with a Location header field and response message-body. Status code 400 'Bad Request' may be used for an unknown request format, such as a missing CONTENT_TYPE. Status code 501 'Not Implemented' may be returned by a script if it receives an unsupported REQUEST_METHOD.

Other valid status codes are listed in section 6.1.1 of the HTTP specifications [1], [4], and also the IANA HTTP Status Code Registry [8] and MAY be used in addition to or instead of the ones listed above. The script SHOULD check the value of SERVER_PROTOCOL before using HTTP/1.1 status codes. The script MAY reject with error 405 'Method Not Allowed' HTTP/1.1 requests made using a method it does not support.

Note that returning an error status code does not have to mean an error condition with the script itself. For example, a script that is invoked as an error handler by the server should return the code appropriate to the server's error condition.

The reason-phrase is a textual description of the error to be returned to the client for human consumption.

6.3.4. Protocol-Specific Header Fields

The script MAY return any other header fields that relate to the response message defined by the specification for the SERVER_PROTOCOL (HTTP/1.0 [1] or HTTP/1.1 [4]). The server MUST translate the header data from the CGI header syntax to the HTTP header syntax if these differ. For example, the character sequence for newline (such as UNIX's US-ASCII LF) used by CGI scripts may not be the same as that used by HTTP (US-ASCII CR followed by LF).

The script MUST NOT return any header fields that relate to client-side communication issues and could affect the server's ability to send the response to the client. The server MAY remove any such header fields returned by the client. It SHOULD resolve any conflicts between header fields returned by the script and header fields that it would otherwise send itself.

6.3.5. Extension Header Fields

There may be additional implementation-defined CGI header fields, whose field names SHOULD begin with "X-CGI-". The server MAY ignore (and delete) any unrecognised header fields with names beginning "X-CGI-" that are received from the script.

6.4. Response Message-Body

The response message-body is an attached document to be returned to the client by the server. The server MUST read all the data provided by the script, until the script signals the end of the message-body by way of an end-of-file condition. The message-body SHOULD be sent unmodified to the client, except for HEAD requests or any required transfer-codings, content-codings or charset conversions.

```
response-body        =        *OCTET
```

7. System Specifications

7.1. AmigaDOS

Meta-Variables

Meta-variables are passed to the script in identically named environment variables. These are accessed by the DOS library routine *GetVar()*. The flags argument SHOULD be 0. Case is ignored, but upper case is recommended for compatibility with case-sensitive systems.

The current working directory

The current working directory for the script is set to the directory containing the script.

Character set

The US-ASCII character set [9] is used for the definition of meta-variables, header fields and values; the newline (NL) sequence is LF; servers SHOULD also accept CR LF as a newline.

7.2. UNIX

For UNIX compatible operating systems, the following are defined:

Meta-Variables

Meta-variables are passed to the script in identically named environment variables. These are accessed by the C library routine *getenv()* or variable *environ*n.

The command line

This is accessed using the argc and argv arguments to *main()*. The words have any characters which are ‘active’ in the Bourne shell escaped with a backslash.

The current working directory

The current working directory for the script SHOULD be set to the directory containing the script.

Character set

The US-ASCII character set [9], excluding NUL, is used for the definition of meta-variables, header fields and CHAR values; TEXT values use ISO-8859-1. The PATH_TRANSLATED value can contain any 8-bit byte except NUL. The newline (NL) sequence is LF; servers should also accept CR LF as a newline.

7.3. EBCDIC/POSIX

For POSIX compatible operating systems using the EBCDIC character set, the following are defined:

Meta-Variables

Meta-variables are passed to the script in identically named environment variables. These are accessed by the C library routine *getenv()*.

The command line

This is accessed using the argc and argv arguments to *main()*. The words have any characters which are 'active' in the Bourne shell escaped with a backslash.

The current working directory

The current working directory for the script SHOULD be set to the directory containing the script.

