Calling R/3 Via OLE/JavaScript

Using the OLE/Active-X functionality of R/3 you can call R/3 from any object aware language. Actually it must be able to do DLL calls to the RFC libraries of R/3. SAP R/3 scatters the documentation for these facilities in several subdirectories of the SAPGUI installation. For details you have to look for the SAPGUI Automation Server and the SDK (RFC software development kit).

Summary

- R/3 can exchange its IDoc by calling a program that resides on the server
- The programs can be written in any language that supports OLE-2/Active-X technology
- Programming skills are mainly required on the PC side, e.g., you need to know Delphi, JavaScript or Visual Basic well
### 13.1 R/3 RFC from MS Office Via Visual Basic

The Microsoft Office suite incorporates Visual Basic for Applications (VBA), a fully object-oriented language. JavaScript and JAVA are naturally object-oriented. Therefore, you can easily connect from JavaScript, JAVA, WORD, EXCEL, and all other VBA compliant software to R/3 via the CORBA compatible object library (in WINDOWS known also DLLs or ACTIVE-X (=OLE/2) components).

**Visual Basic** is a DCOM compliant programming language. Visual Basic is finally designed as an object oriented language compliant to DCOM standard.

**JavaScript or JAVA** are naturally object languages. JavaScript is a typical object oriented language which is compliant to basic CORBA, DCOM and other popular object standards.

**DLLs installed with SAPGUI**. The libraries are installed to the workstation with the SAPGUI installation. They are technically public linkable objects, in WINDOWS these are DLLs or ACTIVE-X controls (which are DLLs themselves).

**Object library SAP provides a method CALL which will call a function module with all interface parameters**. The object library SAP contains among others the object type FUNCTIONS whose basic method CALL performs an RFC call to a specified R/3 function module. With the call you can pass object properties which will be interpreted as the interface parameters of the called function module.

If the RFC call appears not to be working, you should first try out to call one of the standard R/3 RFC function like RFC_CALL_TRANSACTION_USING (calls a specified transaction or RFC_GET_TABLE (returns the content of a specified R/3 database table).

SAP R/3 provides a set of object libraries, which can be registered with Visual Basic. The library adds object types to VBA which allow RFC calls to R/3.

**Try to call standard routines for testing**.

### 13.2 Call Transaction From Visual Basic for WORD 97

This is a little WORD 97 macro, that demonstrates how R/3 can be called with a mouse click directly from within WORD 97.

The shown macro calls the function module RFC_CALL_TRANSACTION_USING. This function executes a dynamic call transaction using the transaction code specified as the parameter.

You can call the macro from within Word, by attaching it to a pseudo-hyperlink. This is done by adding a MACROBUTTON field to the WORD text. The MACROBUTTON statement must call the VBA macro R3CallTransaction and have as the one and only parameter the name of the requested transaction.

MACROBUTTON R3CallTransaction VA02

This will call transaction VA02 when you click on the macrobutton in the text document. You can replace VA02 with the code of your transaction.

For more information see the Microsoft Office help for MACROBUTTON and Visual Basic.

Calling SAP R/3 from within WORD 97 with a mouse click

Word 97 Macro by Axel Angeli Logos! Informatik GmbH D-68782 Bruehl From website http://www.logosworld.com

This WORD 97 document contains a Visual Basic Project which allows to call SAP R/3 transaction using the SAP automation GUI. The call is done via the WORD field insertion MACROBUTTON. You must have the SAP Automation GUI or SAP RFC Development Kit installed on your workstation to give SAP the required OLE functionality.

Example:

Click to start transaction { MACROBUTTON R3CallTransaction VA02 } and another call to { MACROBUTTON R3CallTransaction VA02 }.

To show the coding of the MACROBUTTON statement, right-mouse-click on the transaction code link and choose 'Toggle Field Codes'.
Illustration 17: Visual Basic code with macros to call R/3 from WORD 97

```
Dim fns As Object
Dim conn As Object
Dim SAP_logon As Boolean

Sub R3CallTransaction()
' get the TCODE from the WORD text, MACROBUTTON does not allow parameters
ll = Len("MACROBUTTON R3CallTransaction ") + 3
tcode = Mid$(tcode, ll)
R3CallTransactionExecute (tcode)
End Sub

Sub R3CallTransactionExecute(tcode)
On Error GoTo ErrCallTransaction
R3Logon_If_Necessary
Result = fns.RFC_CALL_TRANSACTION(Exception, tcode:=tcode)
the_exception = Exception
ErrCallTransaction: ' Error Handler General
Debug.Print Err
If Err = 438 Then
   MsgBox "Function module not found or RFC disabled"
   R3Logoff  ' Logoff to release the connection !!!
   Exit Sub
Else
   MsgBox Err.Description
End If
End Sub

Sub R3Logon_If_Necessary()
If SAP_logon <> 1 Then R3Logon
End Sub

Sub R3Logon()
SAP_logon = False
Set fns = CreateObject("SAP.Functions")   ' Create functions object
fns.logfilename = "wdtflog.txt"
fns.loglevel = 1
Set conn = fns.connection
conn.ApplicationServer = "r3"
conn.System = "DEV"
conn.user = "userid"
conn.Client = "001"
conn.Language = "E"
conn.tracel level = 6
conn.RFCWithDialog = True
If conn.logon(0, False) <> True Then
   MsgBox "Cannot logon!.
   Exit Sub
Else
   SAP_logon = conn.IsConnected
End If
End Sub

Sub R3Logoff()
conn.logoff
SAP_logon = False
End Sub
```

