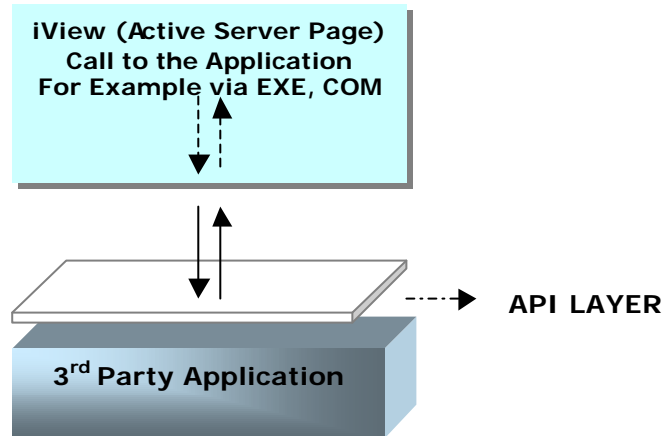


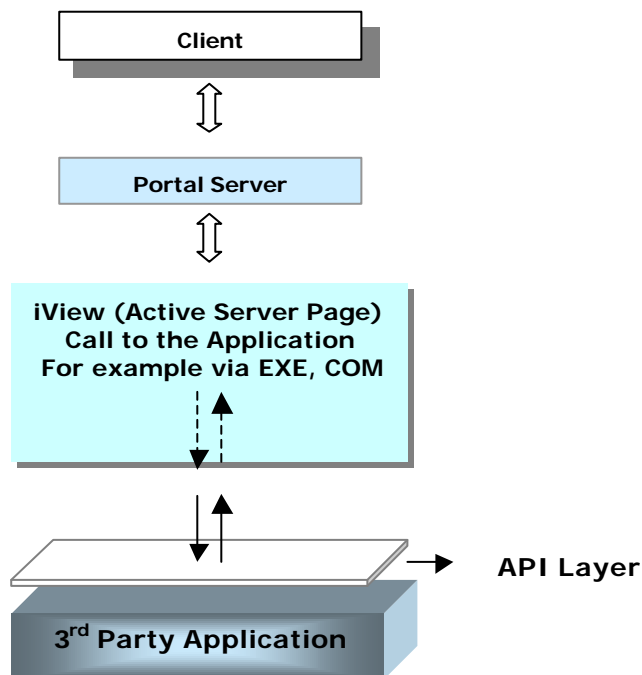
3RD Party Applications- Introduction

iViews can contain data from 3rd Party Applications. The iView can be integrated with the 3rd party application through the 3rd Party Application APIs.

The iView Active Server Page can connect to the 3rd Party Application by utilizing the API Layer exposed by the application. For example, by a call to an Executable, or a COM object in the ASP.



Architecture



The Flow

- The client requests an iView.
- The Portal Server connects to the iView Application (ASP file) and sends the parameters as defined by the iView.
- The Active Server Page connects to the 3rd Party Application and retrieves data using the by utilizing the Application API Layer.
- The Active Server Page parses the returned data, and converts it into SAP Portals XML.
- The Portal Server renders the SAP Portals XML to HTML and serves the page to the client

To achieve Single-Sign On in the Portal for these iViews, please read the document [Single Sign-on](#).

Sample Code

This is just one example. The actual method and technology used to connect depends on the 3rd Party Application.

Consider *yourapp* as your 3rd party application, which provides an Open Query API. Consider an executable *yourapp.exe*, which acts as an intermediate between ASP and *yourapp* and retrieves data using ODBC call to *yourapp*'s Open Query API.

The following piece of code in the ASP prepares the URL which is sent to invoke the executable:

```
set xmlHttp = Server.CreateObject("Msxml2.ServerXMLHTTP.3.0")
'Check if we have a connection
If (xmlHttp) then
    'The Query - this query was created before in yourapp database.
    queryStr = "{call query." + queryName + "}"

    'Executing the query
    urlStr = "http://"
    +Request.ServerVariables("SERVER_NAME")+ ":" +Request.ServerVariable
s("SERVER_PORT")+ "/scripts/yourapp.exe?" urlStr = urlStr +
Server.URLEncode( """" + queryStr + """" +userID+ " " +userPwd)
xmlHttp.open "GET", urlStr,false
xmlHttp.send
```

The data is retrieved by the executable, using ODBC call to the *yourapp* Open Query API, and returned to the ASP as an XML. Following the code which takes this data:

```
' Getting the result of the query.
xmlTextAnswer = xmlHttp.responseText
```

SAP Portals Inc. assumes no responsibility for errors or omissions in these materials. These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP Portals Inc. shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP Portals Inc. does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials and or on these sides.

SAP Portals Inc. has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.