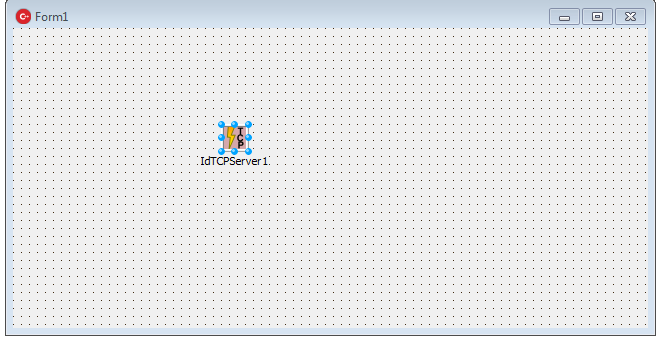
Communication TCP c++

On utilise le composant Indy ServerTCP/IP.



Il permet à l’aide de ses événements OnConnect , OnExecute , OnDisconnect mais aussi à l’aide de ses méthodes et propriétés de mettre en place un serveur TCP/IP.

On peut régler l’adresse Ip et le Port du Serveur TCP à l’aide de :

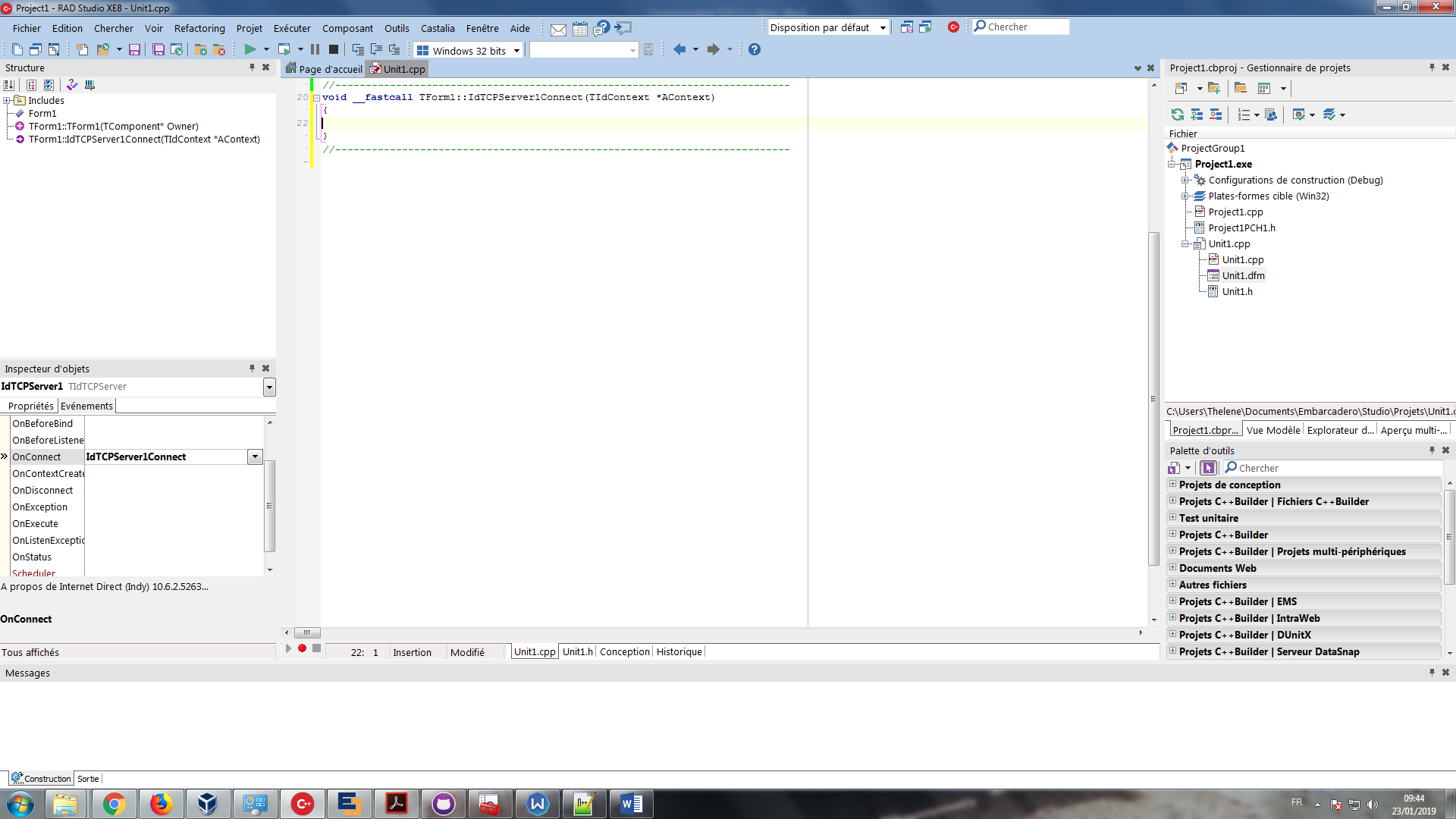
IdTCPServer1->Bindings->Items[0]->IP="192.168.65.26";

IdTCPServer1->Bindings->Items[0]->Port="25565";