Character set

The IBM1047 character set [21], excluding NUL, is used for the definition of meta-variables, header fields, values, TEXT strings and the PATH_TRANSLATED value. The newline (NL) sequence is LF; servers should also accept CR LF as a newline.

media-type charset default

The default charset value for text (and other implementation-defined) media types is IBM1047.

8. Implementation

8.1. Recommendations for Servers

Although the server and the CGI script need not be consistent in their handling of URL paths (client URLs and the PATH_INFO data, respectively), server authors may wish to impose consistency. So the server implementation should specify its behaviour for the following cases:

1. define any restrictions on allowed path segments, in particular whether non-terminal NULL segments are permitted;
2. define the behaviour for "." or ".." path segments; i.e., whether they are prohibited, treated as ordinary path segments or interpreted in accordance with the relative URL specification [2];
3. define any limits of the implementation, including limits on path or search string lengths, and limits on the volume of header fields the server will parse.

8.2. Recommendations for Scripts

If the script does not intend processing the PATH_INFO data, then it should reject the request with 404 Not Found if PATH_INFO is not NULL.

If the output of a form is being processed, check that CONTENT_TYPE is "application/x-www-form-urlencoded" [18] or "multipart/form-data" [16]. If

CONTENT_TYPE is blank, the script can reject the request with a 415 'Unsupported Media Type' error, where supported by the protocol.

When parsing PATH_INFO, PATH_TRANSLATED or SCRIPT_NAME the script should be careful of void path segments ("//") and special path segments (".." and "..."). They should either be removed from the path before use in OS system calls, or the request should be rejected with 404 'Not Found'.

When returning header fields, the script should try to send the CGI header fields as soon as possible, and should send them before any HTTP header fields. This may help reduce the server's memory requirements.

Script authors should be aware that the REMOTE_ADDR and REMOTE_HOST meta-variables (see sections 4.1.8 and 4.1.9) may not identify the ultimate source of the request. They identify the client for the immediate request to the server; that client may be a proxy, gateway, or other intermediary acting on behalf of the actual source client.

9. Security Considerations

9.1. Safe Methods

As discussed in the security considerations of the HTTP specifications [1], [4], the convention has been established that the GET and HEAD methods should be 'safe' and 'idempotent' (repeated requests have the same effect as a single request). See section 9.1 of RFC 2616 [4] for a full discussion.

9.2. Header Fields Containing Sensitive Information

Some HTTP header fields may carry sensitive information which the server should not pass on to the script unless explicitly configured to do so. For example, if the server protects the script by using the Basic authentication scheme, then the client will send an Authorization header field containing a username and password. The server validates this information and so it should not pass on the password via the HTTP_AUTHORIZATION meta-variable without careful consideration. This also applies to the Proxy-Authorization header field and the corresponding HTTP_PROXY_AUTHORIZATION meta-variable.

9.3. Data Privacy

Confidential data in a request should be placed in a message-body as part of a POST request, and not placed in the URI or message headers. On some systems, the environment used to pass meta-variables to a script may be visible to other scripts or users. In addition, many existing servers, proxies and clients will permanently record the URI where it might be visible to third parties.

9.4. Information Security Model

For a client connection using TLS, the security model applies between the client and the server, and not between the client and the script. It is the server's responsibility to handle the TLS session, and thus it is the server which is authenticated to the client, not the CGI script.

This specification provides no mechanism for the script to authenticate the server which invoked it. There is no enforced integrity on the CGI request and response messages.

9.5. Script Interference with the Server

The most common implementation of CGI invokes the script as a child process using the same user and group as the server process. It should therefore be ensured that the script cannot interfere with the server process, its configuration, documents or log files.

If the script is executed by calling a function linked in to the server software (either at compile-time or run-time) then precautions should be taken to protect the core memory of the server, or to ensure that untrusted code cannot be executed.

