BASIC+ Commands

This chapter provides you with a listing of available BASIC+ commands, appropriate syntax, their parameters, and code samples. In this chapter, command types include functions, subroutines, statements, operators, and routines.

Commands in this chapter are organized in alphabetical order, and may be pertinent to programming for Lotus Notes, OpenInsight, or both. Commands specific to programming in Lotus Notes or OpenInsight (native) tables are labeled with an icon to make identification easier. Those which are available in both situations are not labeled.

The next section, "Using the Command Reference," provides an overview of the information available for each command, and a key to the symbols used in this chapter.

For information about OpenInsight properties and events, refer to Chapter 4: Properties and Chapter 5: Events, respectively.

BASIC+ Command Reference

Command	Description
ABORT ALL Statement	The ABORT ALL statement forces the cessation of all engine processing. It guarantees that all processing that is in progress will terminate.
Abs function	The Abs function returns the absolute value of the expression. The absolute value of a real number is its distance from 0 (zero) without regard to direction from zero. Use the Abs function to convert a negative number to a positive number.
Activate_Save_Select subroutine	The Activate_Save_Select subroutine loads a saved list of keys into cursor 0.
Alias_Table subroutine	Creates a temporary synonym (an alias) for an existing table. The permanent database definition is not altered.
Alpha function	The Alpha function returns true (1) if the expression contains non-numeric characters (including punctuation). If the expression contains a numeric value, false (0) is returned.
And and Or logical operators	
ANSI_UNICODE Function	Converts an ANSI string to Unicode string.
ANSI_UTF8 function	Converts a variable containing ANSI characters to a variable containing the corresponding UTF_8 characters.
AppNote subroutine	Used to display QuickHelp built using the QuickHelp Designer in the User Interface Workspace. AppNotes are typically used from event handlers.
Assigned function	Tests if a variable has been assigned. Returns the opposite values from unassigned().
ATan function	Generates the trigonometric arc tangent of an angle.
Attach_Table subroutine	Temporarily includes a table in the database definition. To make this permanent, run Define_Database, or use the Database Manager.
BCol1 function	The BCol1 function is the binary version of Col1(). It will return the numeric value of the character position of the delimiter preceding the extracted data.
BCol2 function	The BCol2 function is the binary version of Col2(). It will return the numeric value of the character position of the delimiter following the extracted data.
Bit-wise operators (BitAnd, BitOr, BitXor, BitNot)	Perform bit-wise And, Or, and Xor logical functions on two numbers. BitNot yields the logical Not of a number.
Blank_Struct function	Builds a blank structure to pass to a DLL which will then fill in the structure's elements.
BRemove statement	Extracts substrings, including fields, values, and subvalues, from dynamic binary arrays, using ANSI characters 249 through 255 as delimiters.
Btree Indexes	
Btree.Extract subroutine	Searches one or more Btree indexes for data matching the search criteria passed in. Returns the keys to rows having matching data.
Btree.Read subroutine	Btree.Read reads a node of an indexed file (a ! or bang file) into a dimensioned array.

[Build_Struct function](#)	Assembles a binary structure out of BASIC+ variables.
[Calculate function](#)	Returns the result of evaluating a dictionary item.
[Call statement](#)	Executes an external subroutine.
[CallFunction Function](#)	Execute an OpenInsight stored procedure function remotely.
[CallSubroutine Function](#)	Execute an OpenInsight stored procedure subroutine remotely.
[Case statement](#)	Provides a framework for conditional branching. Evaluates one or more test expressions, then executes the relevant code at the corresponding internal subroutine label. A Case statement must be introduced with Begin Case and terminated with End Case.
[Change_Popup_Font subroutine](#)	Utility subroutine that will change the font of the text within all popups.
[Change_Window_Font subroutine](#)	Utility subroutine that will change the font of text within a window.
[Char function](#)	Converts the numeric value of an expression to its corresponding ASCII character string value.
[CheckLegalUser function](#)	Checks if the passed user/password combination is legal for the current application and, if so, returns the user level
[Clear_Table subroutine](#)	Clears the data or dictionary information from a table without deleting the table. This subroutine does not allow you to clear system tables.
[ClearFile statement](#)	Deletes all rows from a table, but retains the table structure.
[ClearSelect statement](#)	Sets the list of record keys for a specified cursor to null.
[CloseEngine Function](#)	Closes an OpenEngine handle referenced by CreateEngine().
[CloseQueue Function](#)	Close a queue handle created by an earlier call to the CreateQueue() function.
[Col1 function](#)	The Col1 function is set by the Field and [ ] functions, and returns the value of the character position of the delimiter preceding the data extracted.
[Col2 function](#)	The Col2 function is set by the Field and [ ] functions, and returns the numeric value of the character position of the delimiter following the extracted data.
[ColHeading statement](#)	Specifies the heading or headings that will appear above each column on a report generated with the text output commands.
[Collect.IXVals function](#)	Returns the list of index values for the specified indexed field in the specified file.
[ColLength statement](#)	Specifies the width for each column heading in a printed report.
[Common statement](#)	Global variables, other than pre-defined system variables, are implemented using labeled common blocks. A labeled common block groups a set of global variables using a specified name.
[CompDoc function](#)	The CompDoc function is used to store multiple discrete records within a single record.
[Compute_Dict function](#)	Returns the result of a dictionary calculation. This is a shell around the Calculate() function. It opens a dictionary to @DICT, sets @ID, reads @RECORD, and calls Calculate.
[ContextMenu function](#)	The ContextMenu function creates and executes Context Menus. A Context Menu is a right mouse click menu specific to a control.
[Control_Off routine](#)	Removes transaction and domain validation controls from Native Tables.
[Control_On routine](#)	Installs transaction and domain validation controls for a table.
[Convert statement](#)	Converts one set of characters in a string to another set.
[Copy_OS_To_Row routine](#)	Copies one or more operating system files to a native table.
[Copy_Row subroutine](#)	Copies a row or a group of rows from one table to another.

