

# fn\_vbs\_aw\_getBleedingWounds



<b>Introduced in</b>	
Version:	3.7
<b>Description</b>	
Description:	Returns all bleeding wounds for a unit. Requires the Advanced Wounding Editor Object to be present in the mission.
<b>Syntax</b>	
Syntax:	[unit, bodypart] call <b>fn_vbs_aw_getBleedingWounds</b>
Parameters:	<ul style="list-style-type: none"><li>unit: <a href="#">Object</a> - Soldier to check.</li><li>bodypart: <a href="#">String</a> or <a href="#">Number</a> - Name or ID of <a href="#">Wound</a> to check. If -1 or a non-existent body part name is passed, then all wounds are returned (optional, default: -1).</li></ul>
Return Value:	<a href="#">Array</a> of <a href="#">Wound</a>
<b>Warning</b>	
When using functions in VBS versions <b>older than 3.4</b> , certain limitations and requirements should be kept in mind - the main ones being that capitalization of the function name is crucial, and that <b>#include "vbs2\headers\function_library.hpp"</b> has to be included in every script that utilizes it. More <a href="#">Category VBS Scripting Functions</a> .	
<b>Examples</b>	
Examples:	<pre>_wounds = [player] call fn_vbs_aw_getBleedingWounds;           // Check all wounds _wounds = [player,2] call fn_vbs_aw_getBleedingWounds;        // Check left leg (using index) _wounds = [player,"leftleg"] call fn_vbs_aw_getBleedingWounds; // Check left leg (using string)</pre>
<b>Additional Information</b>	
See also:	<a href="#">fn vbs aw getWounds</a> , <a href="#">fn vbs aw hasWound</a>
Multiplayer:	
Problems:	

## Notes