9.6. Data Length and Buffering Considerations

This specification places no limits on the length of the message-body presented to the script. The script should not assume that statically allocated buffers of any size are sufficient to contain the entire submission at one time. Use of a fixed length buffer without careful overflow checking may result in an attacker exploiting 'stack-smashing' or 'stack-overflow' vulnerabilities of the operating system. The script may spool large submissions to disk or other buffering media, but a rapid succession of large submissions may result in denial of service conditions. If the CONTENT_LENGTH of a message-body is larger than resource considerations allow, scripts should respond with an error status appropriate for the protocol version; potentially applicable status codes include 503 'Service Unavailable' (HTTP/1.0 and HTTP/1.1), 413 'Request Entity Too Large' (HTTP/1.1), and 414 'Request-URI Too Large' (HTTP/1.1).

Similar considerations apply to the server's handling of the CGI response from the script. There is no limit on the length of the header or message-body returned by the script; the server should not assume that statically allocated buffers of any size are sufficient to contain the entire response.

9.7. Stateless Processing

The stateless nature of the Web makes each script execution and resource retrieval independent of all others even when multiple requests constitute a single conceptual Web transaction. Because of this, a script should not make any assumptions about the context of the user-agent submitting a request. In particular, scripts should examine data obtained from the client and verify that they are valid, both in form and content, before allowing them to be used for sensitive purposes such as input to other applications, commands, or operating system services. These uses include (but are not limited to) system call arguments, database writes, dynamically evaluated source code, and input to billing or other secure processes. It is important that

applications be protected from invalid input regardless of whether the invalidity is the result of user error, logic error, or malicious action.

Authors of scripts involved in multi-request transactions should be particularly cautious about validating the state information; undesirable effects may result from the substitution of dangerous values for portions of the submission which might otherwise be presumed safe. Subversion of this type occurs when alterations are made to data from a prior stage of the transaction that were not meant to be controlled by the client (e.g., hidden HTML form elements, cookies, embedded URLs, etc.).

9.8. Relative Paths

The server should be careful of ".." path segments in the request URI. These should be removed or resolved in the request URI before it is split into the script-path and extra-path. Alternatively, when the extra-path is used to find the PATH_TRANSLATED, care should be taken to avoid the path resolution from providing translated paths outside an expected path hierarchy.

9.9. Non-parsed Header Output

If a script returns a non-parsed header output, to be interpreted by the client in its native protocol, then the script must address all security considerations relating to that protocol.

10. Acknowledgements

This work is based on the original CGI interface that arose out of discussions on the 'www-talk' mailing list. In particular, Rob McCool, John Franks, Ari Luotonen, George Phillips and Tony Sanders deserve special recognition for their efforts in defining and implementing the early versions of this interface.

This document has also greatly benefited from the comments and suggestions made Chris Adie, Dave Kristol and Mike Meyer; also David Morris, Jeremy Madea, Patrick McManus, Adam Donahue, Ross Patterson and Harald Alvestrand.

11. References

11.1 Normative References

[1] Berners-Lee, T., Fielding, R. and H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996.

[2] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI) : Generic Syntax", RFC 2396, August 1998.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirements Levels", BCP 14, RFC 2119, March 1997.

[4] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[5] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[6] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[7] Hinden, R., Carpenter, B., and L. Masinter, "Format for Literal IPv6 Addresses in URL's", RFC 2732, December 1999.

[8] "HTTP Status Code Registry", http://www.iana.org/assignments/http-status-codes, IANA.

[9] "Information Systems -- Coded Character Sets -- 7-bit American Standard Code for Information Interchange (7-Bit ASCII)", ANSI INCITS.4-1986 (R2002).

[10] "Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1", ISO/IEC 8859-1:1998.

11.2. Informative References

[11] Berners-Lee, T., "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", RFC 1630, June 1994.

[12] Braden, R., Ed., "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, October 1989.