[Copy_Row_To_OS subroutine](#)	Copies one or more native table rows to operating system files.
[Copy_Table subroutine](#)	Copies a native table, the dictionary for the table, and indexing information if it exists, to a new name or location. The source and target table must be the same filing system type.
[CopyOSFile function](#)	Copies an operating system file to another operating system file.
[Cos function](#)	Calculates the trigonometric cosine of an angle.
[Count function](#)	Counts the number of occurrences of a substring in a string.
[Create_Dialog function](#)	Starts a new window (specified in dialogID) which becomes an owned window of the owner window (specified in parentID).
[Create_Index subroutine](#)	Creates a Btree, Cross Reference, or Relational index for a specified column in a table. The type of index you create dictates the calling syntax.
[Create_Symbolic routine](#)	Creates or redefines calculated (symbolic) columns in a native table dictionary.
[Create_Table subroutine](#)	Creates native tables and their dictionaries.
[Create_User subroutine](#)	Creates a new user or modifies information about an existing user in the current database.
[Create_Volume subroutine](#)	Creates or updates a volume definition, adding a row to SYSPTRS (the system table that stores volume definitions), or modifying the row if it exists.
[Create_XML_XSD_ Schema function](#)	Creates an XML Schema definition document (.XSD file) from fields in an OpenInsight table.
[Cross Reference Indexes](#)	Parameters in the following table are specific to Cross Reference indexes.
[CreateBinaryData Function](#)	
[CreateEngine Constants](#)	
[CreateEngine Function](#)	Creates a reference to an OpenEngine, running either locally or remotely.
[CreateQueue Function](#)	Creates a reference to a queue on an open engine created by the CreateEngine function.
[CreateRequest Function](#)	
[Date function](#)	Returns the current date in internal format.
[DCount function](#)	Counts the number of occurrences of a substring in a string, returning 1 if the there are no occurrences of the substring in the string.
[Debug statement](#)	Use in source code to activate the debugger at a particular point.
[DEBUGGER_FETC H Function](#)	This function provides methods for accessing variable data within a Debugger Intercept routine.
[Debugger_Replace ment Function](#)	The Debugger_Replacement function is used to intercept calls to debugger tool within OpenInsight.
[Declare_FCNS routine](#)	Compiles a .DLL definition record into BASIC+ callable functions.
[Declare statement](#)	Informs the compiler of external functions and subroutines.
[Define_Database subroutine](#)	Defines a database, using all currently attached tables.
[Define_Struct routine](#)	Starts the Structure Designer tool, from the System Monitor or Editor++.
[Delay subroutine](#)	Pauses program execution for specified number of seconds.
[Delete function](#)	Use to delete elements from a dynamic array. Data can be deleted from field, value, and subvalue positions.

Delete statement	Deletes a row from a table.
Delete_Index subroutine	Removes a Btree, Cross Reference, or Relational index from a specified column in a table.
Delete_Row routine	Deletes one or more specified rows from a table.
Delete_Save_Select subroutine	The Delete_Save_Select subroutine erases a saved list of keys.
Delete_Table subroutine	Deletes native tables and their dictionaries.
Delete_User subroutine	Removes a user from the current database.
Detach_Table subroutine	Temporarily prevents access to a single table or list of tables by removing them from the current database. The permanent database definition is not altered.
Detach_Volume subroutine	Temporarily excludes a volume from access by OpenInsight. To make this permanent, run Define_Database, or use Database Manager.
Dialog_Box function	This function is used to create a modal dialog box. It processes an event loop, as long as the dialog box exists, then returns any result to the calling procedure.
Dict_Depend function	Returns the column positions in a row that are required to perform a dictionary calculation.
Dimension statement	Declares the name of a matrix, and allocates storage for it. A matrix must be dimensioned before it can be referenced in a program.
Direct_Print Function	The Direct_Print function provides a method to print data directly to a printer. This function bypasses the use of print drivers enabling the use of escape codes to be sent directly to the printer.
Dir function	Returns directory information (size, date, and time) for an operating system file.
DirExists Function	Used to determine whether a specific directory exists within the operating system.
DirList function	Returns a list of filenames from a directory and mask specified in a previous InitDir statement. The mask is used to create a selection of file names, using the standard DOS wildcards "*" and "?".
DOSTIME subroutine	Returns the number of seconds between midnight and the system time, as of the time the DOSTIME subroutine was called.
Drive function	Returns the current operating system path from which OpenInsight is running.
DSGetProperty function	Gets the specified DataSet Object property; the DataSet Object is specified by the passed handle.
DSInstance function	Creates an instance of the DataSet specified by DSName using the specified Connection Object (created by XOInstance()).
DSMethod function	Executes a DataSet Object method; the DataSet Object is specified by the handle passed.
DSpace subroutine	Returns the amount of total space and available space (in bytes), for a particular drive.
DSSetProperty function	Sets a DataSet Object property; the DataSet Object is specified by the passed handle.
EditFind function	Searches edit controls, including edit tables, for user-specified text.
End_Dialog subroutine	Use to close a dialog box created by Dialog_Box or Create_Dialog.
End_Window function	Stops execution of a running window, optionally changing focus to another window or control.
End statement	Identifies the end of a block of statements, or the physical end of a program.
Equate statement	Use to initialize a constant to an expression.
ETMethod function	ETMethod provides services that are analogous to the Set_Property (properties ROWDATA, COLDATA, LIST, SORTEDCOL) and Send_Message (messages INSERT, DELETE) functionality with the important difference that those services first attempt to modify bound DataSets.
Exp function	Exponential function that calculates the result of base e raised to the power designated by expression.
Extract function	Extracts a field, value, or subvalue from a dynamic array.
Extract_SI_Keys subroutine	Searches a Btree index for a specified value and returns a list of keys based on the search value.

