```
**********************************************************
    7461 Wed Jan 15 13:15:35 2020
new/usr/src/man/man3pool/pool_associate.3pool
12198 typo paramter in some section 3pool man pages
**********************************************************
’\" te
.\" Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH POOL_ASSOCIATE 3POOL "January 15, 2020"
.TH POOL_ASSOCIATE 3POOL "April 9, 2016"
.SH NAME
pool_associate, pool_create, pool_destroy, pool_dissociate, pool_info,
pool_query_pool_resources \- resource pool manipulation functions
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR\&.\|.\|. ] \fIfile\fR\&.\|.\|. \fB-lpool\fR [ \fIlibrary\fR\&.\|
#include <pool.h>

\fBint\fR \fBpool_associate\fR(\fBpool_conf_t *\fR\fIconf\fR, \fB pool_t *\fR\fI
     \fBpool_resource_t *\fR\fIresource\fR);
.fi

.LP
.nf
\fBpool_t *\fR\fBpool_create\fR(\fBpool_conf_t *\fR\fIconf\fR, \fBconst char *\f
.fi

.LP
.nf
\fBint\fR \fBpool_destroy\fR(\fBpool_conf_t *\fR\fIconf\fR, \fBpool_t *\fR\fIpoo
.fi

.LP
.nf
\fBint\fR \fBpool_dissociate\fR(\fBpool_conf_t *\fR\fIconf\fR, \fBpool_t *\fR\fI
     \fBpool_resource_t *\fR\fIresource\fR);
.fi

.LP
.nf
\fBconst char *\fR\fBpool_info\fR(\fBpool_conf_t *\fR\fIconf\fR, \fBpool_t *\fR\
     \fBint\fR \fIflags\fR);
.fi

.LP
.nf
\fBpool_resource_t **\fR\fBpool_query_pool_resources\fR(\fBpool_conf_t *\fR\fIco
     \fBpool_t *\fR\fIpool\fR, \fBuint_t *\fR\fInelem\fR, \fBpool_value_t **\fR\
.fi

.SH DESCRIPTION
.LP
These functions provide mechanisms for constructing and modifying pools entries
within a target pools configuration.  The \fIconf\fR argument for each function
refers to the target configuration to which the operation applies.
.sp
.LP
The \fBpool_associate()\fR function associates the specified resource with
\fIpool\fR. A resource can be associated with multiple pools at the same time.
Any resource of this type that was formerly associated with this pool is no
longer associated with the pool. The new association replaces the earlier one.
.sp
.LP
```

```
The \fBpool_create()\fR function creates a new pool with the supplied name with
its default properties initialized, and associated with the default resource of
each type.
.sp
.LP
The \fBpool_destroy function()\fR destroys the given pool association.
Associated resources are not modified.
.sp
.LP
The \fBpool_dissociate()\fR function removes the association between the given
resource and pool. On successful completion, the pool is associated with the
default resource of the same type.
.sp
.LP
The \fBpool_info()\fR function returns a string describing the given pool. The
string is allocated with \fBmalloc\fR(3C). The caller is responsible for freeing
the returned string. If the \fIflags\fR option is non-zero, the string returned
also describes the associated resources of the pool.
.sp
.LP
The \fBpool_query_pool_resources()\fR function returns a null-terminated array
of resources currently associated with the pool that match the given list of
properties. The return value must be freed by the caller. The \fInelem\fR
argument is set to be the length of the array returned.
.SH RETURN VALUES
.LP
Upon successful completion, \fBpool_create()\fR returns a new initialized pool.
Otherwise it returns \fINULL\fR and \fBpool_error\fR(3POOL) returns the
pool-specific error value.
.sp
.LP
Upon successful completion, \fBpool_associate()\fR, \fBpool_destroy()\fR, and
\fBpool_dissociate()\fR return 0. Otherwise, they return -1 and
\fBpool_error()\fR returns the pool-specific error value.
.sp
.LP
Upon successful completion, \fBpool_info()\fR returns a string describing the
given pool. Otherwise it returns \fINULL\fR and \fBpool_error()\fR returns the
pool-specific error value.
.sp
.LP
Upon successful completion, \fBpool_query_pool_resources()\fR returns a
null-terminated array of resources. Otherwise it returns \fINULL\fR and
\fBpool_error()\fR returns the pool-specific error value.
