



**Version 7.63**

## **DPCComm.dll**

### ***API Documentation***

Information in this document is subject to change without notice and does not represent a commitment on the part of Data Pro Accounting Software, Inc. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. The purchaser may make one copy of this software for backup purposes. No part of this manual or other materials included with the package, may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopying and recording for any purpose, other than the purchaser's personal use, without the written permission of Data Pro Accounting Software, Inc.

© 1985-2024 Data Pro Accounting Software, Inc.

Data Pro Accounting Software is a trademark of Data Pro Accounting Software, Inc.

No investigation was made into the common-law trademark rights of any word. Every effort was made to capitalize or highlight, in some manner, any word with current registrations.

All companies, products, addresses, and persons contained herein are intended to be completely fictitious and are designed solely to document and exemplify the use of Data Pro Accounting Software, Inc.'s products.

# DPCComm.dll API Documentation

This document outlines the Application Programming Interface (**API**) of the Data Pro **COM** object **DPCComm.dll**. Messages are passed to and from the Data Pro **POWERServer** module using this **COM** object. This **DLL** can be used in conjunction with an application development tool such as **Delphi, Visual Basic, Visual C++, ASP, VBScript**, etc. to send and receive messages from the **Data Pro POWERServer** TCP/IP server process to perform database requests and transaction updates to the accounting system.

As a **COM** object, this **DPCComm.dll** file must be installed on the client machine (*usually* **c:\windows\system32**) and registered with the Windows operating system.

**For example: regsvr32 c:\windows\system32\DPCComm.dll**

All string variables are considered **NULL** terminated character arrays with a maximum size of **1024** characters. All integers are considered **2-byte** short integers.

# DPCComm.dll POWERServer Methods (functions)

## DPCConnect (Hostname, Portnumber)

### Description:

This function is used to perform the initial TCP/IP connection to the **POWERServer Host** system.

### Parameters:

The **HostName** variable is a text string containing the domain name of the **POWERServer Host**.

The **PortNumber** variable is a **2-byte** integer containing the TCP/IP Socket Port to connect to. If the **PortNumber** value is a negative number, the Socket Port will be the absolute value of the number and all subsequent messages will be sent and received in an encrypted format.

### Return Value:

This function returns a **2-byte** integer which is zero if the function is successful. The return value is **-1** if the function fails.

## DPDisconnect ()

### Description:

This function is used to disconnect from the **POWERServer Host** system. This function must be called before exiting the application.

### Parameters:

There are no parameters required for this function.

### Return Value:

This function returns a **2-byte** integer which contains a zero if the function call was successful.

## **DPGetMessage ()**

### **Description:**

This function is used to retrieve the response string from the **POWERServer Host** system as a result of the previously sent message.

### **Parameters:**

There are no parameters required for this function.

### **Return Value:**

This function returns a text string with the response from the **POWERServer Host** of the previously sent request message. The size of this string variable must be large enough to contain the expected response, but will never exceed **1024** bytes.

## **DPGetParamCount ()**

### **Description:**

This function is used to retrieve the number of parameters contained in the **POWERServer Host** system response string as a result of the previously sent message.

### **Parameters:**

There are no parameters required for this function.

### **Return Value:**

This function returns a **2-byte** integer which contains the number of parameters contained in the response string.

## **DPGetResponseParam (ParamNumber)**

### **Description:**

This function is used to retrieve a specific response parameter string from the **POWERServer Host** system as a result of the previously sent message.

### **Parameters:**

The **ParamNumber** is a **2-byte** integer which specifies the parameter number of the response string to return.

### **Return Value:**

This function returns a text string with the response parameter from the **POWERServer Host** of the previously sent request message. The size of this string variable must be large enough to contain the expected response, but will never exceed **1024** bytes.

## **DPGetResultCode ()**

### **Description:**

This function is used to retrieve the result code which the **POWERServer Host** system sets in response to the previously sent message.

### **Parameters:**

This function requires no parameters.

### **Return Value:**

This function returns a **2-byte** integer which contains the value of the result code of the previously sent message. This value should be zero if the result of the previous message was successful. A value less than zero indicates the previous request message was not successful. When a non-zero result code is returned, a call to the function **DPGetMessage ()** will return the complete text response from the **POWERServer Host** system which describes the reason for the failure.

## **DPSendMessage (MessageString)**

### **Description:**

This function is used to send a message to the **POWERServer Host** system.

### **Parameters:**

The **MessageString** variable is a text string containing the request message to be sent to the **POWERServer Host**.

### **Return Value:**

This function returns a **2-byte** integer which is zero if the function is successful. The return value is **-1** if the function fails.

### Example VB Code:

```
Dim objPOWER, HostName, PortNumber, ParamNumber, MessageStr, ResultCode, ResponseStr
```

```
HostName = "test.com"
```

```
PortNumber = 9500
```

```
set objPOWER = CreateObject("DPComm.POWERServer")
```

```
objPOWER.DPConnect HostName, PortNumber
```

```
If objPOWER.DPConnect(HostName, PortNumber) = 0 Then
```

**‘This message will set the company and date.’**

```
If objPOWER.DPSendMessage("set~company=ins,date=clock") = 0 Then
```

**‘This message will retrieve the item number and description of  
‘all inventory items.**

```
MessageStr= "getrec~filetype=im01,begkey=#f,endkey=#l,fields=(num,desc)"
```

```
ResultCode = objPOWER.DPSendMessage(MessageStr)
```

```
While ResultCode = 0
```

**‘This will retrieve the entire server response string.**

```
ResponseStr = objPOWER.DPGetMessage()
```

**‘This will retrieve the number of parameters in the response  
‘string.**

```
ParamNumber = objPOWER.DPGetParamCount()
```

**‘This will retrieve a specific parameter from the response string.**

```
ResponseStr = objPOWER.DPGetResponseParam(1)
```

```
MessageStr = "getrec~next"
```

```
ResultCode = objPOWER.DPSendMessage(MessageStr)
```

```
Wend
```

```
End If
```

```
End If
```

**‘This message will disconnect from the POWERServer.**

```
objPOWER.DPDisconnect()
```