[Extract_XML_Schema_Name function](#)	Extracts the name of an XML schema file, when passed the XML file.
[Field function](#)	Returns a substring from a string expression. The substring is specified by a delimiter character and which instance of the delimiter character is to be calculated.
[FieldCount function](#)	Same functionality as DCount() function.
[FieldStore function](#)	Use to replace, delete, or insert substrings in a string.
[Fix_LH subroutine](#)	Compresses overflow in a table; fixes Group Format Errors (GFEs); increments, decrements, resizes the sizelock for a table; defines a new threshold for a linear hash (OpenInsight) table.
[Flush statement](#)	Flushes the native tables cache. Use Flush with the GarbageCollect statement.
[Fmt function](#)	Formats data in a specified pattern. The pattern can include field width, background fill characters, line justification, conversion specifications, and masking.
[Footing statement](#)	Defines a page footer to be printed at the bottom of the page in conjunction with the Print statement. The footer text is printed after the maximum page height lines have been printed, and on Printer Off statement.
[For...Next statements](#)	Specifies the beginning and ending points of a For...Next loop.
[Format function](#)	Applies a specified format to data.
[Forward_Event subroutine](#)	Passes the control to the next event in the event chain.
[FreeCommon statement](#)	Clears the shared area of memory established by Common. The specified common block must exist when FreeCommon is called.
[FSMSG subroutine](#)	Displays an error message, typically after an OpenInsight I/O command such as Open, Read, or Write failed.
[FTPDELDIR Function](#)	Deletes a directory on an FTP server.
[FTPDELFILE Function](#)	Deletes a file on an FTP server.
[FTPGETFILE Function](#)	Retrieves a file from an FTP server.
[FTPLIST Function](#)	Retrieve a list of files from an FTP server.
[FTPMKDIR Function](#)	Creates a directory on an FTP server.
[FTPRENAMEFILE Function](#)	Renames a file from an FTP server.
[FTPSEND Function](#)	Sends a file to an FTP server.
[Function function](#)	Similar to the "call @" syntax for subroutines, the Function syntax allows a function to be called whose name is indeterminate at compile time.
[Function statement](#)	Identifies a program to the compiler as a function.
[GarbageCollect statement](#)	Recovers all unused or abandoned string space, from memory.
[GetBinaryValue Function](#)	Returns the value of a variable at a given offset.
[GetByteSize Function](#)	Returns the size in bytes of a given variable.
[GetCurrentDir subroutine](#)	Returns the current directory (the directory from which the application is running).
[Get.RecCount function](#)	Returns the number of rows in a native table.
[Get_App_Info function](#)	Retrieves information about the application parameter passed to the function. The information is stored in the SYSAPPS table.
[GET_CURRENT_EVENT Function](#)	Retrieves the current event.

[Get_DB_Users function](#)	Returns an @fm delimited list of users in the specified application.
[Get_Contract function](#)	Retrieves contract block information for a specified stored procedure.
[Get_Dialog_Info function](#)	Retrieves the information associated with a dialog box
[Get_Dialog_Params function](#)	Returns a block of parameters associated with this dialog box.
[Get_Env routine](#)	Returns an attribute or list of attributes from the database environment for the current database.
[Get_EventStatus function](#)	Retrieves the status of a forwarded event in an @FM-delimited array containing error codes.
[Get_Language Function](#)	Retrieves the currently active language used by OpenInsight.
[Get_LH_Info Function](#)	Retrieves Linear Hash Information for a specific LH table.
[Get_Printer function](#)	The Get_Printer function returns a printer property, based on the parameter passed to it.
[Get_Property function](#)	Returns the current value of the specified property of a control.
[Get_Repos_Entities function](#)	Returns a list of all entities classified under the specified application, type, and class.
[Get_SI_Values routine](#)	Returns all indexed values from the index for a column.
[Get.Sort.File function](#)	Returns a unique operating system filename, constructed using the workstation ID and a unique number.
[Get_Status function](#)	Returns error status information from a stored procedure. Use after calling any system stored procedure (SSP).
[Get_Sysinfo routine](#)	Retrieves critical system information, such as the OpenInsight serial number, the station ID, and the maximum number of users, from Open Engine.
[Get_XML_Value function](#)	Extracts tag values from records that have raw XML data stored in them.
[GetANSIToUnicode function](#)	Converts the ANSI numeric value to a Unicode numeric value using the current ANSI to Unicode map in effect.
[GetDefaultCharacter function](#)	Gets the default character value used when a Unicode character does not map to an ANSI character.
[GetEntity function](#)	Displays the OpenInsight common dialog as used by the User Interface and Client/Server Workspaces.
[GetLanguageName function](#)	Returns a character string (normally 3 letters) corresponding to the current Operating System regional settings. This string is the same as the extension on the language-specific resource files.
[GETMAIL function](#)	Socket function used to retrieve mail from a POP3 server.
[GetMenu function](#)	Return's the handle of a window's menu, if any. OpenInsight does not return the HANDLE property for a menu entity. Use GetMenu() instead.
[GetMinimumDelimiter function](#)	Gets the minimum system delimiter value used in converting ANSI characters to and from both Unicode and UTF-8 characters.
[GetNetworkType subroutine](#)	Returns the current network driver type of the OpenInsight System.
[GetNoOfDelimiters Function](#)	This function returns the number of characters to be used as System delimiters.
[GetPointer function](#)	Returns a pointer to the contents of a BASIC+ variable (cannot be expression).
[GetSessionCallStack Function](#)	Whenever a run-time error occurs, the call stack is recorded for that session. The GetSessionCallStack function will return the last recorded call stack for that session.
[GetUnicodeToANSI function](#)	Converts the Unicode numeric value to an ANSI numeric value using the current ANSI to Unicode map in effect.
[GetValue function](#)	De-references a pointer, and copies the data to a BASIC+ variable.