Puis pour lancer le serveur il suffit d’utiliser :

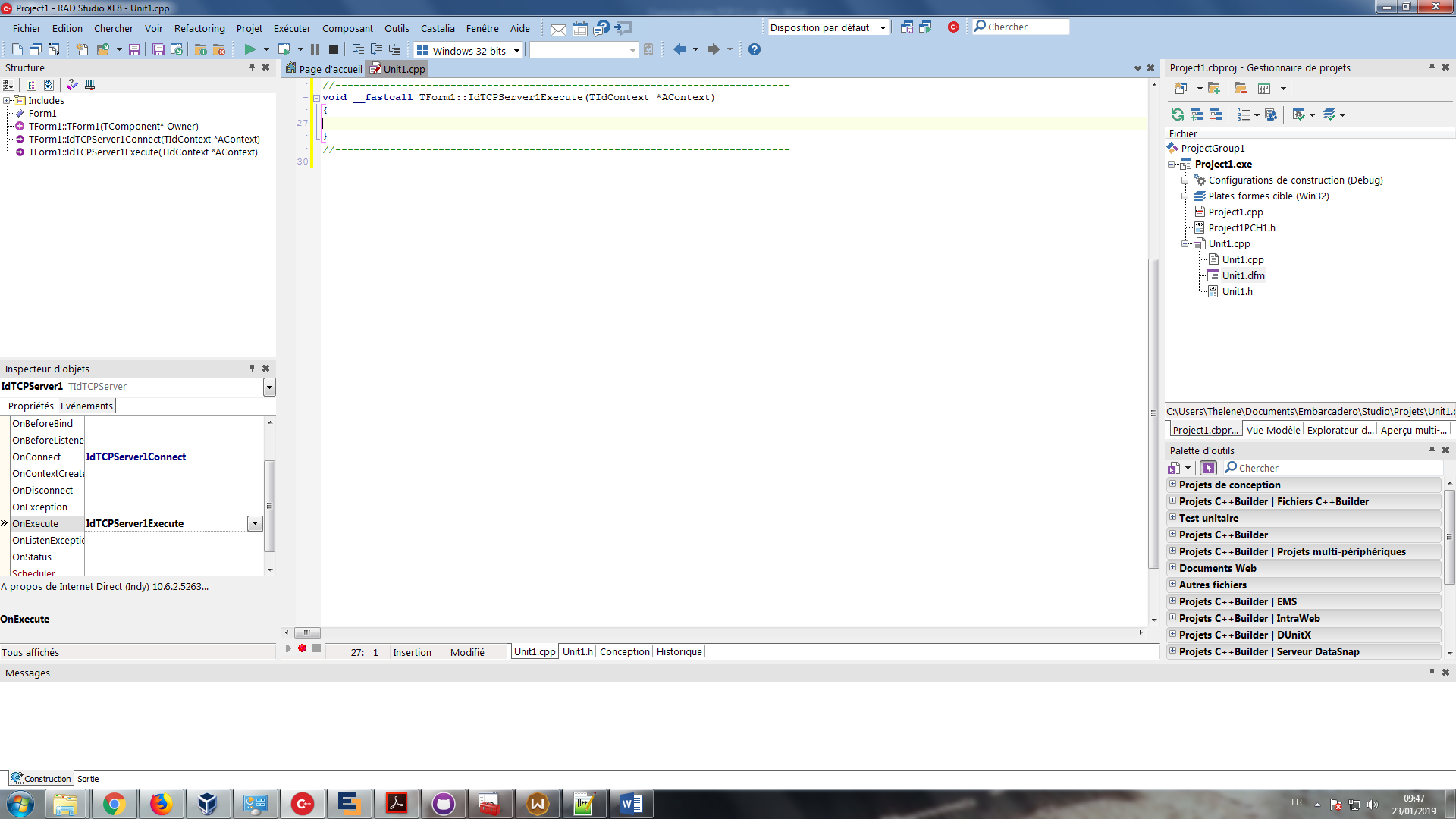
IdTCPServer1->Active=true;

Pour éxécuter des actions lors de la conenxion d’un client , on utilise l’événement OnConnect :



Il suffit de mettre ce que l’on souhaite ici pour lui indiquer ce qu’il faut faire lors de la connexion d’un client.

Puis lorsque le Serveur reçoit quelque chose depuis le client, on utilise l’événement OnExecute :



Dans cet événment on peut indiquer au serveur ce que l’on souhaite renvoyer au client.

Si l’on souhaite récupérer ce qu’a envoyé le client on utilise cette méthode :

RequeteClient=AContext->Connection->Socket->ReadLn().w\_str();

Ce qui nous permet de comparer et de renvoyer une réponse .

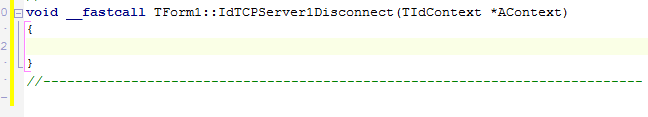
if(RequeteClient == L"temp")

{

AContext->Connection->Socket->Write(“MonRetour”);

}

Puis lorsque le client se déconnecte, on utilise l’événement OnDisconnect:



Ce qui nous permet de réaliser certaines actions lorsque le client se déconnecte.