

SQL iQuery Built-in Functions

SQL IQQUERY DOCUMENTATION

BOB COZZI



This manual contains documentation for the SQL iQuery for IBM i scripting language built-in functions. These built-in functions may be used on Conditional, Assignment and Looping statements within an SQL iQuery Script.

For installation instructions please see Appendix A near the end of this document.

Built-in Functions

The following Built-in Functions are supported in SQL iQuery. They may be used on assignment or conditional statements. Built-in Functions are case insensitive and often have optional parameters. When multiple parameters are specified, a comma is used to separate each parameter.

While the value returned by built-in functions is always considered textual, numeric values that result from built-in functions may be used like any other numeric value.

Return Value	Built-in	Parameters	Description
Numeric position	CHKLIBLE(library)	Library is the name of a library.	The CHKLIBLE and its alias CHKLIBL (Check Library List) function scans the library list for the Library name and returns non-zero if the library is found.
String	COMPRESS(string)	String a session variable or literal whose content is compressed by removed blanks and non-alphanumeric data.	The COMPRESS (remove all but valid name characters) compresses a string into a new string containing a <i>Valid Name</i> . This new string will have blanks removed and any non-alphanumeric characters are filtered out. For example, COMPRESS('Hello World!!') would return: HELLOWORLD to the caller.
BOOL	DDE(session variable)	Session Variable that received data from a previous SELECT INTO statement.	The DDE (returned column had a Decimal Data Error) function returned true when the corresponding column of a SELECT INTO contains legacy decimal data errors. See also "NULL()" function.
BOOL	DEFINED(session variable)	Session Variable is any iQuery Script Session variable name.	The DEFINED (Is Session Variable Defined) function returns a logical true condition when the session variable exist.
String	DTAARA(data area, start, len)	Data Area is a qualified data area name. Start is the location within the data area where the extraction of data begins. Len is the number of bytes to return.	The DTAARA (Retrieve Data Area) function extracts data from an existing data area. The data is stored in the resulting string.

Return Value	Built-in	Parameters	Description
Integer	ELEM(session variable)	Session Variable is any iQuery Script Session variable name.	The ELEM or its alias ELEMS (Array Element Count) returns the current number of elements in the session variable array. If the session variable is not an array, then by definition ELEM returns 1.
String	FILTER(string, 'characters to be filtered', ['replacement-character'])	String a session variable or literal whose content is scanned for any of the characters on parameter 2, and replaced with the characters of parameter 3.	The FILTER (remove all characters) filters a string by removing the characters identified in Parameter 2. The replacement character is a blank or may be specified in parameter 3.
Numeric position	FIRSTOF(values, string, start position)	Values is a list of one or more characters. The String parameter is search for the first occurrence of any of the characters listed on this parameter. String is a session variable or literal that is search for the characters in the Values parameter. Start position (optional) is the first position in String to search. If Start is unspecified the search begins at the first position.	The FIRSTOF (Find first of) function returns the position of the first occurrence of any character specified on the first (Values) parameter. If none of the characters are found, 0 is returned.
String	GETCOOKIE(string)	String is the name of a CGI cookie sent to the application by a web browser. The value of the cookie is returned.	The GETCOOKIE (get CGI cookie) returns the value of the specified CGI Cookie. This is usefully when iQuery for Web is being used and your SQL iQuery script has been called by aCGI program such as WEBiQRY.

Return Value	Built-in	Parameters	Description
String	GETENV(env var, start, length)	<p>Env Var is any Environment Variable whose value is to be returned.</p> <p>Start (optional) is the first position of the returned environment variable value to be returned.</p> <p>Length (option) is the maximum number of bytes to be returned.</p>	The GETENV (Retrieve Environment Variable Value) function returns the value for the specified Environment Variable.
String	GETFILE(ifs file name)	IFS File Name is the name of an existing IFS file whose entire contents is retrieved and copied to the target string or session variable.	The GETFILE (Load IFS File Contents) function reads the entire file and returns its content.
BOOL	ISWEB()		The ISWEB or its alias ISCGI (Is app running in a CGI environment) function returns true when iQuery is being run in a CGI/Web environment. The iQuery WEBIQRY program initializes an internal area with this flag which is tested when the ISWEB() function is tested.
Integer	LEN(session variable)	Session Variable is any iQuery Script Session variable name.	The LEN and its alias STRLEN (return string length) returns the length of the contents in the specified Session Variable. Any trailing blanks are trimmed from the value before the length is calculated.

