

# PhoenixSpeak Parser

---

Author:       Aleksi Asikainen ([sanct@furcadia.com](mailto:sanct@furcadia.com))  
Last Update:   2007-10-31  
Scope:        Describe the interface of PSParser ActiveX Control

## Introduction

PSParser is a small ActiveX control that parses Furcadia PhoenixSpeak responses into a machine-readable form.

The parser is used by feeding Furcadia PhoenixSpeak responses (lines that begin with "(PS ") to the control by calling the parse() function, and then querying the results from the control by using the functions listed below.

It should be noted that the parser is only guaranteed to work, if PhoenixSpeak verbosity (`ps verbosity) is set to level 3 or less.

## Example 1 – Parsing `ps get

```
client>    `ps 1202 get *  
server>   (PS 1202 Ok: get: result: test='hello', test2='world!')
```

### Visual Basic 6:

```
psparser1.parse "(PS 1202 Ok: get: result: test='hello', test2='world!'"  
  
if psparser1.getCmd() = psparser1.CMD_GET and _  
  psparser1.getStatus() = psparser1.STATUS_OK and _  
  psparser1.isDebug() = false _  
then  
  debug.print "Query " & psparser1.getQueryID() & ":" & vbCrLf & vbCrLf  
  
  for i = 0 to psparser1.getFieldCount() - 1  
    debug.print psparser1.getFieldName( i ) & "=" & _  
      psparser1.getFieldValue( i ) & vbCrLf  
  next i  
else  
  debug.print "Error: " & psparser1.getError() & vbCrLf  
endif
```

### Output:

```
Query 1202:
test=hello
test2=world!
```

## Example 2 – Parsing `ps set

```
client>   `ps 1203 set test.moo='Hello world!'
server>   (PS 1203 Ok: set: ok
```

### Visual Basic 6:

```
if psparser1.getCmd() = psparser1.CMD_SET and _
    psparser1.getStatus() = psparser1.STATUS_OK and _
    psparser1.isDebug() = false _
then
    debug.print "Query " & psparser1.getQueryID() & ": " & psparser1.getMessage() & vbCrLf
else
    debug.print "Error: " & psparser1.getError() & vbCrLf
endif
```

### Output:

```
Query 1203: ok
```

## Example 3 – Using Hierarchy

### Visual Basic 6:

```
psparser1.setHierarchy( "this.is.a[ test ][ hierarchy ]" )

for i = 0 to psparser1.getHierarchyCount() - 1
    debug.print "Hierarchy " & i & ": " & psparser1.getHierarchyName( i ) & vbCrLf
next i
```

### Output:

```
Hierarchy 0: this
Hierarchy 1: is
Hierarchy 2: a
Hierarchy 3: test
Hierarchy 4: hierarchy
```

## Function List

### **parse( String/BSTR responseLine )**

Parse the Furcadia PhoenixSpeak response into a machine-readable form.

### **long getFieldCount()**

Return the total number of fields in the response. The return value is always zero, if the parsed response is not a successful 'ps get'.

### **String/BSTR getFieldName( long fieldNo )**

Return the name of the field *fieldNo*. If *fieldNo* is smaller than zero, or *fieldNo* is greater or equal to getFieldCount(), or the parsed response is not a successful 'ps get', the function will return an empty string.

### **String/BSTR getFieldValue( long fieldNo )**

Return the value of the field *fieldNo*. If *fieldNo* is smaller than zero, or *fieldNo* is greater or equal to getFieldCount(), or the parsed response is not a successful 'ps get', the function will return an empty string.

### **long getCmd()**

Return the enumerated id of the command prompting the response. The value will match to one of the CMD\_\* constants.

### **long getStatus()**

Return the enumerated id of the status of the response. The value will match to one of the STATUS\_\* constants.

### **long getQueryID()**

Return the query ID of the parsed line, or zero, if no query ID was used.

### **String/BSTR getMessage()**

Return the message portion of the parsed line, or an empty string, if no message is available.

### **String/BSTR getError()**

Return the error message portion of the parsed line. If getStatus() is not STATUS\_ERROR, the function will return an empty string.

### **String/BSTR getCmdStr()**

Return the command prompting the response in the string form.

### **String/BSTR getStatusStr()**

Return the status of the response in the string form.

### **String/BSTR getResponseStr()**

Return the entire PS response line as passed to the parse() function.

### **Boolean/VARIANT\_BOOL isDebug()**

Return true, if the PS line is recognized as a debug message, false otherwise.

### **Boolean/VARIANT\_BOOL setHierarchy( String/BSTR hierarchyStr )**

Parse the hierarchy in hierarchyStr and return true, if successful.

### **long getHierarchyCount()**

Return the total number of hierarchy levels in *hierarchyStr* passed with the previous *setHierarchy()* call.

### **String/BSTR getHierarchyName( long hierarchyLevel )**

Return the name of the hierarchy level *hierarchyLevel*. If *setHierarchy()* has not been called prior to *getHierarchyName()*, *hierarchyLevel* is less than zero or more or equal to *getHierarchyCount()*, the function will return an empty string.

## Constants

Constant	Description
<b>CMD_UNKNOWN</b>	The processed string was not recognized as a PhoenixSpeak response.
<b>CMD_ACTIVE_DB</b>	Response to `ps active_db
<b>CMD_CLEAR</b>	Response to `ps clear
<b>CMD_DB</b>	Response to `ps db
<b>CMD_DB_COPY</b>	Response to `ps db_copy
<b>CMD_GET</b>	Response to `ps get
<b>CMD_LIST_ACCESS</b>	Response to `ps list_access
<b>CMD_MEMORY</b>	Response to `ps memory
<b>CMD_RM_ACCESS</b>	Response to `ps rm_access
<b>CMD_SET</b>	Response to `ps set
<b>CMD_SET_ACCESS</b>	Response to `ps set_access
<b>CMD_VERBOSITY</b>	Response to `ps verbosity
<b>STATUS_UNKNOWN</b>	The processed string was not recognized as a PhoenixSpeak response.
<b>STATUS_OK</b>	The command was successful.
<b>STATUS_ERROR</b>	The command was not successful.
<b>STATUS_NOTIFY</b>	The processed string is a warning or a notification regarding a command.
<b>STATUS_DEBUG</b>	The processed string is a debug message.
<b>STATUS_DEBUG_DS</b>	The processed string is a debug message originating from DragonSpeak
<b>STATUS_PARTIAL</b>	The processed string contained a message that has more than one part. To process the entire string, feed the following parts to the control through parse() – when complete, the control will return the real status code.