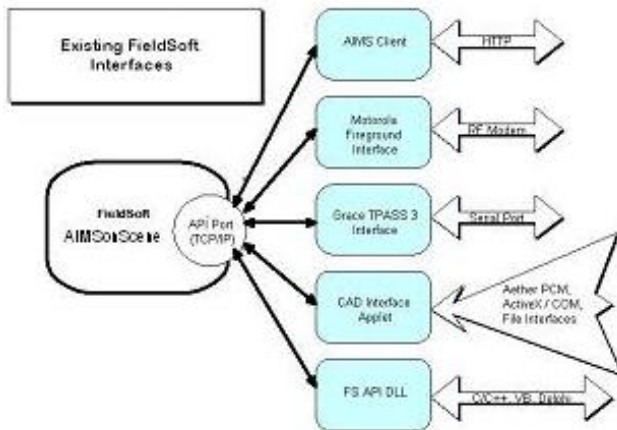


# Enhance Real Time Situational Awareness

**FieldSoft**



**Application Programming Interface (API)**

Create Custom User Interfaces To AIMSonScene

**Seamlessly Integrate Your Application(s) With AIMSonScene**

## API Features

Custom code a means to exchange data between nearly any other application and AIMSonScene.

Connect to local (same computer) or remote (separate computer) software applications or hardware systems.

Use custom code and IP sockets or the FieldSoft API dynamic link library (DLL) to a.) send commands and queries to AIMSonScene, b.) listen for events originating within FieldSoft applications, c.) or to query the applications and return incident and responder accountability data.

Create uni-directional or bi-directional connections and interfaces.

Make up to six separate connections to AIMSonScene via the API.

Take complete control of the AIMSonScene graphic user interface (GUI), code your own interface to AIMSonScene features, or create a combination thereof.

## API Benefits

Share information with new or legacy systems.

No need to develop command, accountability, or tactical information browser applications from scratch.

Enhance a field-tested and field proven product as needed to reflect unique end user requirements.

Integrate different products so the end result appears seamless to the end user.

Leverage the development costs of your hardware based systems by interfacing them to an existing software application used by command staff.

## API Use

API use is straightforward. Under an IP sockets approach for example:

1. Start AIMSonScene
2. Open a TCP/IP socket on the Port used by the API
3. Send a registration message
4. Send your API commands or queries
5. Wait for a response ("OK", "ERROR", "QRESULT", "EVENT")
6. Send de-registration message when complete
7. Close the socket

The following Visual Basic code example illustrates how to use a sockets approach to add a resource (e.g. – fire truck or patrol vehicle) to AIMSonScene, so that the unit is available within AIMSonScene for assignment to a sector, division, or group.

```
Dim g_bSendComplete as Boolean
Dim g_bRcvComplete as Boolean
Dim g_Response as String

Private sub Sample()
    TCP.LocalPort = 0 'Setup Socket & Connect
    TCP.RemotePort = 1906
    TCP.RemoteHost = "127.0.0.1"
    TCP.Connect

    SendCmd ("X1|Y|N|") 'Register
    SendCmd ("IS|STRUCT^COMMAND^G^123 N MAIN ST^HOUSE FIRE^") 'Incident Start
    SendCmd ("SI|Arriving^") 'Sector Insert
    SendCmd ("SA|Arriving^") 'Sector Assign
    SendCmd ("UI|Unit99^") 'Unit Insert
    SendCmd ("UA|Unit99^Arriving^") 'Unit Assign

    SendCmd (Header & "X2||") 'De-register

    TCP.Close 'Disconnect Socket
End sub

Private Function SendCmd(Cmd As String) As Boolean
    Dim Request As String
    Header = "API1.1|SYSTEM|CLIENT|"

    Request = Header & Cmd & "|" & Chr$(ETX)

    If TCP.State = sckConnected Then 'Check for connection
        TCP.SendData Request
        g_bSendComplete = False
        g_bRcvComplete = False

        Do While g_bSendComplete <> True 'Wait for command to complete
            DoEvents
        Loop

        Do While g_bRcvComplete <> True 'Wait for response to cmd
            DoEvents
        Loop
        Send = True
    Else
        Send = False
    End If
End Function

Private Sub TCP_DataArrival(ByVal bytesTotal As Long)
    Dim newData As String
    TCP.GetData Response
```

```

    g_bRcvComplete = True
End Sub

Private Sub TCP_SendComplete()
    g_bSendComplete = True
End Sub

