



Daniel P. Berrangé <berrange@redhat.com>

Shell Programming

- `virsh` command
- Listing domains – `'list'`
- Start/stop domains – `'create', 'shutdown'`
- Export config as XML – `'dumpxml'`
- Suspend – `'suspend', 'resume'`
- Hibernate – `'save', 'restore'`
- Status / capability information – `'info'`

Core C objects

- VirConnectPtr – a hypervisor connection.
- VirDomainPtr – instance of a virtual domain, either a Domain-0 or guest
- VirErrorPtr – error information
- VirNodeInfo – host capabilities
- VirDomainInfo – guest domain status

Hypervisor Connections

- Take a URI specifying what hypervisor to connect to
- Default 'NULL' represents a local Xen connection
- Magic 'test:///default' is a 'mock' hypervisor for test harness
- Later expect xen://joe@myhost/
- RO vs RW – auth / no-auth (require root)

Looking up domains

- By name – unique within set of active domains
- By ID – unique until hypervisor reboots
- By UUID – globally unique forever
- List active domain IDs

Handling errors

- Return NULL or -1 upon error
- Error accessible virGetLastError
- Register callback to get error notifications
- List of potential error codes
- Longer descriptive text
- Additional error specific information

Host information

- VirNodeInfo structure
 - Model – string with CPU name
 - Ncpus – number of active CPUs
 - Mhz – maximum CPU freq
 - Nodes – number of NUMA cells
 - Sockets – number of sockets per node
 - Cores – number of CPU cores per socket
 - Threads – number of hyperthreads per core

Domain information

- VirDomainInfoPtr – data structure
 - State – running state
 - Maxmem – maximum memory limit
 - Memory – current memory allocated
 - NrVirtCPU – number of virtual CPUs
 - CpuTime – CPU execution time in nano-sec

Hardware definition

```
<domain type='xen' id='18'>
  <name>fc4</name>
  <os>
    <type>linux</type>
    <kernel>/boot/vmlinuz-2.6.15-1.43_FC5domU</kernel>
    <initrd>/boot/initrd-2.6.15-1.43_FC5domU.img</initrd>
    <root>/dev/sda1</root>
    <cmdline> ro selinux=0 3</cmdline>
  </os>
  <memory>131072</memory>
  <vcpu>1</vcpu>
```

.....

Hardware definition cont...

```
.....  
<devices>  
  <disk type='file'>  
    <source file='/u/fc4.img' />  
    <target dev='sda1' />  
  </disk>  
  <interface type='bridge'>  
    <source bridge='xenbr0' />  
    <mac address='aa:00:00:00:00:11' />  
    <script path='/etc/xen/scripts/vif-bridge' />  
  </interface>  
</devices>  
</domain>
```

C example

```
#include <stdio.h>
#include <libvirt/libvirt.h>

static void
getDomainInfo(int id) {
    virConnectPtr conn = NULL; /* the hypervisor connection */
    virDomainPtr dom = NULL;   /* the domain being checked */
    virDomainInfo info;        /* the informations being fetched */
    int ret;
    /* NULL means connect to local Xen hypervisor */
    conn = virConnectOpenReadOnly(NULL);
    if (conn == NULL) {
        fprintf(stderr, "Failed to connect to hypervisor\n");
        goto error;
    }
    /* Find the domain of the given id */
    dom = virDomainLookupByID(conn, id);
    if (dom == NULL) {
        fprintf(stderr, "Failed to find Domain %d\n", id);
        goto error;
    }
}
```



C example cont...

```
.....
/* Get the informations */
ret = virDomainGetInfo(dom, &info);
if (ret < 0) {
    fprintf(stderr, "Failed to get informations for Domain %d\n", id);
    goto error;
}
printf("Domains %d: %d CPUs\n", id, info.nrVirtCpu);
error:
if (dom != NULL)
    virDomainFree(dom);
if (conn != NULL)
    virConnectClose(conn);
}
int main(int argc, char **argv) {
    getDomainInfo(0);
    return(0);
}
```

Compiling C

- Use pkg-config for compiler flags
- `gcc -o myapp.o `pkg-config -cflags libvirt` myapp.c`
- Use pkg-config for compiler flags
- `gcc -o myapp `pkg-config -libs libvirt` myapp.o`

Python example

```
import libvirt
import sys
conn = libvirt.openReadOnly(None)
if conn == None:
    print 'Failed to open connection to the hypervisor'
    sys.exit(1)
try:
    dom0 = conn.lookupByName("Domain-0")
except:
    print 'Failed to find the main domain'
    sys.exit(1)
print "Domain 0: id %d running %s" % (dom0.ID(),
dom0.OSType())
print dom0.info()
```

Perl example

```
use strict;
use warnings;
use Sys::Virt;

if (@ARGV != 1) {
    print STDERR "syntax: $0 DOMAIN-NAME\n";
    exit 1;
}

my $con = Sys::Virt->new(readonly => 1);

my $name = $ARGV[0];

my $dom = $con->get_domain_by_name($name);

print $dom->get_xml_description(). "\n";
```





<http://libvirt.org/>