

# FileSystem.FindFileBySizeAndHash

<b>Method</b>	<b>FindFileBySizeAndHash</b>
<b>Module</b>	FileSystem
<b>Library</b>	Core
<b>Action</b>	Finds a file by its size and hash.
<b>Parameters</b>	<p><code>Size</code> (int): the size in bytes of the file.</p> <p><code>Hash</code> (string): the hash of the file.</p> <p><code>Fast</code> (boolean; optional, default <code>true</code>): On Windows chooses between the fast search technology or the previous slower (but fully reliable) technology. Always used for non-Windows.</p> <p><code>Algorithm</code> (string, optional, default "SHA256"): Selects the hashing algorithm used by this method. Supported values:</p> <ul style="list-style-type: none"><li>• SHA1</li><li>• SHA256</li><li>• MD5</li></ul> <p>This must match the <code>Hash</code> parameter; if you specify an MD5 hash, this parameter must also be set to MD5.</p> <p><code>TimeoutSecs</code> (integer; optional, default 600): If the operation has not completed within the specified time period, it will fail with an error. <b>New in v5.2.</b></p>
<b>Return values</b>	<p>For each file found:</p> <ul style="list-style-type: none"><li>• <code>FileName</code> (string): The full path.</li></ul>
<b>Example</b>	<p>This will perform a fast search, looking for a file with the specified SHA-256 hash and size:</p> <pre>FileSystem.FindFileBySizeAndHash(Size:402, Hash:"8eda0d394c2370c4ac8562d5ae6b5e7b01f347940bc58a65ca873c33a33475a3");</pre> <p>This will perform a slow search, looking for a file with the specified MD5 hash and size:</p> <pre>FileSystem.FindFileBySizeAndHash(Size:999, Hash:"08ad2e5d7538091fbbfd352bb323e238", Fast:false, Algorithm:"MD5");</pre>
<b>Platforms</b>	<ul style="list-style-type: none"><li>• Windows</li><li>• Linux</li><li>• MacOS</li><li>• Solaris Intel</li><li>• Solaris Sparc</li></ul>
<b>Notes</b>	<p>This will search all fixed disks which is an expensive process and may take some time.</p> <p>Currently files locked by some other process cannot be hashed and therefore cannot be reported as being found. This is not supported.</p> <p>This will not search for files on remote machines.</p> <p>This works slightly differently on Windows and Non-Windows platforms when the given file or directory where the file resides does not have any permissions assigned/available. Windows platforms do not report such locked-down files whereas non-Windows do report the files irrespective of which level permission has been denied (file or directory). This happens because the 1E.Client service runs as "root" (superuser).</p>