.SH ERRORS
.LP
The \fBpool_create()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR or \fIname\fR is
already in use.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE
```

```
.sp
.ne 2
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The pool element could not be created because the configuration would be
invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_PUTPROP\fR\fR
.ad
.RS 20n
One of the supplied properties could not be set.
.RE

.sp
.LP
The \fBpool_destroy()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 16n
The supplied configuration’s status is not \fBPOF_VALID\fR.
.RE

.sp
.LP
The \fBpool_associate()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 16n
The supplied configuration’s status is not \fBPOF_VALID\fR or the parameters
are supplied from a different configuration.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 16n
A system error has occurred. Check the system error code for more details.
.RE

.sp
.LP
The \fBpool_disassociate()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR or the parameters
are supplied from a different configuration.
.RE
```

```
.sp
.ne 2
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
No resources could be located for the supplied configuration or the supplied
configuration is not valid (for example, more than one default for a resource
type was found.)
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE

.sp
.LP
The \fBpool_info()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR or the \fIflags\fR
parameter is neither 0 or 1.
paramter is neither 0 or 1.
.RE

.sp
.ne 2
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The configuration is invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE

.sp
.LP
The \fBpool_query_pool_resources()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR.
.RE

.sp
.ne 2
```

```
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The configuration is invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE

.SH USAGE
.LP
Pool names are unique across pools in a given configuration file. It is an
error to attempt to create a pool with a name that is currently used by another
pool within the same configuration.
.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled
_
Interface Stability     Unstable
_
MT-Level        Safe
.TE

.SH SEE ALSO
.LP
\fBlibpool\fR(3LIB), \fBpool_error\fR(3POOL), \fBattributes\fR(5)
```

```
**********************************************************
    3448 Wed Jan 15 13:15:35 2020
new/usr/src/man/man3pool/pool_component_info.3pool
12198 typo paramter in some section 3pool man pages
**********************************************************
'\" te
.\" Copyright (c) 2003, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH POOL_COMPONENT_INFO 3POOL "January 15, 2020"
.TH POOL_COMPONENT_INFO 3POOL "April 9, 2016"
.SH NAME
pool_component_info, pool_get_owning_resource \- resource pool component
functions
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR&.\|.\|. ] \fIfile\fR&.\|.\|. \fB-lpool\fR [ \fIlibrary\fR&.\|
#include <pool.h>

\fBconst char *\fR\fBpool_component_info\fR(\fBpool_conf_t *\fR\fIconf\fR,
     \fBpool_component_t *\fR\fIcomponent\fR, \fBint\fR \fIflags\fR);
.fi

.LP
.nf
\fBpool_resource_t *\fR\fBpool_get_owning_resource\fR(\fBpool_conf_t *\fR\fIconf
     \fBpool_component_t *\fR\fIcomponent\fR);
.fi

.SH DESCRIPTION
.LP
Certain resources, such as processor sets, are composed of resource components.
Informational and ownership attributes of resource components are made
available with the \fBpool_component_info()\fR and
\fBpool_get_owning_resource()\fR functions. The \fIconf\fR argument for each
function refers to the target configuration to which the operation applies.
.sp
.LP
The \fBpool_component_info()\fR function returns a string describing
\fIcomponent\fR. The string is allocated with \fBmalloc\fR(3C). The caller is
responsible for freeing the returned string. The \fIflags\fR argument is
ignored.
.sp
.LP
The \fBpool_get_owning_resource()\fR function returns the resource currently
containing \fIcomponent\fR.  Every component is contained by a resource.
.SH RETURN VALUES
.LP
Upon successful completion, \fBpool_component_info()\fR returns a string.
Otherwise it returns \fINULL\fR and \fBpool_error\fR(3POOL) returns the
pool-specific error value.
.sp
.LP
Upon successful completion, \fBpool_get_owning_resource()\fR returns the owning
resource. Otherwise it returns \fINULL\fR and \fBpool_error()\fR returns the
pool-specific error value.
.SH ERRORS
.LP
The \fBpool_component_info()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
```

```
.RS 20n
The supplied configuration's status is not \fBPOF_VALID\fR or the \fIflags\fR
parameter is neither 0 or 1.
paramter is neither 0 or 1.