[13] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

[14] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[15] Hinden R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[16] Masinter, L., "Returning Values from Forms: multipart/form-data", RFC 2388, August 1998.

[17] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.

[18] Raggett, D., Le Hors, A., and I. Jacobs, Eds., "HTML 4.01 Specification", W3C Recommendation December 1999, http://www.w3.org/TR/html401/.

[19] Rescola, E. "HTTP Over TLS", RFC 2818, May 2000.

[20] St. Johns, M., “Identification Protocol”, RFC 1413, February 1993.

[21] IBM National Language Support Reference Manual Volume 2, SE09-8002-01, March 1990.

[22] “The Common Gateway Interface”, http://hoohoo.ncsa.uiuc.edu/cgi/, NCSA, University of Illinois.

12. Authors’ Addresses

David Robinson The Apache Software Foundation
EMail: drtr@apache.org
Ken A. L. Coar The Apache Software Foundation
EMail: coar@apache.org
Robinson & Coar Informational [Page 35] RFC 3875 CGI Version 1.1 October 2004

13. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Html markup produced by *rfcmarkup* 1.66, available from
http://tools.ietf.org/tools/rfcmarkup/

Appendix D

The application/json Media Type for JavaScript Object Notation (JSON)

Included, courtesy of:
Network Working Group D. *Crockford*
Request for Comments: 4627 *JSON.org*
Category: Informational *July 2006*

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

JavaScript Object Notation (JSON) is a lightweight, text-based, language-independent data interchange format. It was derived from the ECMAScript Programming Language Standard. JSON defines a small set of formatting rules for the portable representation of structured data.

1. Introduction

JavaScript Object Notation (JSON) is a text format for the serialization of structured data. It is derived from the object literals of JavaScript, as defined in the ECMAScript Programming Language Standard, Third Edition [ECMA].

JSON can represent four primitive types (strings, numbers, booleans, and null) and two structured types (objects and arrays).

A string is a sequence of zero or more Unicode characters [UNICODE].

An object is an unordered collection of zero or more name/value pairs, where a name is a string and a value is a string, number, boolean, null, object, or array.

An array is an ordered sequence of zero or more values.

The terms “object” and “array” come from the conventions of JavaScript.

JSON’s design goals were for it to be minimal, portable, textual, and a subset of JavaScript.

1.1. Conventions Used in This Document

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

The grammatical rules in this document are to be interpreted as described in [RFC4234].

2. JSON Grammar

A JSON text is a sequence of tokens. The set of tokens includes six structural characters, strings, numbers, and three literal names.

A JSON text is a serialized object or array.

```
JSON-text = object / array
```

These are the six structural characters:

```
begin-array     = ws  %x5B  ws  ;  [  left square bracket
begin-object    = ws  %x7B  ws  ;  {  left curly bracket
end-array       = ws  %x5D  ws  ;  ]  right square bracket
end-object      = ws  %x7D  ws  ;  }  right curly bracket
name-separator  = ws  %x3A  ws  ;  :  colon
value-separator = ws  %x2C  ws  ;  ,  comma
```

Insignificant whitespace is allowed before or after any of the six structural characters.

```
ws =    *(
            %x20    /  ;  Space
            %x09    /  ;  Horizontal tab
            %x0A    /  ;  Line feed or New line
            %x0D       ;  Carriage return
        )
```

2.1. Values

A JSON value MUST be an object, array, number, or string, or one of the following three literal names:

false null true

The literal names MUST be lowercase. No other literal names are allowed.

```
value   =  false / null / true / object / array / number / string
false   =  %x66.61.6c.73.65     ;   false
null    =  %x6e.75.6c.6c        ;   null
true    =  %x74.72.75.65        ;   true
```

2.2. Objects

An object structure is represented as a pair of curly brackets surrounding zero or more name/value pairs (or members). A name is a string. A single colon comes after each name, separating the name from the value. A single comma separates a value from a following name. The names within an object SHOULD be unique.

```
object   =  begin-object [ member *( value-separator member ) ] end-object
member   =  string name-separator value
```

2.3. Arrays

An array structure is represented as square brackets surrounding zero or more values (or elements). Elements are separated by commas.

```
array    =  begin-array [ value *( value-separator value ) ] end-array
```

2.4. Numbers

The representation of numbers is similar to that used in most programming languages. A number contains an integer component that may be prefixed with an optional minus sign, which may be followed by a fraction part and/or an exponent part.