[GETWEBPAGE function](#)	Socket function used to retrieve the contents of a webpage.
[GoSub statement](#)	Provides the means for internal branching.
[GoTo statement](#)	Branches unconditionally to any label within the BASIC+ program.
[Heading statement](#)	Defines a page heading to be printed at the top of the page in conjunction with the Print statement.
[IConv function](#)	Converts data into the native internal OpenInsight format.
[IConv Boolean (B) function](#)	Converts characters or words into internal values of true (1) or false (0).
[IConv Date (D) function](#)	Converts a date string into its internal system format.
[IConv DateTime (DT) function](#)	Converts a date and time string into its internal system format.
[IConv ENCRYPT_FORMAT function](#)	Converts data strings into encrypted internal values.
[IConv Masked Decimal (MC, MD) function](#)	Converts decimal numbers into integers (internal system format).
[IConv Masked Scientific (MS) function](#)	Converts scientific notation into internal system format.
[IConv MX, HEX, MO, MB function](#)	Converts hexadecimal, octal, or binary formatted data into internal system format (decimal numbers). The value to convert can be as great as that represented with 32 bits.
[IConv Time (MT) function](#)	Converts times in external formats into internal system format. Internal system time is calculated as the number of seconds past midnight. A 24-hour day has 86,400 seconds.
[IConv Variable Binary (VB) function](#)	Allows the storage of binary data that might otherwise contain system delimiters.
[Index function](#)	Examines a string for the occurrence of a specified substring. Returns the starting position of the designated substring in the string.
[If...Then statements](#)	
[Indexc function](#)	Case-insensitive version of the Index() function. Examines a string for the occurrence of a specified substring.
[Index.Open subroutine](#)	The Index.Open subroutine is an alternative to the Open statement to open an OpenInsight table. In spite of its name, Index.Open can open any OpenInsight table.
[IndexLookup function](#)	The IndexLookup function is used by the "Index lookup" QuickEvent.
[Inet_Aborted function](#)	Called when an INET procedure has crashed and the Internet Gateway recovers.
[Inet_Finalize function](#)	Finalization of a response to a request.
[Inet_FormLoad function](#)	Internet procedure that retrieves an HTML form from the repository and fills it in with default form values.
[Inet_FormRead function](#)	Internet procedure that reads a row from native OpenInsight table(s) and merges the retrieved data with an HTML form.
[Inet_FormWrite function](#)	Internet procedure that retrieves data from an HTTP request and writes a row to native OpenInsight table(s).
[Inet_Msg function](#)	Converts a message from Msg() to HTML format.
[Inet_OI_XML function](#)	Exports XML data from an OpenInsight table.
[Inet_QueryParam function](#)	Allows you to get or set a specified item in an HTTP-request.
[Inet_Repos function](#)	Retrieves an HTML document from the repository.

[Inet_Rlist function](#)	Internet procedure that produces a List+ report in HTML format. The Report Builder tool (ORMAIN) is used to produce an HTML report.
[Inet_Security function](#)	The function that is called by OECGI.EXE before it calls any Internet procedure specified in the HTTP-request.
[Inet_Test_Connect function](#)	Internet procedure, executed from a Web browser, that tests whether OECGI/OECGI2 is running on a Web server.
[Inet_Trace function](#)	Internet procedure, executed from a Web browser, that tests whether OECGI.EXE is running on the Web server.
[InitDir statement](#)	Prepares for the DirList function, by establishing the default directory and file mask.
[InitRnd statement](#)	Initializes the random number generator, for use with the Rnd function.
[InList function](#)	Determines whether a specified value exists in a delimited list.
[InMat function](#)	Returns the number of elements read into a dimensioned array after a MatRead statement has successfully executed.
[Insert function](#)	Insert a field, value, or subvalue into a dynamic array.
[Int function](#)	Returns an integer numeric value, when calculating an expression.
[IsEventContext function](#)	Returns true if the code is executing in an event context.
[IsUnicode function](#)	Returns whether or not the Win32 Unicode APIs are being called.
[IsUTF8 function](#)	Returns whether or not the string data is interpreted as UTF-8 or ANSI characters.
[IsValidUser Function](#)	Determines whether the user is a valid user for the database.
[LCASE Function](#)	Converts the input parameter's contents from upper/mixed case to lower case.
[Len function](#)	Returns the number of characters in the specified string.
[List_Dict routine](#)	Returns the column definitions for a specified dictionary.
[List_Index routine](#)	Returns information about indexes for a specified table or for all tables.
[List_Index_Detailed routine](#)	Returns detailed information about indexes for a specified table or for all tables.
[List_Keys routine](#)	Returns a list of keys from a currently attached table.
[List_Tables routine](#)	Returns the attached tables in the current database.
[List_Users routine](#)	Returns information about the users in a specified database.
[List_Volume routine](#)	Returns information about the tables in a specified volume.
[List_Volume_Sub function](#)	Returns information about the tables in a specified volume.
[ListBasic routine](#)	Outputs a formatted listing of a BASIC+ program, either to the printer directly, or in Print Preview mode.
[Ln function](#)	Returns the natural (base e) logarithm of an expression.
[Locate statement](#)	Finds the position of a substring in a string, where the substring can be delimited by any ANSI character.
[Locate...By statement](#)	Use this version of Locate when you are dealing with sorted data.
[Lock routine](#)	Provides a method for coordinating access to tables, rows, or columns by setting locks
[Lock statement](#)	Sets locks to coordinate access to files and records on a network system.
[LockVariable statement](#)	Locks a BASIC+ variable, so that its data does not move in memory. Used in conjunction with GetPointer.
[Loop statement](#)	Marks the beginning of a group of statements, where the group is to be executed repeatedly, until the test expression is satisfied.
[Make.List subroutine](#)	The Make.List subroutine creates an active select list of keys from the passed list of keys.
[MAPIAddress function](#)	Displays a standard address list dialog box allowing users to create or modify a set of To:, CC: and BCC: recipients.