.RE

.sp
.ne 2
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The configuration is invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE

.sp
.LP
The \fBpool_get_owning_resource()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 16n
The supplied configuration's status is not \fBPOF_VALID\fR.
.RE

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled
_
Interface Stability     Unstable
_
MT-Level        Safe
.TE

.SH SEE ALSO
.LP
\fBlibpool\fR(3LIB), \fBpool_error\fR(3POOL), \fBattributes\fR(5)
```

```
**********************************************************
    8410 Wed Jan 15 13:15:35 2020
new/usr/src/man/man3pool/pool_resource_create.3pool
12198 typo paramter in some section 3pool man pages
**********************************************************
'\" te
.\" Copyright (c) 2003, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH POOL_RESOURCE_CREATE 3POOL "January 15, 2020"
.TH POOL_RESOURCE_CREATE 3POOL "April 9, 2016"
.SH NAME
pool_resource_create, pool_resource_destroy, pool_resource_info,
pool_query_resource_components, pool_resource_transfer, pool_resource_xtransfer
\- resource pool resource manipulation functions
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR&.\|.\|. ] \fIfile\fR&.\|.\|. \fB-lpool\fR [ \fIlibrary\fR&.\|
#include <pool.h>

\fBpool_resource_t *\fR\fBpool_resource_create\fR(\fBpool_conf_t *\fR\fIconf\fR,
     \fBconst char *\fR\fItype\fR, \fBconst char *\fR\fIname\fR);
.fi

.LP
.nf
\fBint\fR \fBpool_resource_destroy\fR(\fBpool_conf_t *\fR\fIconf\fR,
     \fBpool_resource_t *\fR\fIresource\fR);
.fi

.LP
.nf
\fBconst char *\fR\fBpool_resource_info\fR(\fBpool_conf_t *\fR\fIconf\fR
     \fBpool_resource_t *\fR\fIresource\fR, \fBint\fR \fIflags\fR);
.fi

.LP
.nf
\fBpool_component_t **\fR\fBpool_query_resource_components\fR(
     \fBpool_conf_t *\fR\fIconf\fR, \fBpool_resource_t *\fR\fIresource\fR,
     \fBuint_t *\fR\fInelem\fR, \fBpool_value_t **\fR\fIprops\fR);
.fi

.LP
.nf
\fBint\fR \fBpool_resource_transfer\fR(\fBpool_conf_t *\fR\fIconf\fR,
     \fBpool_resource_t *\fR\fIsource\fR, \fBpool_resource_t *\fR\fItarget\fR,
     \fBuint64_t\fR \fIsize\fR);
.fi

.LP
.nf
\fBint\fR \fBpool_resource_xtransfer\fR(\fBpool_conf_t *\fR\fIconf\fR,
     \fBpool_resource_t *\fR\fIsource\fR, \fBpool_resource_t *\fR\fItarget\fR,
     \fBpool_component_t **\fR\fIcomponents\fR);
.fi

.SH DESCRIPTION
.LP
The \fBpool_resource_create()\fR function creates and returns a new resource of
the given \fIname\fR and \fItype\fR in the provided configuration. If there is
already a resource of the given name, the operation will fail.
.sp
.LP
```

```
The \fBpool_resource_destroy()\fR function removes the specified \fIresource\fR
from its configuration file.
.sp
.LP
The \fBpool_resource_info()\fR function returns a string describing the given
\fIresource\fR. The string is allocated with \fBmalloc\fR(3C). The caller is
responsible for freeing the returned string. If the \fIflags\fR argument is
non-zero, the string returned also describes the components (if any) contained
in the resource.
.sp
.LP
The \fBpool_query_resource_components()\fR function returns a null-terminated
array of the components (if any) that comprise the given resource.
.sp
.LP
The \fBpool_resource_transfer()\fR function transfers \fIsize\fR basic units
from the \fIsource\fR resource to the \fItarget\fR. Both resources must be of
the same type for the operation to succeed. Transferring component resources,
such as processors, is always performed as series of
\fBpool_resource_xtransfer()\fR operations, since discrete resources must be
identified for transfer.