Return Value	Built-in	Parameters	Description
String	MSGID(msgid, msgfile, msgdata, 1 2)	<p>MsgID is any 7-character msg ID whose message text is to be retrieved.</p> <p>MsgFile is the name of a qualified message file.</p> <p>MsgData is an option string containing message substitution data for the retrieved value.</p> <p>Level is either the number 1 (the default) or 2. 1 = 1st level message text is returned. 2 = 2nd level message text is returned.</p>	The MSGID (Retrieve 1 st or 2 nd level message text) function returns the message text for the specified message ID.
BOOL	NULL(session variable)	Session Variable is the name of a variable associated with a column from a previous SELECT INTO statement.	The NULL (is NULL) function returns true when the corresponding column of a SELECT INTO returned a NULL value. See also "DDE()" function.
String	PARSEVAR(session variable)	Session Variable is the name of an iQuery Script Session Variable.	The PARSEVAR (Parse String for Substitution Variables) is used to process the contents of a Session Variable as if it were a line/statement in the Script. Any Session Variables that appear in the Session Variable are translated to their corresponding values. This is primarily an internal-user only function but is being exposed and documented for end-user consideration.

Return Value	Built-in	Parameters	Description
Numeric position	SCAN(Pattern, Scan Data, Start, Len)	<p>Pattern is the data to search for.</p> <p>Scan Data is the data to be scanned for the Pattern.</p> <p>Start (Optional) is the starting position within the Scan Data to start the search.</p> <p>Len (Optional) is the number of characters from the start position to be searched.</p>	The SCAN (Scan String) function performs a case-sensitive scan of the Scan Data for the character string in the Pattern. If found, the position is returned.
Numeric position	SCANI(Pattern, Scan Data, Start, Len)	<p>Pattern is the data to search for.</p> <p>Scan Data is the data to be scanned for the Pattern.</p> <p>Start (Optional) is the starting position within the Scan Data to start the search.</p> <p>Len (Optional) is the number of characters from the start position to be searched.</p>	The SCANI (Scan String, ignore letter-case) function performs a case-insensitive scan of the Scan Data for the character string in the Pattern. If found, the position is returned.
String	SPLIT(session variable, delimiter(s))	<p>Session Variable is any iQuery Script Session variable name.</p> <p>Delimiter Characters (Optional) is a list of 1 or more characters that are used to split the session variable. If not specified, blanks are used.</p>	The SPLIT (Split String) function returns the data in the Session Variable up to but not including the first Delimiter. The Session Variable's content is then reduced by the returned value's length. Use this function with an array and a WHILE loop to extract pieces of a string stored in a session variable.

Return Value	Built-in	Parameters	Description
String	SST(value, start, len)	<p>Value is a session variable or literal that is to have part of it returned.</p> <p>Start is the position within the Value to start the substring.</p> <p>Len is the number of bytes from the Start position to retrieve. If unspecified, the extracted value starts at the start position and continues to the end of string.</p>	The SST (Substring) function returns a portion of the string specified on the first parameter.
String or Integer	SYSVAL(system value)	System Value is the name of the IBM i System Value whose value is being retrieved.	The SYSVAL (Retrieve System Value) function returns the value for the specified system value.
String	TEMPFILE() or TMPFILE() or TMPNAME() or TMPNAM()	None	The TMPNAME (Create Temporary IFS File Name) returns the name of a unique IFS file. Use this to create a new file and have the system assign a unique name to the file.
String	TOHEX(session variable)	Session Variable is any iQuery Script Session variable name.	The TOHEX (Convert to Hexadecimal) function converts the data in the iQuery Session Variable to its hexadecimal form.
String	TOLOWER(session variable)	Session Variable is any iQuery Script Session variable name.	The TOLOWER (Convert to Lower Case) function converts the data in the iQuery Session Variable to all lower case.
String	TOUPPER(session variable)	Session Variable is any iQuery Script Session variable name.	The TOUPPER (Convert to Upper Case) function converts the data in the iQuery Session Variable to all upper case.

