



This Addendum describes new features and software fixes in WebSTAR 1.2. These features and fixes are not described in the printed *WebSTAR Technical Reference*.

Requirements

In addition to requiring System 7, MacTCP, and AppleScript, WebSTAR 1.2 also requires Apple's Thread Manager to run. The Thread Manager Extension is built-in to System 7.5. For earlier versions of System 7, the WebSTAR Installer automatically places the Thread Manager in the Extensions folder.

WebSTAR 1.2 has improved compatibility with Open Transport.

Because of modifications made to memory management in WebSTAR 1.2, an additional 150K memory is required at startup. 1.2MB of application memory is now recommended.

New features

WebSTAR 1.2 includes these new features:

- Connection processing
Lab tests have shown that in a controlled environment, WebSTAR can now process over 500,000 discreet connections per day. This improvement in connection processing is due to more efficient processing of connection setup and teardown, which leads to overall server throughput increases in real world operation of approximately 30%.
- Remote administration
An ACgi application is provided that enables remote administration of the server from any Web browser.
- Server push
“Server push” enables the server to send information in incremental chunks without closing the HTTP connection, which facilitates such services as network monitoring, real-time stock quotes, or simple animations. This feature

Copyright ©1995, StarNine Technologies, Inc. and its Licensors. All rights reserved. Portions Copyright © 1995, Chuck Shotton. StarNine, WebSTAR, ListSTAR, and OmniSTAR are trademarks or registered trademarks of StarNine Technologies, Inc. Apple, the Apple logo, AppleScript, HyperCard, QuickTime, Macintosh, Mac, and MacTCP are trademarks or registered trademarks of Apple Computer, Inc. UNIX is a registered trademark of Novell, Inc. All other trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer.

enables WebSTAR to serve an unlimited amount of data from a CGI (previous versions were limited to 32K).

- Raw file types

A “raw” file is a file that WebSTAR simply opens and dumps the contents of to the client without any header generation or processing.

- Port number display

WebSTAR now displays the port number it is listening on in the title of its Status Window. This is a useful feature when you’re managing more than one server, for example, if you’re running multiple copies of the server across a RAIC (Redundant Array of Inexpensive Computers).

- Localization

WebSTAR is now completely localizable for normal messages (not verbose messages.) All string and HTML info is contained in STR# resources.

- On-line guides in Apple Guide format

If you place the on-line guide files in the same folder as the WebSTAR application, you can access them from the Help menu.

Remote administration

WebSTAR 1.2 includes a simple server configuration HTML page and a new Admin CGI. Together, they enable you to administer most of the essential server functions from any Web browser on the Internet. You can set the number a maximum users, refuse incoming connections, set the default home page, and enable or disable server logging. You can also restart the server remotely.

Server push and raw file types

WebSTAR 1.2 has two significant changes in the way it returns data to clients.

- Server push means that the WebSTAR server maintains an open HTTP connection with a Web client and “pushes” a continuous stream of data from a CGI application through the connection. It does so via a “send partial” Apple event. As long as some data is returned prior to each timeout period expiring, the connection can be kept open indefinitely.
- Raw file types are files of type “RAW!” and creator “WWWΩ”. In WebSTAR 1.2, when the server recognizes a raw file, it opens the files and returns its contents directly to a client, without interpreting the data or generating headers for the file.

Both of these new features have implications for displaying animation on the Web, although that’s not the main reason they’ve been implemented.

Server push

The server push feature relies on the ACGI that is passing data to the server to use the appropriate Apple events and to keep track of multiple connections, either via the Thread Manager or by maintaining a separate context for each connection. The reason is that new connections can come in for “pushing” while one or more old connections are still being pushed.

See “Including server push functions in an ACGI” on page 4 for details.

The server push feature can be used to serve any real-time data. For example, it can be used as a way to serve simple animations, where a series of images is stored in multiple files, which are sequentially sent to the client, displayed, and then replaced by the subsequent image. Or, server push can be used to incrementally return results from an ACGI that takes a long time to calculate an answer.

Raw file types

Raw file types are recognized by the server before any suffix mapping is verified. If WebSTAR receives a URL request and finds a type and creator indicating a raw file, it simply returns the file’s contents as is. You can create raw files that contain any kind of data.

For example, suppose you move an entire branch of your document tree to another Macintosh. You can redirect incoming requests simply by creating a raw file that contains a URL redirect header and new location. Replace the index file in that subdirectory with the raw file, and client requests will retrieve the raw file, redirecting them to the new host.

Raw files can also be used (with help from a C program to build it) to generate multi-part animation files that can be returned from a single file without the need for a CGI! In fact, raw files can be used to replace a CGI if you need to take the script off-line for work for some reason.

Apple event and ACGI changes

Several new Apple events are supported. This section describes those events and shows how to include a “send partial” event in an ACGI application.

New Apple events

These new Apple events are supported:

- `version` is a new Apple event property. `version` is a read-only character string that contains the version number of WebSTAR in it. The property's 4 character keyword is `Pvrs`.

