

# Advanced WEB Server Serving Pages Containing More Than 8 K Bytes of Page Data

## Observed Symptom

You have created a web page containing large blocks of page data (blocks > 8 k bytes). You used the comment tag `<!--RpDisplayText -->` to identify your page data to the Advanced Web Server. You have found that your page was not displayed correctly. In some cases, the Advanced Web Server may have become nonresponsive.

## Cause

The `<!--RpDisplayText -->` comment tag is not the correct comment tag to use when including large amounts of page data in your html files. The correct comment tag to use in such applications is `<!-- RpBufferDisplay -->`. We have tested the `<!-- RpBufferDisplay -->` comment tag in applications involving multiple 10's of kilobytes of page data and have proven that all page data is successfully served from the Advanced Web Server to the browser.

It should be noted that through an oversight, documentation describing the `<!-- RpBufferDisplay -->` comment tag was left out of the Advanced Web Server Toolkit documentation. This problem is being resolved. A description of the `<!--RpBufferDisplay -->` comment tag is included at the end of this document.

## Conclusion

Web pages, to be served by the Advanced Web Server and that include large amounts of data, must utilize the `<!--RpDisplayBuffer -->` comment tag in place of `<!--RpDisplayText -->`. Currently existing applications should be updated to reflect this change. This will ensure successful serving of the pages. Additionally, continued use of the `<!--RpDisplayText -->` comment tag for large amounts of page data, will cause unpredictable results.

### HTML comment tag

```
<!--RpBufferDisplay Name = n RpGetPtr=gp -->
```

The `RpBufferDisplay` tag can be used instead of the `RpDisplayText` tag when very large buffers of data need to be inserted into a Web page. The access function named by the `RpGetPtr` keyword will be called with the Web server data pointer, the name specified by the `Name` keyword and a pointer to the current query index values. The access function returns a pointer to the buffer to be inserted into the Web page and a length. The access function also returns a Boolean value that indicates whether the call is complete or not. If multiple buffers of data need to be prepared, the call should return `False` to indicate that the access function is not complete and needs to be called again. If the access function is complete, it should return `True` to indicate that it is done.

The buffer that is returned by the access function needs to persist until

the Web server is finished sending it. If the application has dynamically allocated the buffer and needs to release the buffer, it should return False when the buffer is given to the Web server and then return True with a length of 0, after the buffer has been released.