BAR Terminology

Prepared by Christopher Allen 2008

Document #: MADFISH-0003
Revision: DRAFT

Deleted:
Deleted:

Signatures

Author _____ Date _____

Read and Understood _____ Date _____

BAR Terminology

This document defines terms used to characterize the BAR protocol, and applies to all controlled documents that reference BAR terminology. The regular expression format used in such controlled documents is listed at the end of the document.

General Terms

Term	Definition
absolute position	An index or offset whose base (0 or 1) is at the beginning of the memory map. Absolute position in RAM refers to the address relative to the first addressable memory location in RAM; absolute position in a file refers to the address relative to the first byte of the file.
address	Term used to specify the physical location of a byte or word in RAM. An address is a key component to the functionality of pointers and references to objects.
algorithm	A finite list of well-defined instructions used to complete a task.
alignment boundary	Term that describes the multiple of bytes or bits where a particular type can start in memory. A 2-byte boundary means that every unit of data starts at an offset of 0 bytes, 2 bytes, 4 bytes, etc. A 4-bit boundary means that every unit of data starts at an offset of 0 bits, 4 bits, 8 bits, etc.
application	A program that, when run, accomplishes a complex task. An application generally has significant user-interface components.
argument	A value passed to a function during a function call.
array	A series of data units (elements). Normally the elements of an array are co-located in ascending order. In most programming languages, an array has a fixed number of elements.
Array of Complex Type	BAR terminology for an array with a fixed number of elements, with elements not of Simple Type.
Array of Simple Type	BAR terminology for an array with a fixed number of elements, with elements of Simple Type.
attribute (BAR construct)	A term for a flag that describes a BAR construct. An attribute is either present or absent for a particular BAR construct.
attribute (general)	A feature or property of an object.
automatic termination	A Node List feature that causes the number of list iterations to be determined by discarding the first invalid (partial) list iteration, in favor of list iterations already deemed valid. Compare "manual termination."
autoterminate	An attribute of a construct in an implementation file that identifies the construct as having automatic termination.
backward tweak	BAR terminology for manipulation of the parse cursor in the reverse (negative) direction.
backward-chain	A list reference linked to the previous object in a sequence.
bandwidth	A networking term for how much data can be transferred over a connection in a specific amount of time.
BAR (interfacial object)	See "BAR object."
BAR (protocol)	Binary Artifact Reference. See "BAR protocol."
BAR compiler	Software that compiles the text rendition of a BAR implementation file into a binary or mixed rendition.

Formatted

Deleted: linked

Term	Definition
BAR engine	The software package that acts as the interface to BAR services and conducts all BAR functionality. There are many different possible BAR engines, but there is only one BAR protocol.
BAR functionality	Term that refers to what BAR "does" in the usage of the BAR services.
BAR IFS	See "BAR Implementation File Specification."
BAR implementation	Term that refers to the means the BAR services are implemented in a particular platform, or the "way it's done."
BAR implementation file	A self-contained file that contains a file format specification and maps the document object model for the specification, for use with the BAR services.
BAR implementation file component	An element of a BAR implementation file that controls the functionality and identity of the implementation file. Most components have names and unique identifiers.
BAR Implementation File Specification	The specification that describes the required format and functionality of BAR implementation files and BAR registry files.
BAR interface	The interfacial objects, functions, variables, and types that permit the client programmer to use the BAR services.
BAR object	An interfacial object to an implementation file. A BAR object is capable of interfacing with exactly one implementation file.
BAR protocol	A platform-independent standard for mapping document object models of binary artifacts (files), facilitating deserialization and serialization operations for such files, and facilitating the exploration of file formats. The BAR protocol aims to be the universal document object model for all binary files.
BAR registry file	A file that stores a collection of compiled BAR implementation files.
BAR services	The BAR protocol capabilities provided to the client programmer.
BAR_Registry (interfacial object)	An interfacial object to a BAR registry file. A BAR_Registry object is capable of interfacing with exactly one BAR registry file.
BAR_Services (interfacial object)	A top-level interfacial object to the BAR services. This object must be fully functional before any child BAR_Registry or BAR objects can be created.
Base64 encoding	A type of encoding that characterizes a grouping of six bits as one printable character, with capital English letters (A-Z), lower case English letters (a-z), Arabic numerals (0-9), and two other characters composing the representative characters. This type of encoding is sometimes used to pack binary data when only text data can be transmitted over a specific medium.
big endian	Used to describe "big end first" for byte or bit order: the most significant portion of a word is stored first, while the least significant portion of a word is stored last. Compare "little endian."
binary	Consisting of two values. The adjective is normally used to describe data for which any bit within the data can be potentially any value. Compare "text."
Binary Artifact Reference	See "BAR protocol."
binary deserialization	The process of deserializing data from a binary format.
binary file	A file composed of a series of bytes to be interpreted verbatim. A binary file is generally much less readable than a text file due to the fact that the bytes can represent any value (not just text characters) and do not have obvious meaning to a human unfamiliar with the specific binary file format. Compare "text file."
binary serialization	The process of serializing data into a binary format.
bit	A unit of memory representing a single Boolean value. A bit can have a value of either zero or one.
bit fiddling	Used to describe population of bits to, or extraction of bits from, a byte or word. Bit fiddling usually involves bitwise operations such as AND, OR, and XOR, as well as left-shift and right-shift operations.