[MAPIDeleteMail function](#)	Deletes messages from MAPI message storage.
[MAPIDetails function](#)	Displays a standard dialog box containing the details of a given address list entry which may or may not be modified.
[MAPILogoff function](#)	Ends a session with the MAPI messaging system.
[MAPILogon function](#)	Begins a session with the MAPI messaging system.
[MAPIOpenMail function](#)	Returns the ids of MAPI messages of a given type.
[MAPIReadMail function](#)	Reads a MAPI message given its id.
[MAPISaveMail function](#)	Updates or creates a MAPI message.
[MAPISendDocuments function](#)	Displays a standard send message dialog box containing the given files as attachments.
[MAPISendMail function](#)	Sends a MAPI message.
[Mat statement](#)	Assigns a value to each element in a matrix. All elements can be assigned the same value. Or, values can be set equal to the corresponding values in a second matrix.
[Matches operator](#)	Compares a string value to a pre-defined pattern. Evaluates to 1 (true) or 0 (false).
[MatParse statement](#)	Assigns the value of each successive field of a dynamic array to successive elements in a matrix.
[MatRead statement](#)	Reads a dynamic array row into a matrix, assigning each field to consecutive elements in the matrix.
[MatUnparse function](#)	Assigns the value of each successive element in a matrix to successive elements in a dynamic array.
[MatWrite statement](#)	Writes the values of a matrix into a record.
[Max function](#)	Returns the maximum value of two passed arguments.
[MAX_OI_USERS Function](#)	This function will return the number of maximum users for a running copy of OpenInsight.
[Message_Box function](#)	Displays a Windows message in event context
[MFS Subroutine](#)	All MFSes should be coded in the SYSPROG application and stored in the SYSPROCS table.
[Min function](#)	Returns the minimum value of two passed arguments.
[MixedCase function](#)	Given a string, returns the string with initial capitalization (the first character of each word in upper case, the rest of the word in lower case).
[Mod function](#)	Returns the modulo (remainder) of two expressions.
[Msg function](#)	Displays dynamically constructed system messages.
[MVBFS Functions](#)	OpenInsight Record Locking for Multivalue Developers
[Name_Volume subroutine](#)	Initializes or renames a location (typically a folder). It assigns a label to the location and builds a location directory, creating a media map in the folder.
[Namecap function](#)	Given a string, returns the string with its proper capitalization.
[Neg function](#)	Returns the inverse of a specified number. This function is identical to the unary minus operator.
[NewEntity function](#)	Displays the OpenInsight common new dialog as used by the User Interface and Client/Server work spaces.
[Not function](#)	Returns the logical complement of an expression.
[Null statement](#)	Use anywhere a statement is required, but where you want nothing to happen.
[Num function](#)	Returns true (1) if the expression is a number or a numeric string. Otherwise, returns 0.
[OConv Boolean (B) function](#)	Converts a Boolean true (1) or false (0) string into any characters.
[OConv Date (D) function](#)	Converts a date, stored in internal system format, into a specified output format.

[OConv DateTime (DT) function](#)	Converts from internal storage format to a specified date and time format.
[OConv ENCRYPT_FORMAT function](#)	Converts encrypted values into data strings.
[OConv Character (MCx) function](#)	Converts a string into a specified output format.
[OConv Masked Decimal (MC, MD) function](#)	Converts an internal storage value to a decimal output format.
[OConv Masked Scientific (MS) function](#)	Converts from internal storage format to scientific notation.
[OConv MX, HEX, MO, MB function](#)	Converts from internal storage format to hex, octal, and binary formats.
[OConv Time (MT) function](#)	Converts a value from internal storage format to a time output format.
[OConv Variable Binary (VB) function](#)	Converts a stored IConv (VB) value back into its original form.
[OConv function](#)	Converts data from the internal system storage format to the specified format. OConv() is identical to Fmt().
[OIGETLOCALEINFO Function](#)	Retrieves Windows Language and Regional Settings from the workstation.
[OI_GET_LOCALE_DATE Function](#)	Returns the short date within the local format.
[OIWin_FormLoadPrev routine](#)	Loads the contents of the previously saved record into the current record.
[OleCallMethod Function](#)	Calls the MethodName method of the OLE object.
[OLECREATEINSTANCE Function](#)	Creates an instance of an OLE object
[OleGetProperty Function](#)	Returns the value of the specified property of an OLE object.
[OlePutProperty Subroutine](#)	Establishes the PropertyName property of the ole object.
[OleRequestLicKey Function](#)	Returns the runtime license key associated with the Classname class.
[OLESTATUS Function](#)	Returns the status of the last OLE operation
[OLE_GETWEBPAGE function](#)	An OLE function used to retrieve the contents of a webpage.
[On...GoSub statement](#)	Transfers program control to one of several internal subroutines that are listed as statement labels.
[On...GoTo statement](#)	Transfers program control to one of several statement labels. Branching is determined by computing an index expression.
[Open statement](#)	Creates a table access variable (a file handle).
[OSBRead statement](#)	Reads from an operating system file opened with OSOpen.
[OSBWrite statement](#)	Writes data into a previously opened operating system file, beginning at a specified point in the file. The file must have been opened using the OSOpen statement.
[OSClose statement](#)	Close an operating system file that has been opened with an OSOpen statement.
[OSDelete statement](#)	Deletes an operating system file.
[OSOpen Statement](#)	Opens an operating system file for use with OSBRead and OSBWrite statements.
[OSRead statement](#)	Reads an entire operating system file into a BASIC+ variable.