Octal and hex forms are not allowed. Leading zeros are not allowed.

A fraction part is a decimal point followed by one or more digits.

An exponent part begins with the letter E in upper or lowercase, which may be followed by a plus or minus sign. The E and optional sign are followed by one or more digits.

Numeric values that cannot be represented as sequences of digits (such as Infinity and NaN) are not permitted.

```
number        =  [ minus ] int [ frac ] [ exp ]
decimal-point =  %x2E          ;   .
digit1-9      =  %x31-39       ;   1-9
e             =  %x65 / %x45   ;   e E
exp           =  e [ minus / plus ] 1*DIGIT
```

```
frac        =   decimal-point 1*DIGIT
int         =   zero / ( digit1-9 *DIGIT )
minus       =   %x2D          ;       -
plus        =   %x2B          ;       +
zero        =   %x30          ;       0
```

2.5. Strings

The representation of strings is similar to conventions used in the C family of programming languages. A string begins and ends with quotation marks. All Unicode characters may be placed within the quotation marks except for the characters that must be escaped: quotation mark, reverse solidus, and the control characters (U+0000 through U+001F).

Any character may be escaped. If the character is in the Basic Multilingual Plane (U+0000 through U+FFFF), then it may be represented as a six-character sequence: a reverse solidus, followed by the lowercase letter *u*, followed by four hexadecimal digits that encode the character's code point. The hexadecimal letters *A* though *F* can be upper or lowercase. So, for example, a string containing only a single reverse solidus character may be represented as "*\u005C*".

Alternatively, there are two-character sequence escape representations of some popular characters. So, for example, a string containing only a single reverse solidus character may be represented more compactly as "\\".

To escape an extended character that is not in the Basic Multilingual Plane, the character is represented as a twelve-character sequence, encoding the UTF-16 surrogate pair. So, for example, a string containing only the *G clef* character (*U+1D11E*) may be represented as "*\uD834\uDD1E*".

```
string      =       quotation-mark *char quotation-mark

char        =       unescaped /
    escape (
    %x22 /          ;       " quotation mark    U+0022
    %x5C/           ; \     reverse solidus     U+005C
    %x2F /          ; /     solidus             U+002F
    %x62/           ; b     backspace           U+0008
    %x66/           ; f     form feed           U+000C
    %x6E /          ; n     line feed           U+000A
    %x72 /          ; r     carriage return     U+000D
    %x74 / ; t      tab     U+0009
    %x75 4HEXDIG ) ;        uXXXX U+XXXX

escape          =   %x5C ; \

quotation-mark  =   %x22 ; "

 unescaped      =   %x20-21 /       %x23-5B         / %x5D-10FFFF
```

3. Encoding

JSON text SHALL be encoded in Unicode. The default encoding is UTF-8.

Since the first two characters of a JSON text will always be ASCII characters [RFC0020], it is possible to determine whether an octet stream is UTF-8, UTF-16 (BE or LE), or UTF-32 (BE or LE) by looking at the pattern of nulls in the first four octets.

00	00	00	xx	UTF-32BE
00	xx	00	xx	UTF-16BE
xx	00	00	00	UTF-32LE
xx	00	xx	00	UTF-16LE
xx	xx	xx	xx	UTF-8

4. Parsers

A JSON parser transforms a JSON text into another representation. A JSON parser MUST accept all texts that conform to the JSON grammar. A JSON parser MAY accept non-JSON forms or extensions.