Formatted

Deleted:

Term	Definition
bit field	A sequence of a fixed number of bits. The number of bits is not necessarily a multiple of 8.
Bit Field Structure	A BAR specific construct that represents a fixed-length portion of a file, composed of a series of named bit fields. The bit fields of the data structure are consistently aligned relative to each other, although they do not need to be aligned on byte boundaries.
bit length	See "bit size."
bit position offset from parse cursor	A value, from 0 to 7, that refers to the bit position within a byte where data start at the parse cursor.
bit scan	A series of bit fields packed in such a manner that the fields are not aligned on consistent boundaries, and do not need to be aligned on byte boundaries.
Bit Scan Block	A BAR specific construct that is a subtype of Organized Block. A Bit Scan Block represents a variable-length portion of a file, composed of a bit scan. This type of block restricts its children to the types Bit Scan Block and Bit Field Structure.
bit size	A size of a chunk of memory, measured in the number of bits composing the memory.
bit stream	See "bit scan."
bitposition	A built-in global variable that constitutes the bit position offset from parse cursor.
block	A BAR general construct that represents a variable-length portion of a file.
block scope	During compilation, this is the prevailing scope just inside of a block definition's braces.
block size limit	A momentary upper limit to block size during deserialization
BlockSize	A method called during deserialization that calculates the byte size or bit size of a construct.
bookmark	A reference to a specific location in a dataset. Also see "node bookmark."
Boolean	Referring to a type with two possible values: true (or one) and false (or zero).
bounds checking	A process that compares a memory address from a calculation against upper and/or lower limits before attempting to use the address.
buffer	A quantity of system memory, generally unstructured, used to store data.
byte	A unit of memory composed of 8 bits.
byte length	See "byte size."
byte ordering	The ordering of bytes that compose a larger word. See "big endian" and "little endian."
byte size	A size of a chunk of memory, measured in the number of bytes composing the memory.
C	A general-purpose procedural programming language developed by Dennis Ritchie at Bell Labs. This language is intended to run very low-level operations, and its functionality is designed to have little dependence on target machine or operating system. C does not offer built-in support for object-oriented programming, although object orientation can be "simulated" with pointers.
C standard library	A library for C and C++ that offers general, low-level functions that bear similarities to low-level functions found in most other programming languages. The functions in such a library perform very basic tasks such as string formatting, input/output processing, and floating-point arithmetic.
C#	An object-oriented programming language developed by Microsoft Corporation that has many similarities to C and C++ programming languages. C# is a preferred language to use for development for the .NET platform. Unlike C and C++, C# aims to introduce type and pointer safety, as well as removal of global data and functions.
C++	A general-purpose procedural programming language Bjarne Stroustrup and Bell Labs developed to serve as the next-generation language to C. While C++ is designed to do anything that C can do, C++ also offers significant built-in support for object-oriented programming.

