

27.5.1 Klasse RpcServer

Ein Objekt der Klasse `RpcServer` (`gb.xml.rpc`) können Sie so erzeugen:

```
Dim hRpcServer As RpcServer
hRpcServer = New RpcServer() As "EVENT-NAME"
```

27.5.1.1 Eigenschaften

Die Klasse `RpcServer` hat genau zwei Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Count	Integer	Gibt die Anzahl der maximal zulässigen Verbindungen an.
Listening	Boolean	Ist True, wenn der RPC-Server am vorgegebenen Port lauscht.

Tabelle 27.5.1.1.1 : Eigenschaften der Klasse `RpcServer`

27.5.1.2 Methoden

Hier finden Sie die Beschreibung der Methoden der Klasse `RpcServer`:

Methode	Beschreibung
<code>Listen(Port As Integer, MaxConn As Integer)</code>	Der Server lauscht am angegebenen Port.
<code>Register(remoteFunction As RpcFunction)</code>	Die angegebene <code>RpcFunction</code> wird am Server registriert.
<code>Unregister(methodName As String)</code>	Die registrierte Methode wird aus der Liste der registrierten Methoden entfernt.
<code>SetReply(Data As Variant)</code>	Der Funktionswert der aufgerufenen RPC-Funktion wird an den Client gesendet.
<code>Stop()</code>	Stoppt den RPC-Server und zerstört das <code>RPC-Server-Objekt</code> .

Tabelle 27.5.1.2.1 : Methoden der Klasse `RpcServer`

27.5.1.3 Ereignisse

Die Klasse `RpcServer` verfügt nur über ein Ereignis:

```
RemoteCall(Method As String, Params As Variant[])
```

Das Ereignis wird ausgelöst, wenn eine Anfrage (Request) von einem Client eingeht und die aufzurufende Funktion (Method) mit den erforderlichen Argumenten als Variant-Array (Params) übergeben wird. In der Ereignis-Behandlungsroutine wird die angeforderte Funktion in der Liste der registrierten Methoden über ihren Namen in einer Auswahl-Kontrollstruktur ausgewählt und die zugeordnete (interne) Prozedur zur Berechnung des Funktionswertes mit den Eingabe-Argumenten aufgerufen, in der mit der Gambas-Prozedur `RpcServer.SetReply(...)` der Funktionswert an den Client (Response) zurück gegeben wird:

```
Public Sub hXMLRPC_RemoteCall(sMethod As String, aParameters As Variant[])
  Select Case sMethod
    Case "add2integer" ' 1. Service
      Sum(aParameters)
    End Select
End

Public Sub Sum(aParameters As Variant[])
  hXMLRPC.SetReply(aParameters[0] + aParameters[1])
End
```

27.5.1.4 Projekt XML-RPC-Server

Der vorgestellte XML-RPC-Server nutzt die Klasse `RpcServer` und stellt nach dem Start neben den 3 (Standard-)Methoden zur Inspektion des Servers nur die eine Funktion 'add2integer' als lokalen Dienst auf dem System auf dem Port 1088 zur Verfügung → Kapitel 27.5.2 XML-RPC-Client.

Der Quelltext für den Server in einem Modul umfasst überraschend wenige Zeilen:

```
' Gambas module file

Public hXMLRPC As RpcServer

Public Sub Main()

  Dim hXMLRPCFunction1 As RpcFunction
  Dim iXMLRPCPort, iXMLRPCServerMaxConnections As Integer

  hXMLRPC = New RpcServer As "hXMLRPC"
  iXMLRPCPort = 1088 ' This Port officially not used or reserved
  iXMLRPCServerMaxConnections = 10

  hXMLRPCFunction1 = New RpcFunction("add2integer", [XmlRpc.xInteger, XmlRpc.xInteger], XmlRpc.xInteger)
  hXMLRPCFunction1.Help = ("Addition of 2 integers")
  hXMLRPC.Register(hXMLRPCFunction1)

  Try hXMLRPC.Listen(iXMLRPCPort, iXMLRPCServerMaxConnections)
  If Error Then
    Print ("Error-Text: ") & Error.Text & gb.NewLine & ("Error-Location: ") & Error.Where
    hXMLRPC.Unregister(hXMLRPCFunction1.MethodName)
    hXMLRPC.Stop()
  Endif
  If hXMLRPC.Listening Then
    Print ("The XMLRPC server listens for port ") & Str(iXMLRPCPort)
  Endif

End

Public Sub hXMLRPC_RemoteCall(sMethod As String, aParameters As Variant[])
  Select Case sMethod
    Case "add2integer" ' 1. Service
      Sum(aParameters)
  End Select
End

Public Sub Sum(aParameters As Variant[])
  hXMLRPC.SetReply(aParameters[0] + aParameters[1])
End
```

