

Windows Desktop Sharing API

Stipe Kodzoman

Writing a Desktop Sharing Application

Windows Desktop Sharing (WDS) API provides a rich set of API to share your complete desktop or just individual applications.

Windows Desktop Sharing API allows a Windows Desktop Session to be shared across multiple viewers.

There are two participants in Windows Desktop Sharing:
- the sharer and the viewer.

The sharer is the system whose screen is being shared out

The viewer is the system that can view the sharer's screen

Writing a Desktop Sharing Application

Object Model of the API:

API currently is published as an in-proc COM DLL (RdpEncom.dll)

There are 2 primary objects that can be created:

1. IRDPSRAPISharingSession - COM object that enables sharing of desktop
2. IRDPSRAPIViewer - ActiveX control that can be embedded into a host window for viewing the sharing session.

Starting the sharing Session:

```
private void OnAttendeeConnected(object pObjAttendee)
{
    IRDPSRAPIAttendee pAttendee = pObjAttendee as IRDPSRAPIAttendee;
    pAttendee.ControlLevel = CTRL_LEVEL.CTRL_LEVEL_VIEW;
    LogTextBox.Text += ("Attendee Connected: " + pObjAttendee.RemoteName +
                       Environment.NewLine);
}
```

```
// Create a new RdpSession instance
```

```
m_pRdpSession = new RDPSession();
```

```
// Subscribe to events
```

```
m_pRdpSession.OnAttendeeConnected +=  
new _IRDPSessionEvents_OnAttendeeConnectedEventHandler(OnAttendeeConnected);
```

```
m_pRdpSession.OnAttendeeDisconnected +=  
new _IRDPSessionEvents_OnAttendeeDisconnectedEventHandler(OnAttendeeDisconnected);
```

```
m_pRdpSession.OnControlLevelChangeRequest +=  
new _IRDPSessionEvents_OnControlLevelChangeRequestEventHandler  
    (OnControlLevelChangeRequest);
```

```
// Start the Sharing Session
```

```
m_pRdpSession.Open();
```

```
LogTextBox.Text += "Presentation Started. Your Desktop is being shared." + Environment.NewLine
```

Creating an Invitation:

```
// Create invitation.
```

```
IRDPSRAPIInvitation pInvitation =  
    m_pRdpSession.Invitations.CreateInvitation("WinPresenter","PresentationGroup","",5);
```

```
string invitationString = pInvitation.ConnectionString;
```

```
// Save Connection String to File
```

```
WriteToFile(invitationString);
```

Stopping the Sharing Session

```
m_pRdpSession.Close();
```

```
LogTextBox.Text += "Presentation Stopped." + Environment.NewLine;
```

```
Marshal.ReleaseComObject(m_pRdpSession);
```

```
m_pRdpSession = null;
```

Viewer Application:

- You will need to create a windows application and then add RdpViewer Class as an ActiveX control.
- Visual Studio will create an instance of AxRdpViewer and you can resize the ActiveX control to fit to your needs on the Windows Form
- You can also choose to create this ActiveX control dynamically and place it in the host Form.

Connecting to the Sharing Session:

Code for OnError event handler method:

```
private void OnError(object sender, _IRDPSessionEvents_OnErrorEvent e)
{
    int ErrorCode = (int)e.errorInfo;
    LogTextBox.Text += ("Error 0x" + ErrorCode.ToString("X") + Environment.NewLine);
}
```

Connecting to the Sharing Session:

Code to connect to the Sharing Session:

```
string ConnectionString = ReadFromFile();  
  
if (ConnectionString != null)  
{  
    pRdpViewer.Connect(ConnectionString, "Viewer1", "");  
}
```

Disconnecting from the Sharing Session:

```
pRdpViewer.Disconnect();
```