Return Value	Built-in	Parameters	Description
<p>String</p>	<p>TRIML(session variable, 'chars to remove')</p> <p>TRIML(session variable, number of chars to remove)</p>	<p>Session Variable is any iQuery Script Session variable name.</p> <p>Chars to Remove is a list of one or more characters that are removed from the end (right-side) of the session variable.</p> <p>Number of Chars to Remove is an integer that specifies the number of characters to remove from the right-side of the session variable.</p>	<p>The Trim Length (TRIML) function removes trailing (right-side) characters from the Session variable and returns a string with the new value. It does not change the existing session variable content.</p> <p>In the first form, a list of characters may be specified. The TRIML function removes characters from the right-side of the session variable that matches any of the characters in the list. It stops when it encounters the first character that is not in the list of characters specified for the 2nd parameter.</p> <p>The second form of the function deletes the number of characters from the end (right-side) of the session variable. If the number is greater than then number of characters in the session variable, an empty value is returned.</p>
<p>String</p>	<p>USRSPC(user space, start, len)</p>	<p>User Space is a qualified user space name.</p> <p>Start is the location within the User Space where the extraction of data begins.</p> <p>Len is the number of bytes to return.</p>	<p>The USRSPC (Retrieve User Space Data) function extracts data from an existing user space object. The data is stored in the resulting string.</p>

Built-in Function Examples

Built-in	Example	Result	Description
CHKLIBL	<pre>If (chklibl('COZTOOLS') = 0); call addlibl('COZTOOLS'); endif;</pre>	Returns the position of the library in the library list or 0 if the library is not on the library list.	<p>The CHKLIBL() and CHKLIBLE() built-in functions search the entire library list for the library name specified. If it is found, its position in the library list is returned. For example, if COZTOOLS is the 7th library on the library list, then CHKLIBL('COZTOOLS') returns 7.</p> <p>If it is not found, 0 is returned. (NOTE: the ADDLIBLE SQL Stored Procedure is included with SQL iQuery)</p>
COMPRESS	<pre>eval &title = 'Hello World'; eval &jobname = sst(compress(&title),1,10); CL: SBMJOB JOB(&JOBNAME) CMD(...);</pre>	Returns a compressed form of the text string, removing blanks and non-alphanumeric characters.	<p>The COMPRESS() built-in function removes each non-alphanumeric character from the Session Variable, including blanks. The result is a "valid name" that may be used with IBM i system interfaces. The length of the result is not truncated, therefore if up to 10 positions are required, then the SST() built-in function may be used to extract the first (up to) 10 characters.</p>

Built-in	Example	Result	Description
DEFINED	<pre>If defined(&remote); Connect to &remote USER &USER using &PWD; endif;</pre>	Returns true if the Session Variable exists and is not empty.	The DEFINED() or DEFN() built-in functions test the Session Variable for Existence and if it exists if it is an empty value. If it is empty or it does not exist, this built-in function returns false, otherwise it returns true.
DTAARA	<pre>eval &pwd = dtaara(qgpl/qrypWD);</pre>	The contents of the QRYPWD data area is returned.	The contents of the QRYPWD data area in QGPL is retrieved and copied to the &PWD session variable. Since the start/length parameters are omitted, the entire data area is retrieved.
ELEM	<pre>#define &stg = "Hello There World" #define &x = 0 while &x < 3; eval &x += 1; eval &word[&x] = split(&stg); endwhile; eval &ArrayCnt = elems(&word);</pre>	The number of elements in the Session Variable named &WORD is returned.	The ELEM() and ELEMS() built-in functions return the current array element count for the specified Session Variable. Internally all Session Variables are arrays, but normally contain only one value. The ELEM() function allows you to determine how many array elements are stored in the Session variable. If the session variable contains the normal, one element, then 1 is returned.