An implementation may set limits on the size of texts that it accepts. An implementation may set limits on the maximum depth of nesting. An implementation may set limits on the range of numbers. An implementation may set limits on the length and character contents of strings.

5. Generators

A JSON generator produces JSON text. The resulting text MUST strictly conform to the JSON grammar.

6. IANA Considerations

The MIME media type for JSON text is application/json.

Type name: application

Subtype name: json

Required parameters: n/a

Optional parameters: n/a

Encoding considerations: 8bit if UTF-8; binary if UTF-16 or UTF-32

JSON may be represented using UTF-8, UTF-16, or UTF-32. When JSON is written in UTF-8, JSON is 8bit compatible. When JSON is written in UTF-16 or UTF-32, the binary content-transfer-encoding must be used.

Security considerations:

Generally there are security issues with scripting languages. JSON is a subset of JavaScript, but it is a safe subset that excludes assignment and invocation.

A JSON text can be safely passed into JavaScript's eval() function (which compiles and executes a string) if all the characters not enclosed in strings are in the set of characters that form JSON tokens. This can be quickly determined in JavaScript with two regular expressions and calls to the test and replace methods.

```
var my_JSON_object =
        !(/[^,:{}\[\]0-9.\-+Eaeflnr-u \n\r\t]/.test( text.replace(/"(\\.|[^"\\])*"/g, '')))
        && eval('(' + text + ')');
```

Interoperability considerations: n/a

Applications that use this media type:

JSON has been used to exchange data between applications written in all of these programming languages: ActionScript, C, C#, ColdFusion, Common Lisp, E, Erlang, Java, JavaScript, Lua, Objective CAML, Perl, PHP, Python, Rebol, Ruby, and Scheme.

Additional information:

Magic number(s): n/a

File extension(s): .json

Macintosh file type code(s): TEXT

Person & email address to contact for further information:

Douglas Crockford
douglas@crockford.com

Intended usage: COMMON

Restrictions on usage: none

Author:

Douglas Crockford
douglas@crockford.com

Change controller:

Douglas Crockford
douglas@crockford.com

7. Security Considerations

See Security Considerations in Section 6.

8. Examples

This is a JSON object:

```
{
        "Image"          :                    {
        "Width":         800,
        "Height":        600,
        "Title":         "View from 15th Floor",
        "Thumbnail":      {
                         "Url":          "http://www.example.com/image/481989943",
                         "Height":       125,
                         "Width":        "100"
        },
        "IDs": [116, 943, 234, 38793]
        }
}
```

Its Image member is an object whose Thumbnail member is an object and whose IDs member is an array of numbers.

This is a JSON array containing two objects:

```
 [
     {
             "precision" :     "zip",
             "Latitude" :      37.7668,
             "Longitude" :      -122.3959,
             "Address" :        "",
             "City" :           "SAN FRANCISCO",
             "State" :          "CA",
             "Zip" :            "94107",
             "Country" :        "US"
     },
     {
             "precision" :      "zip",
             "Latitude" :       37.371991,
             "Longitude" :     -122.026020,
             "Address" :       "",
             "City" :          "SUNNYVALE",
             "State" :         "CA",
             "Zip" :           "94085",
             "Country" :       "US"
     }

 ]
```

9. References

9.1. Normative References

[ECMA] European Computer Manufacturers Association, "ECMAScript Language Specification 3rd Edition", December 1999, *<http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf>*.

[RFC0020] Cerf, V., "ASCII format for network interchange", RFC 20, October 1969.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[UNICODE] The Unicode Consortium, "The Unicode Standard Version 4.0", 2003, *<http://www.unicode.org/versions/Unicode4.1.0/>*.

Author's Address

Douglas Crockford
JSON.org
EMail: *douglas@crockford.com*

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

Html markup produced by rfcmarkup 1.66, available from *http://tools.ietf.org/tools/rfcmarkup/*

Index