Formatted

Deleted: s

Deleted:

Deleted: n

Deleted: developed by

Term	Definition
call	Term used to describe an instruction pointer's temporary jump to another memory location (the beginning of a function), while passing possible arguments for use by the function. The function then "returns" from the call after the call is over, moving the instruction pointer back to the memory location just after the location where the function call originally took place.
calling function	Term used to describe the function that called the function where the instruction pointer currently resides.
calling program	Term used to describe the program that called the function where the instruction pointer currently resides.
case sensitivity	Describes whether upper case vs. lower case matters in text data. For example, a case-sensitive context distinguishes between the characters "A" and "a," while a case-insensitive context treats "A" and "a" as functionally identical.
cast	See "typecast."
character	A single unit of text. In many contexts, character refers to a single byte in size. In BAR documentation, a character refers to a unit of text that could be any number of bytes in size.
character size	See "character width."
character width	The number of bytes that compose a single character. ASCII characters are one byte in size; Unicode characters can be one or more bytes in size.
child	A conceptual object in a hierarchy, such that a more general "parent" object owns the child. Child objects often contain references to the memory location of the parent.
class	A data structure which qualifies as an object in object-oriented programming. A class can have data members and/or member functions.
client	The program or process that makes use of BAR services.
client programmer	The person who designs the program or process that makes use of BAR services.
code	Statements that compose either symbolic or actual machine instructions for a computer to execute.
co-located	Having adjacent memory addresses, such that data structures are in a sequence, in a specific order, with no gaps in memory that contain data of any other type.
compile	The act of converting data of one form into data of another form using a complex series of operations. For computers, compilation normally means converting statements in text to object-coded instructions.
compiler	A software application that compiles a file or combination of files.
Complex Structure	BAR terminology for a data structure with a fixed number of named elements, with elements not necessarily of Simple Type.
compression	Term for the compaction of data from their normal size to a smaller size.
constant	Refers to a value that does not change. In programming, this can also be used as a noun to refer to a variable whose value is not allowed to change.
construct	A component of a BAR implementation file that composes part of a binary file's document object model.
construct definition	A series of statements in a BAR implementation file that define the functionality of a construct.
construct scope	During compilation, this is the prevailing scope just inside the braces of a block definition or data structure definition.
constructor	A function called automatically upon the creation of an object. A constructor is used to set up resources and capabilities as a "de-facto" part of the object's initialization.
continuation condition	BAR terminology for the go-ahead condition for continuation of repeating list iterations. A false return value from a Termination function iindicates this.

Formatted

Deleted: where

Deleted: where

Deleted: s

Deleted: its

Deleted: This is indicated by

Deleted: a

Deleted: .

Term	Definition
control flag	A macro definition defined in a BAR implementation file that influences the functionality of the implementation file.
control variable	A global variable defined in a BAR implementation file that influences the functionality of the implementation file.
copying	In programming, "copying" refers to the reading from a specific portion of system memory or secondary storage memory, and writing an exact copy of the memory to another location within system memory or secondary storage memory.
core functionality	The principal capabilities of BAR, including file deserialization, file serialization, format conversion services, and information lookup and retrieval.
count	In programming, a count is an integer value that indicates a certain number of iterations, such as in a loop. A count can contain the number of elapsed iterations (upward counting) or the number of iterations remaining (downward counting).
counter	A variable that contains a count.
CPU	Central Processing Unit.
creation	In programming, "creation" refers to the allocation or reservation of a portion of system memory or secondary storage memory for a specific purpose. A "created" object is merely an abstract concept applied to existing computer hardware.
critical error	BAR terminology for an error that is so serious that serialization or deserialization operations cannot continue. Serialization and deserialization operations are allowed to start over from the beginning, but not to continue, after a critical error occurs.
critical step	A specific method called, if it exists, as a formal part of deserialization and serialization procedures.
current node	The node pointed to by the node cursor. Many BAR services operate on the current node.
cycle	Computer term for an iteration in a loop, especially in the execution of instructions.
data	A quantity of system or secondary storage memory.
data member	A member of a data structure composed of a variable.
data structure	In BAR terminology, this is a BAR general construct that represents a fixed-length portion of a file. In general terminology, this is any fixed-length quantity of data that has individual data members as children.
data structure scope	During compilation, this is the prevailing scope just inside a data structure definition's braces.
database	An organized collection of data, usually found in secondary storage, designed for service-based read and write access to portions of the data. A database is often a single file, but it can span multiple files.
decimal	A base-10 numeric system based on 10 Arabic numerals (0-9).
Decision List	A BAR specific construct that represents a dynamically sized portion of a file. A Decision List exists only to determine a specific order in which blocks and data structures shall be deserialized. A Decision List does not persist as a node after deserialization.
declaration	The entirety of a definition of types, data, data structures, functions, or other components, plus its identification as a declaration, in text.
definition	The lines of text that define types, data, data structures, functions, or other components.
dereference	The act of taking a reference (usually an address to a memory location) and accessing the memory at this reference. This access can be a read operation, a write operation, or both.
deserialization	The concept of taking a sequence of data in a serialized form and constructing deserialized nodes, which facilitate easier extraction and population of data.