Es sind nur wenige ergänzende oder zusätzliche Zeilen notwendig, um einen weiteren Dienst zu implementieren:

```
Dim hXMLRPCFunction1, hXMLRPCFunction2 As RpcFunction

hXMLRPCFunction2 = New RpcFunction("sub2integer", [XmlRpc.xInteger, XmlRpc.xInteger], XmlRpc.xInteger)
hXMLRPCFunction2.Help = ("Subtraction of 2 integers")
hXMLRPC.Register(hXMLRPCFunction2)

Select Case sMethod
  Case "add2integer" ' 1. Service
    Sum(aParameters)
  Case "sub2integer" ' 2. Service
    Sub(aParameters)
End Select

Public Sub Sub(aParameters As Variant[])
  hXMLRPC.SetReply(aParameters[0] - aParameters[1])
End
```

Bei einer erneuten Inspektion des RPC-Servers zeigt sich bei einem RPC-Client der hinzugefügte Dienst in der Liste der implementierten Funktionen auf dem RPC-Server:

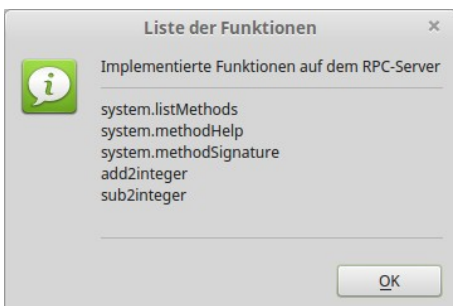


Abbildung 27.5.1.4.1: Liste der implementierten Dienste (Funktionen)

27.5.1.5 Projekt PHP-XML-RPC-Server

Der Server im zweiten Projekt bietet auch einen Service an. Der XML-RPC-Server wird immer dann gestartet, wenn die PHP-Datei *xmlrpc.server.php* im Web-Ordner *public_html* aufgerufen wird. Das setzt aber voraus, dass auf Ihrem System ein entsprechend konfigurierter HTTP-Server läuft. Das Kapitel *24.13_Exkurs_Webserver_LIGHTTPD* beschreibt ausführlich die Installation und Konfiguration dieses HTTP-Servers, der uneingeschränkt empfohlen werden kann!

Als RPC-Funktion ist ebenso wie im ersten Projekt *add2integer* implementiert.

Das ist der Inhalt der PHP-Server-Datei, die auf mehrere Bibliotheken von Edd Dumbill zurückgreift. Die Bibliotheken müssen im Verzeichnis *~/public_html/rpclib* gespeichert werden, weil das PHP-Skript sie dort erwartet. Die zwei benötigten Bibliotheken werden Ihnen als Archiv zur Verfügung gestellt.

```
<?php
// Kamil: Complete, simplest possible XMLRPC Server implementation

include "rpplib/xmlrpc.inc"; // from http://phpxmlrpc.sourceforge.net/ - version 3
include "rpplib/xmlrpcs.inc"; // from http://phpxmlrpc.sourceforge.net/ - version 3

// function that handles add2integer XML-RPC method
// function "mapping" is defined in 'new xmlrpc_server' constructor parameter.
function add2integer ($xmlrpcmsg)
{
    $summand_1 = $xmlrpcmsg->getParam(0)->scalarval(); // first parameter becomes variable $summand_1
    $summand_2 = $xmlrpcmsg->getParam(1)->scalarval(); // second parameter becomes variable $summand_2

    $result = $summand_1 + $summand_2; // calculating result

    $xmlrpcretval = new xmlrpcval($result, "int"); // creating value object - type integer
    $xmlrpcresponse = new xmlrpcresp($xmlrpcretval); // creating (new) response object

    return $xmlrpcresponse; // returning response
}

// creating XML-RPC server object that handles only 1 method
$s = new xmlrpc_server(
    array(
        "add2integer" => array( // xml-rpc function/method name
            "function" => "add2integer", // php function name to use when "add2integer" is called
            // signature with defined IN and OUT parameter types
            "signature" => array(array($xmlrpcInt, $xmlrpcInt, $xmlrpcInt)),
            "docstring" => 'Returns sum = summand1 + summand2' // description of method
        )));
?>
```

Hinweis: Wenn Sie die Grundinstallation von PHP ihres Betriebssystems verwenden, müssen Sie u.U. das Erweiterungspaket 'php5-xmlrpc' oder 'php-xmlrpc' nachinstallieren und dann Ihren Webserver erneut starten. Für die Erprobung des Projektes wurde der o.a. Web-Server Lighttpd eingesetzt.