

SCA Developer's Guide

APPENDIX B

IDL for XYZ Waveform Physical layer

Generated by Rational Rose for use by OIS OrbExpress

```
//Source file: H:/jtrs/idl/xyz_physical.idl

#ifndef __XYZ_PHYSICAL_DEFINED
#define __XYZ_PHYSICAL_DEFINED

/* CmIdentification
%X% %Q% %Z% %W% */

#include "CF_Components/cf.idl"

module PhysicalNonRealTimeAPI {

    interface Transmit_Inhibit {

        /* @roseuid 3C02B9D000EB */

        boolean inhibitTransmit (
            in boolean Inhibit
        );
    };
};

module XYZ {

    module XYZ_Physical_NRT {

        enum ModeType {
            VOICE_MODE,
            DATA_MODE
        };

        interface SetMode {

            /* @roseuid 3C14FD420382 */

            void setMode (
                in ModeType newMode
            );
        };
    };

    interface Control_Physical_XYZ :
        XYZ_Physical_NRT::SetMode,
        PhysicalNonRealTimeAPI::Transmit_Inhibit {
    };

    interface XYZ_Physical_Control_Interface :
        Control_Physical_XYZ,
        CF::Resource, CF::Port {
    };
}
```

```
module XYZ_Physical_RT {  
  
    struct NullControlPhys {  
  
        /* dummy is placed here only because  
        OrbExpress seems unwilling to generate code  
        if structure is completely empty. */  
  
        char dummy;  
    };  
  
    interface TransmitPacketPhys {  
  
        /* This operation is used to push Client data to the Server  
        with a Control element and a Payload element.  
        @roseuid 3C03926D03E5 */  
  
        oneway void pushPacket (  
            in NullControlPhys control,  
            in CF::OctetSequence payload  
        );  
  
    };  
  
    interface UserProvider_Physical_XYZ :  
        TransmitPacketPhys {  
  
        /* This operation is used to notify client  
        whether or not data is available to be received.  
        @roseuid 3C039422002C */  
  
        oneway void signalDetected (  
            in boolean signalPresent  
        );  
  
    };  
  
};  
  
interface XYZ_Physical_Interface :  
    CF::Resource,  
    CF::Port,  
    Control_Physical_XYZ,  
    XYZ_Physical_RT::UserProvider_Physical_XYZ {  
};  
  
interface XYZ_Physical_Packets_Interface :  
    CF::Port,  
    XYZ_Physical_RT::UserProvider_Physical_XYZ {  
};  
  
};  
  
#endif
```