

原文は[こちら](#)をご覧ください。

## wxClient

[wxClient](#) オブジェクトは、クライアント - サーバ間の DDE ( Dynamic Data Exchange ) ライクな通信のクライアント部分を表現する。wxDDEClient を使用した実際の DDE ベースの実装は、Windows でのみ利用可能である。しかし、同じ API を持つ wxTCPClient を使用すると、プラットフォームに依存しない、ソケットベースの API の使用が可能である。

A [wxClient](#) object represents the client part of a client-server DDE-like (Dynamic Data Exchange) conversation. The actual DDE-based implementation using wxDDEClient is available on Windows only, but a platform-independent, socket-based version of this API is available using wxTCPClient, which has the same API.

適合するサーバとの通信を行うクライアントを作成するためには、[wxConnection](#) を継承したクラスと、[wxClient](#) を継承したクラスを準備する必要がある。カスタマイズした [wxConnection](#) クラスはサーバとの通信を受信する。カスタマイズした [wxClient](#) では、コネクションを確立するときに、ユーザによりオーバーライドされた [wxClient::OnMakeConnection](#) メンバが要求されたクラスの [wxConnection](#) を返すことができるよう要求される。IPC サンプルと[プロセス間通信の概要](#)を参照のこと。

To create a client which can communicate with a suitable server, you need to derive a class from [wxConnection](#) and another from [wxClient](#). The custom [wxConnection](#) class will intercept communications in a 'conversation' with a server, and the custom [wxClient](#) is required so that a user-overridden [wxClient::OnMakeConnection](#) member can return a [wxConnection](#) of the required class, when a connection is made. Look at the IPC sample and the [Interprocess communications overview](#) for an example of how to do this.

plugin::pdf::PDFParser=HASH(0x53e16c) 基本クラス

[wxClientBase](#)  
wxObject

plugin::pdf::PDFParser=HASH(0x53e16c) インクルードファイル

<wx/ipc.h>

plugin::pdf::PDFParser=HASH(0x53e16c) 参考

[wxServer](#), [wxConnection](#), [プロセス間通信の概要](#)

plugin::pdf::PDFParser=HASH(0x53e16c) メンバ

[wxClient::wxClient](#)  
[wxClient::MakeConnection](#)  
[wxClient::OnMakeConnection](#)  
[wxClient::ValidHost](#)

## wxClient::wxClient

### wxClient()

クライアントオブジェクトを構築する。

Constructs a client object.

## wxClient::MakeConnection

wxConnectionBase \* MakeConnection(const wxString& host, const wxString& service, const wxString& topic)

host( UNIX ではマシン名であり、同一マシンでは 'localhost' を使用する。Windows の DDE では無視される。) サービス名 service、topic 文字列を使用してコネクションの確立を試みる。サーバがコネクションを許可すると、wxConnection オブジェクトが返る。継承されたコネクションオブジェクトを返すために、wxClient::OnMakeConnection メンバをオーバーライドすることにより、返される wxConnection の型を変更することができる。

Tries to make a connection with a server by host (machine name under UNIX - use 'localhost' for same machine; ignored when using native DDE in Windows), service name and topic string. If the server allows a connection, a wxConnection object will be returned. The type of wxConnection returned can be altered by overriding the wxClient::OnMakeConnection member to return your own derived connection object.

Unix では、サーバ名は、インターネットドメインソケットが通信に使用される場合にはポート番号を、Unix ドメインソケットを使用する場合には適当なファイル名(存在しないファイル。後で削除される。)のどちらかを指定する。呼び出しに失敗すると(例えば、ポート番号が既に使用されている) false が返る。

Under Unix, the service name may be either an integer port identifier in which case an Internet domain socket will be used for the communications, or a valid file name (which shouldn't exist and will be deleted afterwards) in which case a Unix domain socket is created.

セキュリティに関して：インターネットドメインソケットに対してアクセスを完全に制限できないため、IPC に対して心配がある場合には、可能ならば Unix ドメインソケットを使用したほうが良い。

SECURITY NOTE: Using Internet domain sockets is extremely insecure for IPC as there is absolutely no access control for them, use Unix domain sockets whenever possible!

## wxClient::OnMakeConnection

wxConnectionBase \* OnMakeConnection()

wxClient::MakeConnection から呼ばれ、デフォルトでは、単に新しい wxConnection オブジェクトを返すだけである。アプリケーション向けにカスタマイズのために継承した wxConnection を返す場合には、このメソッドをオーバーライドする。

Called by `wxClient::MakeConnection`, by default this simply returns a new `wxConnection` object. Override this method to return a `wxConnection` descendant customised for the application.

自身でコネクションクラスを継承することで、サーバ主導のメッセージ受信が可能になる。例えば、`wxConnection::OnAdvise` である。また、新しいクラスのインスタンスにアプリケーション固有のデータを持たせることも可能である。

The advantage of deriving your own connection class is that it will enable you to intercept messages initiated by the server, such as `wxConnection::OnAdvise`. You may also want to store application-specific data in instances of the new class.

## `wxClient::ValidHost`

`bool ValidHost(const wxString& host)`

`host` が正当なホスト名であれば `true` を返し、そうでなければ `false` を返す。MS Windows では、常に `true` を返す。

Returns true if this is a valid host name, false otherwise. This always returns true under MS Windows.