

```
*****
4856 Sun Feb 5 13:41:02 2017
new/usr/src/man/man9f/vmem_create.9f
more copy-editting from Josh
*****
```

```
1 .\"
2 .\" This file and its contents are supplied under the terms of the
3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
4 .\" You may only use this file in accordance with the terms of version
5 .\" 1.0 of the CDDL.
6 .\"
7 .\" A full copy of the text of the CDDL should have accompanied this
8 .\" source. A copy of the CDDL is also available via the Internet at
9 .\" http://www.illumos.org/license/CDDL.
10 .\"
11 .\"
12 .\" Copyright 2017, Richard Lowe.
13 .\"
14 .Dd Jan 18, 2017
15 .Dt VMEM_CREATE 9F
16 .Os
17 .Sh NAME
18 .Nm vmem_create ,
19 .Nm vmem_xcreate ,
20 .Nm vmem_destroy
21 .Nd create and destroy vmem arenas
22 .Sh SYNOPSIS
23 .In sys/vmem.h
24 .Vt "typedef struct vmem vmem_t;"
25 .Vt "typedef void *(vmem_alloc_t)(vmem_t *, size_t, int);"
26 .Vt "typedef void (vmem_free_t)(vmem_t *, void *, size_t);"
27 .Vt "typedef void *(vmem_ximport_t)(vmem_t *, size_t *, size_t, int);"
28 .Ft vmem_t *
29 .Fo vmem_create
30 .Fa "const char *name"
31 .Fa "void *base"
32 .Fa "size_t size"
33 .Fa "size_t quantum"
34 .Fa "vmem_alloc_t *afunc"
35 .Fa "vmem_free_t *ffunc"
36 .Fa "vmem_t *source"
37 .Fa "size_t qcache_max"
38 .Fa "int vmflag"
39 .Fc
40 .Ft vmem_t *
41 .Fo vmem_xcreate
42 .Fa "const char *name"
43 .Fa "void *base"
44 .Fa "size_t size"
45 .Fa "size_t quantum"
46 .Fa "vmem_ximport_t *afunc"
47 .Fa "vmem_free_t *ffunc"
48 .Fa "vmem_t *source"
49 .Fa "size_t qcache_max"
50 .Fa "int vmflag"
51 .Fc
52 .Ft void
53 .Fo vmem_destroy
54 .Fa "vmem_t *vmp"
55 .Fc
56 .Sh INTERFACE LEVEL
57 illumos DDI specific
58 .Sh PARAMETERS
59 .Bl -tag -width Ds
60 .It Fa name
61 A character string giving a name to the vmem
```

```
62 arena to be created.
63 .It Fa base
64 An address indicating the lowest possible value in the arena.
65 .It Fa size
66 The size of the arena to create.
67 .It Fa quantum
68 The arena's
69 .Dq quantum .
70 The granularity of the arena. The amount allocated at minimum by each
71 request. Must be a power of 2.
72 .It Fa afunc
73 A function which is called to import new spans from
74 .Fa source ,
75 which may be
76 .Dv NULL
77 if this arena does not import from another.
78 When calling
79 .Fn vmem_create ,
80 .Fa afunc
81 is a
82 .Vt vmem_alloc_t ,
81 is an
82 .Vt vmem_alloc_t
83 a function taking three parameters and returning a pointer to
84 .Vt void
85 (the imported space):
86 .Bl -tag -width Ds
87 .It Fa "vmem_t *"
88 The source arena from which we'll import. The
89 .Fa source
90 argument to
91 .Fn vmem_create .
92 .It Fa size_t
93 The size to import.
93 The size to import
94 .It Fa int
95 The
96 .Fa vmflag
97 argument used for the import.
98 .El
99 .Pp
100 When calling
101 .Fn vmem_xcreate ,
101 .Fn vmem_xcreate
102 .Fa afunc
103 is a
104 .Vt vmem_ximport_t ,
103 is an
104 .Vt vmem_ximport_t
105 a function taking four parameters and returning a pointer to
106 .Vt void
107 (the imported space):
108 .Bl -tag -width Ds
109 .It Fa "vmem_t *"
110 The source arena from which we'll import. The
111 .Fa source
112 argument to
113 .Fn vmem_xcreate .
114 .It Fa "size_t *"
115 The size of the import.
115 The size of the import,
116 .Fa afunc
117 may
118 .Em increase
119 this size if that is desirable, but must never decrease it.
```

```

120 .It Fa size_t
121 The desired alignment of the imported space.
122 .It Fa int
123 The
123 the
124 .Fa vmflag
125 argument used for the import.
126 .El
127 .It Fa ffunc
128 A function which is called to return spans to
129 .Fa source ,
130 which may be
131 .Dv NULL
132 if this arena does not import from another.
133 This is a
134 .Vt vmem_free_t ,
135 a function taking three parameters and returning void:
134 .Vt vmem_free_t
135 a function taking three parametes and returning void:
136 .Bl -tag -width Ds
137 .It Fa "vmem_t"
138 The arena to which space is being returned. The
139 .Fa source
140 argument to
141 .Fn vmem_create
142 or
143 .Fn vmem_xcreate .
144 .It Fa "void *"
145 The span being returned to the source arena.
146 .It Fa "size_t"
147 The size of the span being returned to the source arena.
148 .El
149 .It Fa source
150 An arena from which this arena will import,
151 which may be
152 .Dv NULL
153 if this arena does not import from another.
154 .It Fa qcache_max
155 Each arena offers caching of integer multiples of
156 .Fa quantum
157 up to
158 .Fa qcache_max ,
159 which may be 0.
160 .It Fa vmflag
161 A bitmask of flags indicating the characteristics of this arena.
162 .Bl -tag -width Ds
163 .It Dv VMC_IDENTIFIER
164 The arena represents arbitrary integer identifiers, rather than virtual
165 memory.
166 .El
167 .It Fa vmp
168 A pointer to the vmem arena to be destroyed.
169 .El
170 .Sh DESCRIPTION
171 A
172 .Em vmem arena
173 is a section of an arbitrary address space (a range of integer addresses).
174 This commonly represents virtual memory, but can in fact be an arbitrary set
175 of integers. The
176 .Dv VMC_IDENTIFIER
177 flag set at arena creation time differentiates between these two cases.
178 .Pp
179 The
180 .Fa afunc ,
181 .Fa ffunc , and
182 .Fa source

```

```

183 arguments combine to support a hierarchical structure of arenas, each
184 importing from a single parent (the
185 .Fa source ) .
186 The
187 .Fn vmem_create
188 and
189 .Fn vmem_xcreate
190 functions differ in that the latter provides an interface for
191 .Fa afunc
192 to alter the size of the span imported from
193 .Fa source .
194 It is only legal to
195 .Em increase
196 this size.
197 .Sh CONTEXT
198 These functions can be called from user or kernel context.
199 .Sh RETURN VALUES
200 Upon successful completion the
201 .Fn vmem_create
202 and
203 .Fn vmem_xcreate
204 functions return a pointer to a vmem arena. Otherwise,
205 .Dv NULL
206 is returned to indicate the arena could not be created.
207 .Sh SEE ALSO
208 .Xr vmem 9 ,
209 .Xr vmem_add 9F ,
210 .Xr vmem_alloc 9F

```