[OSWrite statement](#)	Writes data to the specified operating system file.
[Page statement](#)	Force printer to advance one page. This statement is supported for backward compatibility only; newer programs should use the Print_ functions.
[Parse_Struct subroutine](#)	Extracts elements from a structure into BASIC+ variables.
[Ping Function](#)	Sends a signal to an IP address.
[PlaceDialog() routine](#)	Places a window on the screen at specific positions. This routine should be used during the CREATE event of a window. This routine will also make the window visible.
[Popup function](#)	Displays tabular data and allows user selection.
[Pop.Select subroutine](#)	Restores a previously active select list, saved by the Push.Select subroutine, into Cursor 0.
[Pop.Session Subroutine](#)	Restores the environment parameters saved by a preceding call to Push.Session.
[Post_Event function](#)	Posts an event onto the event queue. The posted event will not be executed until the current event chain is completed. Compare to Send_Event.
[Print_End function](#)	Terminates the print job.
[Print_Init function](#)	Starts a printing session.
[Print_SetFont function](#)	Sets the font for the text to be printed.
[Print_SetFooter function](#)	Sets the page footer print and font.
[Print_SetHeader function](#)	Sets the page header print and font.
[Print_Text function](#)	Outputs a single line of text to the printer.
[Printer statement](#)	Directs output of Print statement to the printer.
[PrintLineFeed function](#)	Advances printing to a new line or allows a font change on the current line.
[PrintNewPage function](#)	Inserts a page break.
[PrintPageStatus function](#)	Returns the number of lines to the end of page.
[PrintRect function](#)	Prints a rectangle relative to the page, beginning at the top left and extending to the bottom right coordinate of the rectangle measured in inches.
[PrintSetBrush function](#)	Sets the style, hatch, and color of the brush as defined by the Windows SDK.
[PrintSetMargins function](#)	Sets the margins for the page to be printed for the printable area of the page.
[PrintSetPen function](#)	Sets the style, width, and color of the pen as defined by the Windows SDK.
[Push.Select subroutine](#)	Saves the currently active select list in Cursor 0 into four (4) variables. Cursor 0 is then available for program use. The select list can be restored using the Pop.Select subroutine.
[Push.Session Subroutine](#)	Saves the current environment parameters in the variables specified.
[PutBinaryValue Function](#)	This function will update the contents of a binary variable.
[Pwr function](#)	Calculates the value of an expression raised to the specified power.
[QryGetProperty function](#)	Retrieves a Query property.
[QryInstance function](#)	Creates a Query (statement handle) for a specified Connection Object.
[QryMethod function](#)	Executes a query method; a query corresponds to a statement handle for a Connection Object.

QrySetProperty function	Sets a Query property, returning TRUE on success and FALSE otherwise.
Quote function	Places double quotation marks around a given expression.
Read_Column routine	Returns one or more columns from a row in a table.
Read_Row routine	Returns one or more rows from a table.
Read statement	Reads a record from a native table, assigning the record to the specified variable.
ReadNext...By statement	Extracts the next key from a selected list of keys (a cursor), until the list is exhausted.
ReadNext statement	Extracts the next key from the list of keys, until the list is exhausted.
READONLY Property	A boolean property to set or retrieve the control's READONLY status.
ReadO statement	Use to read a record that you do not intend to modify.
ReadV statement	Reads a specific column into a BASIC+ variable.
Recompile_Proc subroutine	Recompiles one or more specified stored procedures.
Reduce subroutine	Creates a cursor using a filter or applies a filter to an existing cursor.
Relational Indexes	
Rem function	
Remove statement	Extracts substrings, including fields, values, and subvalues, from dynamic arrays, using ANSI characters 249 through 255 as delimiters.
Rename_Table subroutine	Changes the name of an attached table.
Replace function	Replaces a field, a value, or a subvalue, in a dynamic array.
Repository function	Use to issue methods to repository objects (entities). Repository entities are stored as rows in the SYSREPOS table.
RetStack function	Returns an @fm delimited list of the currently executing procedures as seen in the debugger's Call Stack window.
Return statement	Terminates an internal or external subroutine, and returns control back to the main or calling program.
Rev_Utility Function	The Rev_Utility function provides an API for various operating system functions.
RGB function	Returns the RGB (Red, Green, Blue) color value
RList routine	The processing module for OpenList queries RList takes the OpenList statement. The Report Designer uses RList to access data for queries.
Rnd function	Returns a random numeric integer. You can use the InitRnd statement to initialize the Rnd function.
RowExists function	Determines whether all rows specified in rowlist are in the specified native table.
RTI_CDOMail function	This function will send emails from OpenInsight using authenticated logins and/or a Secure Sockets Layer (SSL).
RTI_CreateGUID Function	This function will create a guaranteed unique identifier.
RTI_Crypto function	The RTI_Crypto provides for the ability to create encrypted values for authentication purposes.
RTI_DistinctList Function	Creates a distinct list of delimited values using passed values.
RTI_Expiration_Date function	Returns the expiration date of the OpenInsight license.
RTI_GetNetworkUserName function	Returns the current network name of the user.
RTI_JSON Function	The RTI_JSON function allows you to evaluate either a javascript function or a JSON (JavaScript Object Notation) expression.
RTI_List Function	The RTI_LIst function is a utility that helps in the use and management of saved lists in the SYSLISTS table.

