

## Getting\_the\_host\_address\_in\_Symbian\_C++

The below code demonstrates retrieving of the host address in Symbian C++.

### Library needed:

```
LIBRARY esock.lib euser.lib insock.lib
```

## Headers

```
#define MAX_IP_ADDR_LEN 16
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <in_sock.h>
```

## Getting Host address

```
/*
 * ret buf :- on return contains the Host Address
 * ret TInt :- Error codes KErrNone if success system wide error codes
 */

TInt GetHostAddress(char *buf)
{
    RSocketServ      iSocketServer;
    RSocket          iSockClient;
    TRequestStatus  iStatus;
    TBuf<(MAX_IP_ADDR_LEN)>      buffer;
    TBuf8<(MAX_IP_ADDR_LEN*2)>   buf_8bit;
    char              ip_addr_l[MAX_IP_ADDR_LEN];
    TRequestStatus    protostartStatus;
    TBool            ip_found_l = EFalse;
    TInt             Err;

    Err = iSocketServer.Connect();
    if (Err != KErrNone)
    {
        return (Err);
    }

    iSocketServer.StartProtocol(KAfiNet, KSockStream, KProtocolInetTcp, protostartStatus);
    User::WaitForRequest(protostartStatus);

    if (protostartStatus.Int() != KErrNone)
    {
        iSocketServer.Close();
        return (protostartStatus.Int());
    }

    Err = iSockClient.Open(iSocketServer, KAfiNet, KSockStream, KProtocolInetTcp);
    if (Err != KErrNone)
    {
        iSocketServer.Close();
        return (Err);
    }
}
```

## Getting\_the\_host\_address\_in\_Symbian\_C++

```
TSoInetInterfaceInfo* inf = new TSoInetInterfaceInfo;

if(inf ==NULL)
{

    iSockClient.Close();
    iSocketServer.Close();
    return KErrNoMemory;
}

TPckg<TSoInetInterfaceInfo> opt(*inf);

/* Set SocketOption For enumerating interfaces */
Err = iSockClient.SetOpt(KSoInetEnumInterfaces, KSolInetIfCtrl);

if (Err != KErrNone)
{
    iSockClient.Close();
    iSocketServer.Close();
    delete inf;
    return Err;
}

/* Start Enumerating the Interfaces */
while (iSockClient.GetOpt(KSoInetNextInterface, KSolInetIfCtrl,opt) == KErrNone)
{

    inf->iAddress.Output(buffer);

    buf_8bit.Copy(buffer);
    memset(ip_addr_l, 0, MAX_IP_ADDR_LEN);
    strcpy(ip_addr_l, (char *)buf_8bit.PtrZ());

    if (inf->iAddress.IsUnspecified())
    {
        // Display Unspecified
    }
    else if (inf->iAddress.IsLoopback())
    {
// display LoopBack
    }
    else if(inf->iAddress.IsLinkLocal())
    {
        // Display LinkLocal address
    }
    else
    {

        strcpy(buf, ip_addr_l);

        ip_found_l = ETRUE;
        break;
    }
} //end of while

/* Free the resources */
iSockClient.Close();
iSocketServer.Close();
delete inf;

if (ip_found_l == ETRUE)
```

## Getting\_the\_host\_address\_in\_Symbian\_C++

```
{
    return KErrNone;
}
else
{
    return KErrNotFound;
}

}
```