

---

```
R """Capsule
```

1

```
Author : Anders Andersen
Created On : Mon Aug 24 00:09:00 1998
Last Modified By: Anders Andersen
Last Modified On: Sat Jul 3 12:57:29 1999
Status : Unknown, Use with caution!
```

Copyright © 1998, 1999 Lancaster University, UK and NORUT Information Technology Ltd., Norway. See COPYING for details.

A capsule provides an environment for programs using the Open-ORB programming model. A capsule represents an address space in our model, but it also provides features that makes it possible for objects in different capsules to cooperate.

A remote capsule is accessed through a `CapsuleProxy`. You usually get hold on a capsule proxy from a name server or an interface reference. The serve loop in a capsule represented by a capsule proxy must be running before you can access it through the proxy. The serve loop is started either by the `serve` or the `servethread` method of the capsule (`servethread` is usually recommended because it starts the serve loop in in a new thread).

You can access the local capsule through the `local` attribute of the capsule module after you have imported it:

```
import capsule                # Import the module (local capsule created)
capsule.local.servethread() # Start the serve loop
```

You can run the capsule module as a program. It will then automatically create a serve loop and export itself to a nameserver with the key "`capsule host`", where `host` is the name host where the capsule is running. The location of the nameserver (and optionally its port) is given as an argument when the capsule is started:

```
python capsule.py nameserver-host:nameserver-port
```

A closer description of the capsule services are found in the file actually implementing this module (see `__capsule__.py`).

```
"""
```

50

```
# The actual capsule and capsule proxy class
```

51

```
import __capsule__
```

52

53

54

55

```
# Name mapping
```

56

```
Capsule = __capsule__.Capsule
```

57

```
CapsuleProxy = __capsule__.CapsuleProxy
```

58

59

```
# Create the local capsule
```

60

```
local = Capsule()
```

61

62

```
# If you are running the capsule module, export it and start the server loop
```

63

```
if __name__ == "__main__":
```

64

```
    import sys
```

65

```
    import nameserver
```

66

```
    if len(sys.argv) > 1:
```

67

68

```
        # The (optional) first argument should be a name server (host:port)
```

69

```
        import re
```

70

```
        nsinfo = re.match(r'([^:]+):(\d+)', sys.argv[1])
```

71

```
        if nsinfo:
```

72

```
            import string
```

73

---

```
    ns = nameserver.NameServerProxy(
        nsinfo.group(1), string.atoi(nsinfo.group(2)))
else:
    ns = nameserver.NameServerProxy(sys.argv[1])

# The (optional) second argument is the export key used
if len(sys.argv) > 2:
    key = sys.argv[2]
else:
    from socket import gethostname
    key = "capsule %s" % (gethostname(),)

# Default values for name server and export key
else:
    from socket import gethostname
    ns = nameserver.NameServerProxy()
    key = "capsule %s" % (gethostname(),)

# Export capsule and start serving
ns.exportCapsule(key, local)
local.serve()
```