- The connection ID is now passed to CGIs and ACGIs as an additional parameter. The connection ID is the unique identifier used by WebSTAR internally to track each connection. It is unique for each and every connection for as long as WebSTAR is running. The data type of this parameter is a long integer and the Apple event keyword value is `Kcid`.
- The Send Partial Apple event is new. This event is designed to support the return of large amounts of data from ACGIs and to implement true server push functions.

The WebSTAR Apple event dictionary describes the syntax of this event:

```
send partial: Sends partial ACGI execution results to WebSTAR
for transmission to clients

    send partial    'char'    -- data to return to the WWW client
    connection      integer    -- connection ID to return data to.
    more             boolean    -- is there more data to send?
```

Here's the data necessary to generate your own Apple events from C or other languages:

```
#define kMyAESendPartial    'SPar' //event code. The suite is
'WWWΩ'

//keywords
// data is sent in the direct parameter, '----'
#define kConnectionIDKeyword 'Kcid'
#define kMoreKeyword         'Kmor'
```

As you can see, this is why the connection ID parameter was added to the `WWWΩsdoc` event, as it is used to join up data sent via the Send Partial event with the connection that WebSTAR is holding open to the client.

These new additions can be used to implement ACGIs that return large blocks of data by dividing the data up across multiple Send Partial calls back to WebSTAR. They can also be used to implement server push as described at:

http://home.netscape.com/assist/net_sites/pushpull.html

NOTE: AppleScript turns out to be a poor language for implementing server push applications, because it is impossible to reply to the initial `sdoc` event without terminating the execution of the `sdoc` handler within the AppleScript.

Including server push functions in an ACGI

An ACGI has to inform the WebSTAR server that it intends to return its results in pieces over a period of time, rather than simply lumping it all into the value

returned from the `WWWΩsdoc` event. WebSTAR examines the results from an ACGI's reply to see if it matches the string:

`<SEND_PARTIAL>`

If this 14 character string is matched exactly (case sensitive), then WebSTAR doesn't send anything to the client and keeps the connection open until the timeout period expires or WebSTAR receives data via the Send Partial event for that connection.

Send Partial events received before the `<SEND_PARTIAL>` response to the `sdoc` event are also a legal way to indicate that server push functions are to be performed for a given connection.

As long as Send Partial events are received for a given connection, the timeout timer is restarted and the data is sent to the client. If the Send Partial event's "more" parameter is `FALSE`, the server closes the connection and assumes that the ACGI has finished sending data.

To send events back to WebSTAR from a language like C or Pascal, you must extract and save the "from" Apple event attribute from the reply event sent to your `WWWΩsdoc` handler. The "from" event contains the `AEAddressDesc` used to address the "SPar" events to WebSTAR.

Here's the general flow of events

1. A WWW client sends an ACGI URL request to WebSTAR.
2. WebSTAR sends a `WWWΩsdoc` event to the ACGI, passing the connection ID.
3. The ACGI decides it needs to return data in pieces, so it replies to the `sdoc` event with the string `<SEND_PARTIAL>` and saves the connection ID.
4. WebSTAR sees that the ACGI wants to return partial data, so it sets a flag indicating that the connection should be checked periodically for timeouts and resets the timer.
5. The ACGI sends a Send Partial event to WebSTAR with the first chunk of data, the connection ID, and the "more" flag set to `TRUE`, indicating more data to come.
6. WebSTAR handles the Send Partial event, finds the requested connection, and queues up the data for transmission to the client (allowing the thread owning the connection to schedule the transmission.) WebSTAR then resets the timeout for the connection.

Steps 5 and 6 repeat at whatever interval the client decides until the "more" parameter is false. WebSTAR receives the final Send Partial event with the "more" flag set to false, sends the accompanying data, and closes the client connection.

Software fixes

- A work-around has been added for the CERN proxy problem with \$ as a path arg separator. The first \$ or encoded \$ (%24) found in a URL is treated as the path arg separator now. That means that Mac file names containing the \$ character will be improperly interpreted, making \$ an illegal character for Mac file names.
- DNS look-ups that fail will now return the simple IP address of the target instead of the in-addr.arpa-style responses.
- Setting the timeout value to an out-of-bounds value resets the value to the closest bounds, rather than resetting it to the default of 90. So setting timeout to 750 actually sets it to the max of 600 rather than the default of 90.
- Behavior for client timeouts has been changed. WebSTAR now aborts client connections that time out, rather than attempting a graceful close. This speeds termination under Open Transport substantially.
- Timeouts that occur with ACGIs now return a 504 Gateway Timeout error to the client instead of simply closing the connection. This eliminates the problem of failed ACGIs returning a blank page to the client. Timed out ACGIs that were in the middle of a server push operation (Send Partial) simply have their connection gracefully closed. As WebSTAR cannot determine what the ACGI has already communicated to the client, it cannot send the 504 message in this case.
- The “halting” problem, where the server went into an intermittent hang or slowdown that corrected itself after a short time, has been corrected. TCP Close requests now happen asynchronously.
- WebSTAR correctly handles “malformed URLs” to directories that do not contain a trailing slash character (/). WebSTAR redirects the client to a correct URL with the trailing / included.