Built-in	Example	Result	Description
FIRSTNOTOF and FINDFIRSTNOTOF	<pre>eval &part = 'VHS 1252'; eval &loc= findFirstNotOf('VHS12',&PART)</pre>	The location of the first character that is not one of the V H S 1 or 2 characters in the &PART session variable.	The findNotFirstOf() and firstNotOf() built-in functions return the first position in the 2nd parameter of any character that does not match one of the characters specified on the first parameter.
FIRSTOF and FINDFIRSTOF	<pre>eval &part = 'VHS 1252'; eval &loc= findFirstOf('0123456789',&PART)</pre>	The location of the first digit in the &PART session variable.	The findFirstOf() and firstOf() built-in functions return the first position in the 2nd parameter of any of the characters specified on the first parameter.
GETCOOKIE	<pre>if isWeb(); eval &id = getcookie('SESSIONID'); endif;</pre>	Returns the value associated with the Web Browser's cookie specified as the parameter.	The GETCOOKIE() built-in function scans the HTTP_COOKIE environment variable for a cookie name that matches the function's first parameter. If it finds one, the cookie's contents are copied to the target Session Variable. To set Cookies, see the SETCOOKIE command.
GETENV	<pre>Eval &jobCCSID = getenv('iq_CCSID');</pre>	Returns the value stored in the Environment Variable ("iq_CCSID" in this example).	The GETENV() built-in function retrieves the Environment Variable specified on the first parameter.
GETFILE	<pre>eval &msg = getfile('/home/msg.txt');</pre>	The entire contents of the MSG.TXT file is returned.	Reads the entire contents from the file and copies that data to the &MSG session variable. The file needs to be an ASCII text file, or a source file member.

Built-in	Example	Result	Description
ISWEB	<pre>If isWeb(); #http_output HTML endif;</pre>	Return true (1=1) if the iQuery Script is being run from a CGI application.	The ISWEB() and ISCGI() built-in functions can be used on conditional statements to control when a section of the script is run only when running from the web. For example, users may wish to control JSON formatting when the result is being sent to the web browser but may have different settings when being directed to an output IFS text file.
LEN	<pre>If len(&name) > 0;</pre>	The length of the data stored in the &NAME session variable is returned.	The LEN() and strlen() built-in functions return the length of the data stored in the session variable or quoted character string specified on the parameter. Trailing blanks are removed in performing the length calculation. If the session variable does not exist, 0 is returned.
MSGID	<pre>Eval &msg = msgid(QWX2221);</pre>	Returns the first or second level text for the specified MSGID. If not message file is specified, then QCPFMSG is used.	The MSGID() built-in function returns the first or 2nd level message text for the message ID specified. If Message Data (the 3rd parameter) is specified, that msgdta is inserted into the resulting message by the system before it is returned by the built-in function.

Built-in	Example	Result	Description
SCAN and SCANI	<pre>eval &value = "These are the times to remember" eval &pos = scan('time',&value);</pre>	The location of the word 'time' in the &VALUE session variable.	<p>The position of the text string 'time' within the string stored in the &VALUE session variable is returned. SCAN() is often used for "if contains" functionality.</p> <p>The SCAN() and SCANI() functions offer case-sensitive and insensitive scanning capabilities.</p> <p>SCAN() is case sensitive.</p> <p>SCANI() is case-insensitive.</p>
SPLIT	<pre>#define &stg = "Hello There World" #define &x = 0 while &x < 3; eval &x += 1; eval &word[&x] = split(&stg); endwhile;</pre>	3 elements in the &WORD array.	<p>The string in the &STG session variable is split up based on each embedded blank. The WHILE loop iterates 3 times (X starts at 0) and uses SPLIT to extract the next value in &STG up to the delimiter. All arrays are dynamically sized so &WORD ends up with 3 elements.</p> <ul style="list-style-type: none"> • &WORD[1] = 'Hello' • &WORD[2] = 'There' • &WORD[3] = 'World'
SST	<pre>If sst(sysval(qtimzon),7,3) = 'CST';</pre>	Positions 7 to 9 of the QTIMZON system value is returned.	The 3 positions of the QTIMZON (7 to 9) system value are compared to the 'CST' literal. If the time zone contains 'CST' in positions 7,8,9, then the condition is true.