[RTI_Load_Database Function](#)	Programmatically loads a DBT file.
[RTI_Lock_Owner function](#)	Returns the name of the computer containing the lock on a record.
[RTI_OS_Directory function](#)	This function provides methods for manipulating operating system directories.
[OERUN.EXE](#)	OERUN allows you to invoke an OpenInsight routine from the DOS command line.
[RTI_ParseCSV Function](#)	This function parses CSV files into OpenInsight delimited data.
[RTI_Remove_Style _Sheet Function](#)	This function will remove the style sheet from the window record.
[RTI_SendFax Function](#)	A function that allows for the sending of faxes from OpenInsight.
[RTI_Sets function](#)	IDX_SETS is a function that performs union, intersection and difference of delimited sets of row data.
[RTI_Stringbuilder function](#)	This function provides a utility to build and extract data from a string.
[RTI_AUTOEXEC](#)	RTI_AUTOEXEC is a support routine that manages the processing of startup tasks
[RTI_TASK_SCHEDULER](#)	RTI_TASK_SCHEDULER creates a Windows scheduler entry to launch your task at the desired time.
[RTI_TASK_SHUTDOWN](#)	RTI_TASK_SHUTDOWN stops the polling service that processes background tasks.
[RTI_TASK_STARTUP](#)	RTI_TASK_STARTUP starts a polling service to process background tasks.
[RTI_TASK_STATUS](#)	RTI_TASK_STATUS returns the current status of a submitted background tasks, and its results (if the task is complete).
[RTI_TASK_SUBMIT](#)	RTI_TASK_SUBMIT creates and optionally launches a background task. It is a member of the RTI_TASK_xxx programs. Use RTI_TASK_SUBMIT to trigger a background task from a program.
[RTI_TASKMANAGER](#)	RTI_TASKMANAGER is a support routine that manages the processing of background tasks, possibly creating additional OInsight.exe instances to perform those tasks.
[RTI_TRANSACTION subroutine](#)	RTI_TRANSACTION enables transaction processing (commit and rollback) for those filing systems that support it.
[RTP5 subroutine (BASIC+ Compiler)](#)	Calls the BASIC+ compiler. You can determine whether a program compiles correctly, and if not, to determine compilation errors.
[Run_Label subroutine](#)	Runs Report Builder+ labels.
[Run_BRW_Report subroutine](#)	Runs Banded Report Writer reports
[Run_Report subroutine](#)	Runs Report Builder reports.
[Save_Env routine](#)	Saves the current environment settings.
[Save_Select subroutine](#)	The Save_Select subroutine saves an active select list of keys from cursor 0.
[Select...By statement](#)	An extended version of the Select statement which supports multiple cursors and sort options. The Select...By statement results in a resolved (not latent) cursor.
[Select statement](#)	Creates a latent cursor of keys for an opened native table.
[Select.Seek subroutine](#)	Select.Seek is used to change the current position in a select list.
[Select_Into function](#)	This function accepts an Rlist statement and returns the results of the statement in a specified format.
[Send_Event function](#)	Executes an event (as compared to Post_Event(), which queues an event for execution).
[Send_Info subroutine](#)	The Send_Info subroutine sets the text for the system receiver. If no receiver has been set, the subroutine displays a line of text in the OpenInsight System Monitor.

[Send_Message function](#)	Requests a specific action from a form or control.
[SENDMAIL function](#)	Socket function used to send mail from an OpenInsight workstation.
[Seq function](#)	Returns the numeric value of an ASCII character (0-255).
[Serial function](#)	Returns the OpenInsight serial number.
[Set_Bgnd_IX_Time function](#)	Sets the number of seconds the engine waits before and between indexing. This function is used with Set_IDXSvr() to control dedicated indexing.
[Set_Dialog_Info function](#)	Sets the information associated with a dialog box.
[Set_Dialog_Params function](#)	Replaces any dialog-specific parameter data with new values.
[Set_Env routine](#)	Defines the value of an environment attribute or list of attributes for the current database.
[Set_EventStatus function](#)	Sets the status of an event forwarded from within an event handler.
[Set_FSError function](#)	Sets a filing system error to be retrieved by the caller.
[Set_IDXSvr function](#)	Toggles the dedicated indexing mode. To turn dedicated indexing on, pass 1; to turn dedicated indexing off, pass 0 (zero).
[Set_Language Function](#)	Sets the OpenInsight language.
[Set_MFS subroutine](#)	Programmatically attach Modifying Filing Systems (MFS) to specified tables.
[Set_Printer](#)	The Set_Printer function provides intelligent defaults for the parameters that are unassigned. The format of the parameters that are passed to the Set_Printer function are consistent with standard OI parameters.
[Set_Property function](#)	Sets properties in OpenInsight objects such as controls and windows.
[Set_Sort_File Function](#)	This function is used to set the path and file to be used by the sort mechanisms within OpenInsight when write temporary files.
[Set_Status function](#)	Sets or clears error information.
[SetANSIToUnicode subroutine](#)	Establishes the mapping between an ANSI numeric value and a Unicode numeric value.
[SetDebugger ("SPY") routine](#)	Enables or disables event logging on the System Monitor.
[SetDefaultCharacter subroutine](#)	Establishes the default character value used when a Unicode character does not map to an ANSI character.
[SetInitDirOptions subroutine](#)	Used to set attributes for the file mask used by the InitDir statement and DirList function.
[SetMinimumDelimiter subroutine](#)	Establishes the minimum system delimiter value.
[SetNoOfDelimiters routine](#)	This routine sets the number of characters downward from 255 that should be treated as System Delimiters.
[SetUTF8 subroutine](#)	Establishes whether or not string data is interpreted as UTF-8 or ANSI.
[Sin function](#)	Calculates the sine of an angle.
[SMTPMAIL function](#)	Socket function used to send SMTP mail from an OpenInsight workstation. Allows for the use of Multi-part MIME when sending email.
[Space function](#)	Generates a string value containing a specified number of blank spaces. Use Space to provide a particular number of spaces between any two strings.
[SQRT function](#)	Calculates the square root of a positive number.
[Start_MDIChild function](#)	Executes MDI child windows.
[Start_Window function](#)	Begins execution of an OpenInsight window (OIWINEXE). Optionally, returns an existing structure for modification before execution.

