I have added a new engine call to 4.0 beta 5 as follows:
cScTextObj *Send_Client_Text(const WideStringClass& message, TextMessageEnum type, bool
popup, int senderId, int receiverId, bool dodirtybit, bool doact)

dodirtybit determines whether cScTextObj::Init should make its normal calls to the various
Set_Object_Dirty_Bit functions or not
doact determines whether cScTextObj::Init should make its normal calls to cScTextObj::Act

The function will create a new cScTextObj, call its constructor, call cScTextObj::Init and return the
new object.
Note that you should not call Set_Delete_Pending, nor should you delete the returned object as
Set_Delete_Pending is already handled by cScTextObj::Init and the object will be automatically
deleted by the netcode.

You are free to call any of the Set_Object_Dirty_Bit functions and to call Send_Object_Update.
Note however that you can only call Send_Object_Update ONCE for each cScTextObj object as
cScTextObj was not designed to be sent multiple times. You can also (at any point) call
cScTextObj::Act (a virtual function) to cause the message to be processed by the server. (i.e. on
the FDS it would do its normal thing and display the output on the console)

The purpose of the engine call is to make this stuff easier to use and to make sure it wont break in
the future if cScTextObj has to change for some reason.