13.3 R/3 RFC from JavaScript

JavaScript is a fully object oriented language. Therefore you can easily connect from JavaScript to R/3 via the CORBA compatible object library (in WINDOWS known also DLLs or ACTIVE-X (=OLE/2) components).

JavaScript is a typical object oriented language which is compliant to basic CORBA, DCOM and other popular object standards.

SAP R/3 provides a set of object libraries, which can be registered with JavaScript to allow RFC calls to R/3.

**DLLs installed with SAPGUI**
The libraries are installed to the workstation with the SAPGUI installation.

The object library SAP contains among others the object type FUNCTIONS whose basic method CALL performs an RFC call to a specified R/3 function module.

**Try to call standard routines for testing**
If the RFC call appears to be not working, you should first try out to call one of the standard R/3 RFC functions like RFC_CALL_TRANSACTION_USING (calls a specified transaction) or RFC_GET_TABLE (returns the content of a specified R/3 database table).

Illustration 17: HTML Page with a button to call a transaction via RFC
13.4 R/3 RFC/OLE Troubleshooting

Problems connecting via RFC can usually be solved by reinstalling the full SAPGUI and/or checking your network connection with R/3.

**Reinstall the full SAPGUI**

If you have problems to connect to R/3 via the RFC DLLs then you should check your network installation. It would be out of the reach of this publication to detail the causes and solutions when an RFC connection does not work.

I may say, that in most cases a full install of the SAPGUI on the computer which runs the calling program will secure a reliable connection, provided that you can login to R/3 problem-free with this very same SAPGUI installation.

Another trivial but often cause are simple network problems. So impossible it may appear, you should always go by the book and first check the network connection by pinging the R/3 system with the PING utility and checking the proper access authorities.

However, if you successfully passed the SAPlogon method, then the problem is mostly a misspelling of object or method names or an incompatibility of the called function.

**Check spelling**

If you are quite sure that you spelled everything right and correct, and still get an error executing the SAP.FUNCTIONS.CALL method then you should investigate the function module in R/3.

**Check for syntax errors**

Generate the function group to see if there is an syntax error.

Make sure that the function is tagged as RFC allowed.

---

Program 9: JavaScript example to call an R/3 function module via OLE/RFC

```javascript
<script language="JavaScript">

let retcd = 0;
exceptions = 0;
// *** SAPlogon() creates an object that has the methods to
// execute a call to an SAP function module

function SAPlogon()
{
  fns = new ActiveXObject("SAP.Functions");
  trans = fns.Transactions;
  conn = fns.connection;
  // get a new connection

  conn.System = "DEV"; /* Set the system ID (see: SY-SYSID) */
  conn.user = "userid"; /* set userid (blank for dialog) */
  conn.password = ""; /* set password (blank for dialog) */
  conn.Client = "100"; /* set language (blank for default) */
  conn.Language = "E"; /* set language (blank for default) */
  conn.tracelevel = 6; /* set password (blank for dialog) */
  conn.RFCWithDialog = 1; /* true: opens visible session window */

  exceptions = 0;
  conn.logon(0, 0); /* *** this call creates the object *** */
}

function SAPlogoff()
{
  conn.logoff(0, 0);
  exceptions = 0;
}

function SAPcallTransaction(tcode)

  exceptions = 0;

  callta = fns.add("RFC_CALL_TRANSACTION_USING");
  callta.exports("TCODE") = "VA02";
  callta.exports("MODE") = "E";
  retcd = callta.call;

  conn.logoff();
  alert(retcd);

  return retcd;
}

SAPcallTransaction = retcd;

// *** execute the SAP function MODULE "RFC_CALL_TRANSACTION_USING" as a method execution of object type SAP.functions

function SAPcallTransaction(tcode)
{
  exceptions = 0;

  callta = fns.add("RFC_CALL_TRANSACTION_USING");
  callta.exports("TCODE") = "VA02";
  callta.exports("MODE") = "E";

  retcd = callta.call;

  exceptions = 0;

  conn.logoff();

  alert(retcd);

  return retcd;
}

// ---//</script>

<body>
</body>
<input TYPE = "submit" VALUE = "VA02"
ONCLICK = "SAPlogon(); SAPcallTransaction( &quot;VA02&quot; ); SAPlogoff()"
>
</body>
```

---