.sp
.LP
The \fBpool_resource_xtransfer()\fR function transfers the specific
\fIcomponents\fR from the \fIsource\fR resource to the \fItarget\fR. Both
resources must be of the same type, and of a type that contains components
(such as processor sets). The \fIcomponents\fR argument is a null-terminated
list of \fBpool_component_t\fR.
.sp
.LP
The \fIconf\fR argument for each function refers to the target configuration to
which the operation applies.
.SH RETURN VALUES
.LP
Upon successful completion, \fBpool_resource_create()\fR returns a new
\fBpool_resource_t\fR with default properties initialized. Otherwise,
\fINULL\fR is returned and \fBpool_error\fR(3POOL) returns the pool-specific
error value.
.sp
.LP
Upon successful completion, \fBpool_resource_destroy()\fR returns 0. Otherwise,
-1 is returned and \fBpool_error()\fR returns the pool-specific error value.
.sp
.LP
Upon successful completion, \fBpool_resource_info()\fR returns a string
describing the given resource (and optionally its components). Otherwise,
\fINULL\fR is returned and \fBpool_error()\fR returns the pool-specific error
value.
.sp
.LP
Upon successful completion, \fBpool_query_resource_components()\fR returns a
 Upon successful completion, \fBpool_query_resource_components()\fR returns a
null-terminated array of \fBpool_component_t *\fR that match the provided
null-terminated property list and are contained in the given resource.
Otherwise, \fINULL\fR is returned and \fBpool_error()\fR returns the
pool-specific error value.
.sp
.LP
Upon successful completion, \fBpool_resource_transfer()\fR and
 Upon successful completion, \fBpool_resource_transfer()\fR and
\fBpool_resource_xtransfer()\fR return 0. Otherwise -1 is returned and
\fBpool_error()\fR returns the pool-specific error value.
.SH ERRORS
.LP
The \fBpool_resource_create()\fR function will fail if:
.sp
```

```
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR or \fIname\fR is in
use for this resource type.
.RE

.sp
.ne 2
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The resource element could not be created because the configuration would be
invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_PUTPROP\fR\fR
.ad
.RS 20n
One of the supplied properties could not be set.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
 A system error has occurred. Check the system error code for more details.
.RE

.sp
.LP
The \fBpool_resource_destroy()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 16n
The supplied configuration’s status is not \fBPOF_VALID\fR.
.RE

.sp
.LP
The \fBpool_resource_info()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR or the \fIflags\fR
parameter is neither 0 nor 1.
paramter is neither 0 nor 1.
.RE

.sp
.ne 2
.na
```

```
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The configuration is invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE

.sp
.LP
The \fBpool_query_resource_components()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR.
.RE

.sp
.ne 2
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The configuration is invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE

.sp
.LP
The \fBpool_resource_transfer()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 16n
The supplied configuration’s status is not \fBPOF_VALID\fR, the two resources
are not of the same type, or the transfer would cause either of the resources
to exceed their \fBmin\fR and \fBmax\fR properties.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 16n
A system error has occurred. Check the system error code for more details.
```

```
.RE

.sp
.LP
The \fBpool_resource_xtransfer()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBPOE_BADPARAM\fR\fR
.ad
.RS 20n
The supplied configuration’s status is not \fBPOF_VALID\fR, the two resources
are not of the same type, or the supplied resources do not belong to the
source.
.RE

.sp
.ne 2
.na
\fB\fBPOE_INVALID_CONF\fR\fR
.ad
.RS 20n
The transfer operation failed and the configuration may be invalid.
.RE

.sp
.ne 2
.na
\fB\fBPOE_SYSTEM\fR\fR
.ad
.RS 20n
A system error has occurred. Check the system error code for more details.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRCreate a new resource of type \fBpset\fR named \fBfoo\fR.
.sp
.in +2
.nf
#include <pool.h>
#include <stdio.h>

\&...

pool_conf_t *conf;
pool_resource_t *resource;
\&...

if ((resource = pool_resource_create(conf, "pset",
    "foo")) == NULL) {
    (void) fprintf(stderr, "Cannot create resource\\B{}n");
    ...
}
.fi
.in -2

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
```

```
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled
_
Interface Stability     Unstable
_
MT-Level        Safe
.TE

.SH SEE ALSO
.LP
\fBlibpool\fR(3LIB), \fBpool_error\fR(3POOL), \fBattributes\fR(5)
```