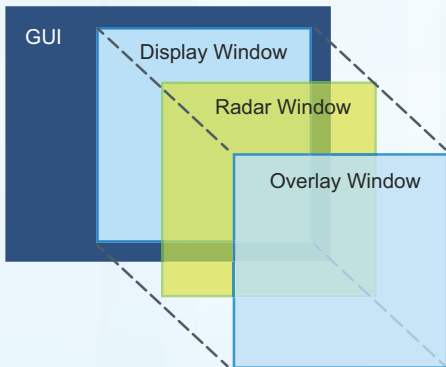


SPx

Radar Display Coprocess (RDC)



Features:

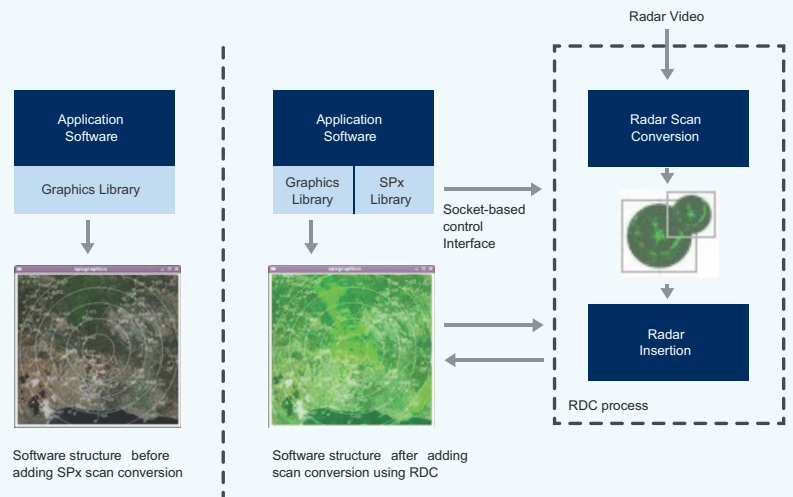
- Radar scan conversion coprocess
- Simplifies radar addition into application
 - Minimal new code in application
 - Use configuration files in RDC
 - Clean separation of scan conversion and application
- Multiple radar input types
 - HPx family
 - Network radar input
 - Test channel
 - Recorded video (SPx format)
- Windows and Linux C++ interface
- .NET support under Windows
- Multiple radar windows
- Automatic resize of radar windows
- Real-time radar updates
- PPI and B-Scan support
- Configurable radar colour and fade
- Configurable scales, centering
- Underlay and Overlay graphics
- Multiple composition modes
- Optional plug-ins for processing

SPx Radar Display Coprocess (RDC) is a software application that handles radar receipt, processing and scan conversion on behalf of a client application, and then combines the scan converted radar picture with the graphics of the application.

Scan conversion can be added into a Windows or Linux/X11 application using the SPx C++ client library. In this situation, the scan conversion runs as a thread as part of the client application, and all the SPx code for radar input and display is part of the application. Although this mode provides for the greatest flexibility in terms of using SPx features, a much simpler architecture is provided by the RDC.

RDC runs as a separate Windows or Linux process. The client application connects to RDC over a simple socket interface and makes calls to configure the display. RDC handles the radar processing and combines its scan converted radar video with the graphics from the original application. The image combination occurs with minimal cooperation from the application.

In the diagram below, the situation on the left shows the original application before addition of radar. The application software uses the native graphics library (X11 or Windows) to create the graphics of the application. The situation on the right shows the software after the addition of RDC. The application uses a lightweight SPx library to send commands to RDC, which is running as a separate process. The RDC handles radar receipt, scan conversion and insertion of the image into the graphics application. If the application wants to change the radar view or range scale it simply sends a command to the RDC to make a corresponding change to the radar image.



The RDC supports all of the principal features of the underlying SPx class library for radar display, including full control of the radar presentation (scale, centering, colour, fade, brightness etc), support for multiple radar windows, support for multiple source types (HPx hardware, network, test patterns) and support for trail history retention on view change.

RDC allows the application to configure graphical underlays and overlays from either windows or bitmaps. The radar video may then be blended with the underlay graphics and overlaid by a window or bitmap designated as overlay. In the simplest scenario, an existing application creating Windows or X11 graphics can be easily enhanced to add radar video with just a few new function calls. ■

Using RDC in a Client Application

One of the main benefits of the RDC is that the radar receipt and scan conversion is handled outside of the application in a separate process. This minimises the amount of additional code that needs to be added to the application and also ensures that activities of the application and scan conversion have minimal interaction. In the simplest use of RDC, the application simply connects to RDC and defines the identity of the window where radar scan conversion should appear. The initial state of the RDC, as read from a configuration file, can then be used to configure the details of the radar source, colour, fade, view and update rate. The RDC is responsible for automatically sizing, and where necessary resizing, the radar window to fit the designated graphics window.