```

The DLL approach to API use is also straightforward. For example,

1. Start AIMSonScene
2. Create FSAPI object
3. Issue Open()
4. Issue Registration()
5. Issue IncidentStart()
6. Issue desired command checking for TRUE
7. Issue DeRegistration()
8. Issue Close

The following C/C++ code example shows how to add a resource to AIMSonScene when the DLL approach is used. The end result of the "Unit Insert" is the same as that described in the Visual Basic sockets example above.

```

void sample()
{
    CFSApi* pFieldSoft = new CFSApi;
    pFieldSoft->Open("API1.1", "CAD System X", "CAD");
    Registration( FALSE , FALSE );
    IncidentStart("STRUCT", "COMMAND", "", "G", "123 N MAIN ST", "HOUSE FIRE");
    pFieldSoft->SectorInsert("Arriving");
    pFieldSoft->SectorAssign("Arriving");
    pFieldSoft->UnitInsert("UNIT99");
    pFieldSoft->UnitAssign("UNIT99", "Arriving");
    pFieldSoft->DeRegistration();
    pFieldSoft->Close();
    delete pFieldSoft;
}

```

## API Use To Date

Note that the above examples are very limited in scope. The FieldSoft API has successfully been used by FieldSoft and by other developers to create a number of different interfaces and to connect or integrate other systems with AIMSonScene.

FieldSoft used the API, for example, to create the FieldSoft Interface Applet To CAD (computer aided dispatching), which is a commercial off-the-shelf (COTS) software interface between AIMSonScene and most CAD systems. FieldSoft also used the API to integrate AIMSonScene with both the Grace Industries Inc. T PASS firefighter accountability system and Motorola's Fireground Communications system for firefighter accountability.

A CAD system vendor used the API to create their own CAD interface to AIMSonScene. A system integrator actually went so far as to create a totally unique GUI "front end" to AIMSonScene through use of [the FieldSoft API and Java](#), and it took the integrator but [3 days](#) to code and test the custom GUI.

## API SDK Contents

The API SDK includes a 180 page plus technical manual, code examples in both Visual Basic and C++, a developer's interface to the API that makes it possible for a programmer to actually code and test one or more API commands and queries on an interactive basis, the DLL described earlier, API unit test code, and FieldSoft's Interface Applet (FSIA) to CAD (computer aided dispatching).

Other files that come with the API SDK include FSAPI.H (a C/C++ header file), FSAPI.LIB (a C/C++ import library), and FSAPI.BAS (a VB module).

## API SDK License

An API SDK development license can be purchased directly from FieldSoft only. The purchase includes the API SDK, two AIMSonSceneSU development only licenses, and one FSIA development license. Also included is a one to two day API programmer training session conducted in FieldSoft's metropolitan Phoenix office. Alternatively, developers, integrators, and customers who have an established business relationship may be able to obtain the API SDK for no additional charge upon request, provided that the party first attends the training session described above. AIMSonScene and FSIA development licenses are ***not*** however, included with any API SDK license provided for no charge.

Any product created with the API SDK is the property of the developer or integrator and may be distributed royalty free. AIMSonScene and FSIA development licenses provided with the API SDK may not be distributed under any circumstances.



480-899-2128    [www.fieldsoft.com](http://www.fieldsoft.com)  
P.O. Box 1378 Chandler, Arizona  
85244-1378 U.S.A.