Built-in	Example	Result	Description
SYSVAL	<pre>if sysval(QTIMZON) = 'QN0600CST2';</pre>	The Time Zone system value is returned.	The value for QTIMZON system value is retrieved. In this statement, it is compared with a literal and if it matches then the condition is true. System Values may be quoted or unquoted. That is SYSVAL(QTIMZON) and SYSVAL('QTIMZON') produce the same result.
TMPFILE	<pre>Eval &workfile = tmpfile(); #msg The File name is: &workfile</pre>	Returns a system-supplied unique IFS file name. When this example was run on our system it returned: The File name is: /tmp/QACXHHM4V3	The TMPFILE() and TMPNAME() functions cause the IBM i system to generate a unique IFS file name. Normally that file is located in the /tmp directory off of root. There is normally no suffix (such as .txt) associated with the temp file name.
TOHEX	<pre>#define &Code = 'ABCDEF' eval &hex = tohex(&code); #msg &code in hex is: &hex</pre>	Returns the message: ABCDEF in hex is: C1C2C3C4C5C6	The TOHEX() built-in function expands normal single-byte characters into 2-character hexadecimal values.
TOUPPER	<pre>IF toUpper(&REGION) = 'ENU';</pre>	The content of ®ION in uppercase.	The uppercase form of the data stored in the ®ION session variable is returned. The IF statement causes that data to be compared with 'ENU'.

Built-in	Example	Result	Description
USRSPC	<pre>eval &Company = usrspc(data1ib/company,21,50);</pre>	Positions 21 through 70 of the COMPANY data area are returned.	The contents of the User Space named COMPANY in library DATALIB is accessed. Positions 21 through 70 are returned and copied to the &COMPANY session variable.

Appendix A

Installation Instructions

SQL iQuery licensed program 2COZ-IQ4 must be installed with the IBM i RSTLICPGM (Restore License Program) CL command. To restore the product once a save file containing it has been loaded onto your IBM i system, run the following command:

```
RSTLICPGM LICPGM(2COZIQ4) DEV(*SAVF) SAVF(QGPL/IQUERY)
```

Normally the product is delivered via PC media and compressed into a ZIP file (Windows and Mac). To extract the iQuery.savf file from the iQuery.zip file use whatever extraction tools are available on your PC or Mac. To obtain the latest version of iQuery visit www.SQLiQuery.com and click on the download link. A license is required after the initial trial period has expired.

Once the iQuery.savf file is available, it must be uploaded to the IBM i server. You can use any method you desire, but the most common approach is to FTP the file from the PC to the IBM i system. When doing so, be sure your FTP software is sending the iQuery.savf file to the IBM i server using BINARY mode.

This example assumes you have an IBM i server named CHICAGO on your network and you're uploading the current version of iQuery from your PC.

I prefer to use a freeware product named FILEZILLA And drag drop the files onto the IBM i system, but this command line approach works great.

From the Windows Start menu, type in the CMD command and press Enter. You'll see a command window open. In that window, type in the following:

```
FTP CHICAGO <enter>
/* At this point FTP will start, */
/* you should be prompted for USER ID/password */
/* Enter your User ID and password, then press Enter. */
/* Then type in the following statements: */
namefmt 1
or
site namefmt 1
BINARY
PUT C:\iquery.savf /qsys.lib/qgpl.lib/iQuery.savf
QUIT
```

At this point the save file should be uploaded to QGPL. Most people do not realize that by naming the PC file with the .SAVF suffix, the IBM i FTP server knows you are sending it a save file, so it creates the save file for you automatically if it doesn't already exist. If the file suffix is anything else, such as .FILE and the save file does not already exist on the IBM i system, then the IBM i FTP server creates a non-save file (database table) which is unusable. To resolve this problem, either make sure the PC file contains the .SAVF suffix or create the Save File on the IBM i system before uploading.