The following code fragment shows how radar can be added into a graphics application using RDC (error checking is removed for brevity). The identifier myWindowID is the window identifier (type HWND for Windows, type Window for X11).

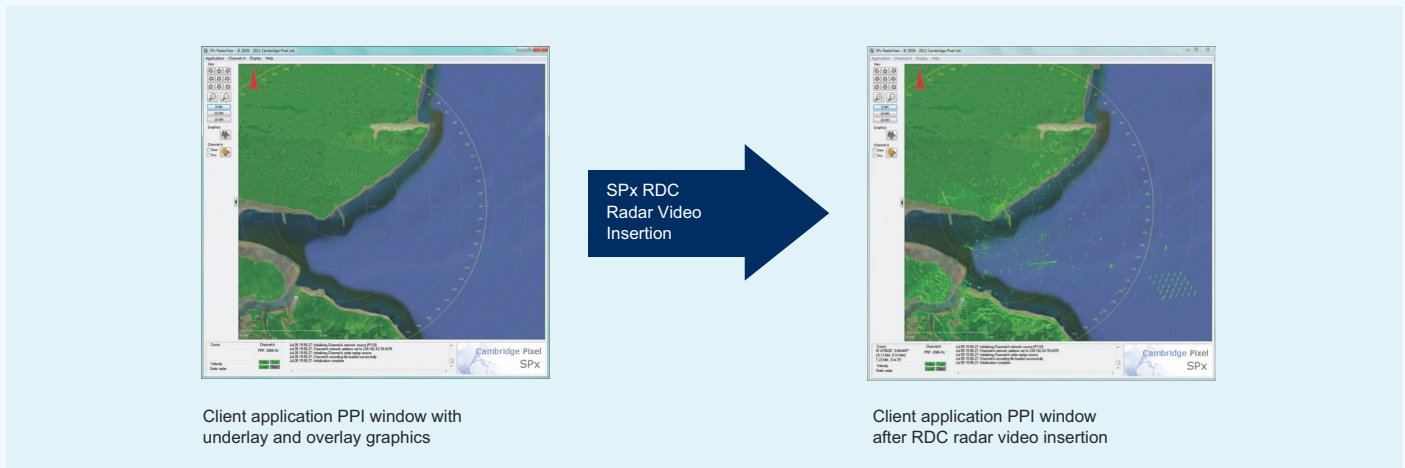
```

/* SPx Initialisation */
SPxInit();

/* Create object used to interface with the RDC. */
m_rdc = new SPxRemoteServer();

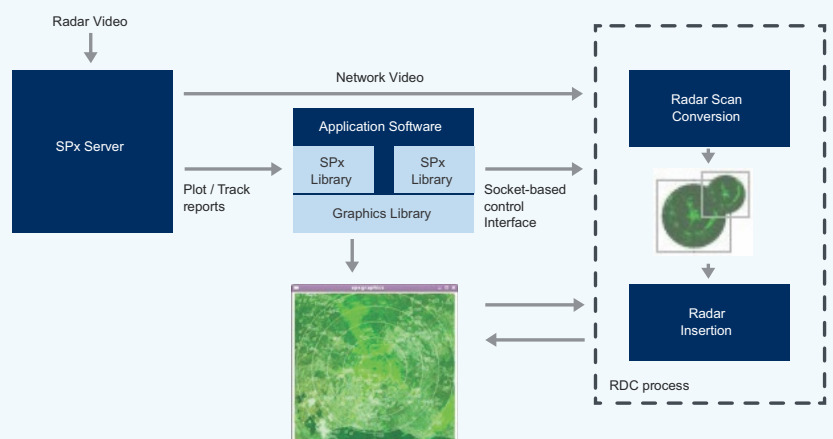
/* Connect to RDC - assume RDC on same machine here */
m_rdc->Connect("127.0.0.1");

/* Associate the PPI window with client window 0 on the RDC */
SPxSprintf(arg, sizeof(arg), "%u", myWindowID);
m_rdc->SetRemoteParameter("Win0", "Associate", arg);
    
```



Using RDC with SPx Server

RDC may be used to receive radar video from an SPx Server. The network video is received, decompressed and scan converted into one or more radar windows that may then be combined with the application graphics. If SPx Server is also providing plot extraction or target tracking capabilities then the application code may be receiving that data for display as overlay graphics – see diagram opposite.



Ordering and Licensing Information

RDC is provided as a standard software component in the SPx Development Licence and requires a standard scan conversion runtime licence for deployed systems.

For more information, please contact: