The HTML help client-server pair sample consists of three parts:

1) The Client, derived from Client from the IPC samples and now containing wxRemoteHtmlHelpController

2) The Server, derived from helpview - the app is called helpview

3) The sample help files, specifically called by the Client, which are from the HTML/Help sample

Normally, the Client is started first, and it will in turn start up the Server (helpview) the first time help is requested with Display. If the Server is to be started first, the default service name ("generic\_helpservice" for MSW, "/tmp/generic\_helpservice" for Unix or "4242" otherwise) should be used for the connection in the client.

It is assumed that there will be one copy of the server for each app which uses help. Depending on the OS, it may be possible to use different instances of the same server, or even to use one instance of the server (probably not a good idea) to service more than one app requesting help.

At the moment, if a connection has been established and if the client is killed, the server is also killed. If the client is started up again, it will use the same connection.

If USE\_REMOTE in client.cpp is undefined, the "client" app will use wxHtmlHelpController instead of wxRemoteHtmlHelpController, that is the remote server will not be used. In MSW, this results in freezing of the help window when called from a modal dialog. In GTK, the window is not frozen, but it must be closed before you can return to the app itself.

------------------------------------------------

wxRemoteHtmlHelpController Class

This class can be added to an application (the client) to provide wxWidgets HTML help. It will start up the remote help controller (helpview) the first time the Display member function is called, passing the server the connection name or port number, the name of the help window and the help book (.hhp, .htb, .zip). This class will only work with a particular server app, helpview.

Most of the functions of wxHtmlHelpController are available - the relevant arguments are just passed to wxHtmlHelpController in the server. The functions involving wxConfig are not implemented - the config parameters (font, windows size and position) are kept in the server.

Use the members SetServer and SetService to pass the name of the server app and service name or port. This must be done before the first call to the Display member function, or defaults will be used. The default server app is "helpview" and default service depends on the platform: for MSW it is "appname\_helpservice" (a service name string); for UNIX platforms it is "/tmp/appname\_helpservice" (a file name) and otherwise it is "4242" (a TCP/IP port number), where "appname" is the name of the application without extension.

A wxRemoteHtmlHelpController can be created in wxApp::OnInit, or some high level window like a mainframe. It should be deleted explicitly, for example in wxApp::OnExit, or else the various connection objects are deleted in the wrong order. Using a wxRemoteHtmlHelpController member of wxApp does not work because of problems on exit.

This has been tested on wxMSW and wxGTK. It does not work on wxMac because the connection classes, as well as wxExecute, are not fully supported.

----------------------------------------------

Changes to helpview sample

The constructor now tests the command-line arguments, looking first for help books (containing .hhp, .htb, .zip). Then it looks for "--server". If arguments not falling in these categories are present, they are taken to be 1) the service name or 2) the window name (see wxRemoteHtmlHelpController for the meaning of service name).

The client passes most functions of wxHtmlHelpController, with argument, to helpview using wxConnection::Poke. The string for the Display function is passed with wxConnection::Execute.