

# nearestCostPosition



<b>Introduced in</b>	
<b>Version:</b>	1.31
<b>Description</b>	
<b>Description:</b>	Returns the center position of the nearest cost field, of the given cost.
<b>Syntax</b>	
<b>Syntax:</b>	<code>nearestCostPosition [position, costtypes, radius, unit]</code>
<b>Parameters:</b>	<ul style="list-style-type: none"><li>• position: <a href="#">Position-2</a> - Position from which the search starts.</li><li>• costtypes: <a href="#">Array of String</a> - Cost types to search for. Can be:<ul style="list-style-type: none"><li>• "Avoid"</li><li>• "AvoidBush"</li><li>• "AvoidTree"</li><li>• "Normal"</li><li>• "Road"</li><li>• "RoadForced"</li><li>• "Space"</li><li>• "SpaceBush"</li><li>• "SpaceHardBush"</li><li>• "SpaceRoad"</li><li>• "SpaceTree"</li><li>• "Water".</li></ul></li><li>• radius: <a href="#">Number</a> - Maximum radius to check around <code>position</code></li><li>• unit: <a href="#">Object</a> - If specified, restricted areas of this unit are also considered (optional).</li></ul>
<b>Return Value:</b>	<a href="#">Position-2</a> - Nearest position found. If no qualified position is found, an empty array is returned.
<b>Examples</b>	
<b>Examples:</b>	<pre>nearestCostPosition [getpos person1, ["Normal", "Space"], 20, person2] // Looks for nearest fields with costs of "Normal" or "Space" 20 meters around unit person1, and takes also restricted areas of unit person2 into account.</pre>
<b>Additional Information</b>	
<b>See also:</b>	<a href="#">cost</a> , <a href="#">assignAsRestrictedArea</a>
<b>Multiplayer:</b>	
<b>Problems:</b>	

## Notes