[Status function](#)	Returns a value that has been set by certain previous operations. Some procedures, both system and user-written, use Status() to set an error or result code.
[Str function](#)	Returns a string containing the expression repeated the specified number of times.
[Struct_Flush subroutine](#)	Flushes any cached structure definitions.
[Struct_Len function](#)	Returns the length in bytes of a structure based on the structure's definition.
[Struct_To_Var function](#)	Converts a binary structure to a dynamic array, placing each element from the structure into the corresponding field.
[Subroutine statement](#)	Identifies the source code that follows as a subroutine. Use in the first line of code.
[Sum function](#)	In a dynamic array, adds numeric data at the lowest delimiter level to numeric data at the next higher level.
[Swap statement](#)	Within a variable, replaces all occurrences of one string with another.
[TableExists function](#)	Determines whether the table specified exists in the current database.
[Tan function](#)	Finds the tangent of an angle, expressed in degrees.
[Time function](#)	Returns the time of day, in internal storage format.
[TimeDate function](#)	Returns a string expressing the current time and date in external (display) format.
[Transact subroutine](#)	Controls transaction processing.
[Transfer statement](#)	Moves the contents of one variable into another variable, and clears the original variable.
[Trim, TrimB, TrimF functions](#)	Deletes extra blank spaces from a character string or variable.
[U2 Functions in OI](#)	The following functions are used in OpenInsight to communicate with U2.
[UCASE Function](#)	Converts the input parameter's contents from lower/mixed case to upper case.
[Unassigned function](#)	Tests if a variable has been assigned. Returns True (1) if the variable is unassigned and False (0) if the variable is assigned - the opposite result from the assigned() function.
[UNICODE_ANSI Function](#)	Converts a UNICODE string to an ANSI string.
[UNICODE_UTF8 Function](#)	Converts a UNICODE string to an UTF8 string.
[UnLock statement](#)	Releases a lock set by the station in a previous Lock statement. A table or row can only be locked again (by another user, for example) after the UnLock statement has been issued.
[Unlock subroutine](#)	Releases locked records.
[UnlockVariable statement](#)	Unlocks a BASIC+ variable that has been locked with LockVariable.
[Update_Index subroutine](#)	Updates or rebuilds indexes in a specified column or for all columns in an attached table.
[UTF8_ANSI function](#)	Converts a variable containing UTF_8 characters to a variable containing the corresponding ANSI characters.
[UTF8_Unicode Function](#)	Converts an UTF8 string to Unicode string.
[Utility function](#)	The Utility function is the OpenInsight interface to the Windows API, which provides a series of services that can be used to perform various tasks.
[V119 (String Sort) subroutine](#)	V119 is an Assembly Language commuter module that supports eight different types of operations depending on the first parameter passed.
[Validate routine](#)	Validates and converts data passed to the engine based on a specified validation pattern.
[Var_To_Struct function](#)	Converts an @fm-delimited variable into a structure placing, obtaining each structure element from the corresponding field in the variable.
[Verify_LH stored procedure subroutine](#)	Tests linear hash tables, and stores statistical information about data distribution and space utilization for a table or tables. Verify_LH also diagnoses group format errors (GFEs).

Write statement	Inserts or updates a row in the table. If the row does not exist, a new one will be created in the table. It the row does exist, it will be overwritten.
Write_Column routine	Writes one or more columns to a specified row in a table.
Write_Row routine	Writes one row to a specified table.
WriteV statement	Updates the data content of a specified column in a row in a table.
Xlate function	Returns data from the specified element in a table that has not been opened. Typically used in symbolic fields to bring back column data from a related table.
XML_Importer function	Imports XML data into an OpenInsight table, optionally using an XML template.
XOGetProperty function	Retrieves a Connection Object property; the Connection Object is specified by the passed handle.
XOInstance function	Creates a Connection Object instance and returns the handle.
XOMethod function	Executes a Connection Object method; the Connection Object is specified by the passed handle.
XOSetProperty function	Sets a Connection Object property; the Connection Object is specified by the passed handle.
XREF subroutine	Reviews a string and divides it into "words," the boundaries of which are determined by delimiters you specify.
Yield subroutine	Checks for pending events in the Windows event queue and executes them. Returns control as soon as the event queue is empty.

Search this documentation

Recently Updated Pages