Using an already active connection on Symbian

Overview

You can use an already active `RConnection` to establish a connection to the Internet. This saves resources and memory.

`RConnection` has enumeration of the active connections. The source code finds active connections and attaches to the selected connection via the IAP. If there is no selected active connection, a new one is created. Errors are not handled in the source code.

MMP file

```
LIBRARY http.lib
LIBRARY ecom.lib
LIBRARY esock.lib
LIBRARY commdb.lib
CAPABILITY NetworkServices
```

Header file

```
#include <http.h>
#include <es_sock.h>
#include <commdbconnpref.h>
```

Source file

```
// Selected IAP
TInt selectedIap=6;

// Open socket server and connection
RHTTPSession session;     // TODO: Set to your class member variable
RSocketServ socketServ;   // TODO: Set to your class member variable
RConnection connection;   // TODO: Set to your class member variable
TCommDbConnPref connPref; // TODO: Set to your class member variable
socketServ.Connect();
connection.Open(iSocketServ);

// Search through existing connections.
// If there is already a connection that matches the given IAP, try to attach to
// existing connection.
TBool connected(EFalse);
TConnectionInfoBuf connInfo;
TUint count;
if ( connection.EnumerateConnections(count) == KErrNone )
    {
    for (TUint i=1; i<=count; i++)
    {
        // Note: GetConnectionInfo expects 1-based index
    if ( connection.GetConnectionInfo( i, connInfo ) == KErrNone )
        {
        if ( connInfo().iIapId == selectedIap )
        {
        if ( connection.Attach(connInfo, RConnection::EAttachTypeNormal)
        == KErrNone )
            {
            connected = ETrue;
            break;
            }
        }
        }
    }
    }

// Is there an active connection?
if ( !connected )
    {
    // Could not attach to the existing connection.
    // => Start a new connection.
    connPref.SetDialogPreference(ECommDbDialogPrefDoNotPrompt);
    connPref.SetIapId(selectedIap);
    connection.Start(connPref);
    }

// Open session
session.OpenL();

// Set the session's connection info...
RStringPool strPool = session.StringPool();
RHTTPConnectionInfo connInfo = session.ConnectionInfo();

// ...to use the socket server
connInfo.SetPropertyL ( strPool.StringF(HTTP::EHttpSocketServ,
    RHTTPSession::GetTable() ), THTTPHdrVal (socketServ.Handle()) );

// ...to use the connection
connInfo.SetPropertyL ( strPool.StringF(HTTP::EHttpSocketConnection,
    RHTTPSession::GetTable() ),
    THTTPHdrVal (REINTERPRET_CAST(TInt, &(connection))) );
```


(C) Copyright Nokia 2013. All rights reserved.

```
// TODO: Remember to close handles in your class destructor, such as:
CYourClass::~CYourClass()
    {
    session.Close(); // Close handles in this order
    connection.Close();
    socketServ.Close();
    }
```

See also

How can I determine active connections using Symbian C++?

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.