

FindInArray v1.11

Function that finds item in an array; by Faguss (ofp-faguss.com)

1. Overview

Operation Flashpoint 1.96 does not have a native command for searching arrays and you end up having to write the algorithm yourself. This package provides a robust search function.

Version 1.99 of the game features `find` command which works much faster and should be used instead. However, it's case sensitive so if you're looking for a case insensitive search you still might want to use **FindInArray**.

Included is also a variant of the function that's portable between 1.96 and 1.99.

2. Usage

- Copy `FindInArray.sqf` to the mission directory.
- Write in `Init.sqs`:

```
FindInArray = preProcessFile "FindInArray.sqf"
```

Function call syntax:

```
[<search FOR>, <search IN>, <optional mode>, <opt position>] call FindInArray
```

Where `<search FOR>` is the item you're looking for and `<search IN>` is the array that the function will go through.

Return value is an integer – index of the first occurrence of the item in the array. If it was not found then `-1` is returned.

For example:

```
_index = ["c", ["a","b","c"]] call FindInArray  
hint Format ["%1", _index]
```

Game will display:

```
2
```

3. Optional Comparison Method

By default function uses `a==b` to compare elements. It's case insensitive and will generate error if the data types aren't identical (e.g. you're searching for a string in an array containing numbers).

The alternative is to use `a in [b]` which is safer (no error messages) but case sensitive. Pass argument `"in"` to enable it. For example:

```
["c", [0,1,"c"], "in"] call FindInArray
```

Another way to avoid error is to convert both elements to string using `Format` command so that the comparison will be: `Format["%1",a] == Format["%1",b]`. It proves useful when searching for a number in an array with numbers written as strings or vice versa. For example:

```
[2, ["0","1","2"], "format"] call FindInArray
```

You can combine both methods by using `formatin` argument. For example:

```
[item,_array, "formatIn"] call FindInArray
```

4. Optional Starting Position

By default search is conducted from the beginning of the array (index zero). To start from any other position pass additional number. For example:

```
_index = ["c", ["a", "b", "c", "d", "e", "c"], "", 3] call FindInArray  
hint format ["%1", _index]
```

Game will display:

```
5
```

5. OFP-CWA Variant

This is a simpler version that works identically on both OFP and CWA. It uses `find` command on 1.99 and `foreach` on 1.96. Case sensitive.

- Copy `FindInArray_OFPCWA.sqf` to the mission directory.
- Write in `Init.sqs`:

```
IS_CWA = Format ["%1",getworld]!="scalar bool array string 0xfcffffef"  
FindInArray_OFPCWA = preProcessFile "FindInArray_OFPCWA.sqf"
```

Function call syntax:

```
[<search FOR>, <search IN>] call FindInArray OFPCWA
```

6. Version history

1.0 (24.08.14)

First release.

1.1 (04.07.21)

- replaced `while do` with `foreach` which is faster and has no item limit
- added `FindInArray_CWA` variant

1.11 (09.07.21)

- fixed optional starting position not working
- renamed `FindInArray_CWA` to `FindInArray_OFPCWA`