Formatted

Deleted: n

Term	Definition
Deserialize	A method called during deserialization that replaces a node with an entirely different substitute node, which does not need to be the same size or type as the original node. The function return value is a pointer to the output buffer to the data in deserialized form.
destabilization	Any characteristic that reduces or prevents normal operability, such as the ongoing occurrence of resource or memory leaks and/or continually degraded system performance.
destroy	The act of removal of an object from conceptual representation. In programming, this normally amounts to freeing of the memory the object had been using and removal/invalidation of references to the object.
destructor	A function called automatically upon the destruction of an object. A destructor is used to clean up resources set up by the object.
determinism	Describes a process, state, or algorithm that delivers consistent inputs and outputs regardless of independent variables and context of implementation. Deterministic behavior is either self-evident or provable via testing and analysis.
device driver	A component of an operating system that operates on a low level (especially computer hardware level) and provides services to applications and to the operating system itself.
discardable	An attribute of a construct in an implementation file that identifies the construct as a "pause point" for incremental serialization and incremental deserialization.
disk	A secondary storage device or other equivalent device.
disk space	Physical memory mapped to a secondary storage device (mainly, a disk).
distributed object	An interfacial object whose access is immediately available to the client.
document	Equivalent to "file." A document is more associated with containing text than binary data.
document object model	The formal representation of "what's in a document of a particular type." With a complete document object model, an application can readily access all portions of a file because the format is known ahead of time.
driver	See "device driver."
DTD	Document Type Definition. A DTD file is one way to define the format of an XML document.
dynamic	Changing or non-static. "Dynamic" in computer terms usually means a memory location of a dynamic object will not be the same every time.
dynamic library	A library that is loaded and linked at run-time with an application or operating system.
dynamic sizing	The principle of sizing an object "on the fly" with no predefined, fixed size.
EBNF	Extended Backus-Naur Form. This is a syntactic metalanguage specification used to characterize expression formats.
embedded compiler	A compiler embedded within the body of another application or platform. For example, the BAR services engine would need to contain an embedded BAR compiler in order to load text-based implementation files.
embedded file	A "file within a file." For example, many BAR implementation files are embedded within a BAR registry file.
empirical	Meaning types are derived from clearly traceable origins.
emulation	The concept of performing functionality or interpreting a data format, in a literal fashion, on a system not designed to process such functionality or formats.
encryption	The concept of data translation with the intent of obfuscating the appearance or purpose of the data.
enumerated constant	An enumerated constant.
enumerated constant block	A definition of several enumerated constants in sequence.
environment	See "hosting environment."

Formatted

Term	Definition
error	A computer state that is contrary to intended functionality.
error code	A number that identifies a type or category of error.
exception	A special type of error associated with an invalid processor state. Exceptions tend to represent low-level hardware errors, such as division by zero or out-of-bounds memory access.
exception handling	The means of identifying exceptions and running specific routines in the event of an exception.
execute	The act of literal interpretation of instructions and carrying out of the tasks associated with their meaning.
exit condition	A condition that will cause the flow of instructions to exit from (move outside of) a function, loop, or process.
explicit sizing	Sizing via exact numeric specification, especially in bytes or bits. See also "implicit sizing" and "termination sizing."
extensible	Adjective describing something that facilitates additional future capabilities.
extraction	The act of pulling data from a data source.
field	A sub-component of a portion of data.
file	Bytes containing data, whose principal storage location is secondary storage.
file access	File access can include read, write, or read/write access. Access rights limit what can be done on a file or directory.
file close	An operation involving shutting down the read, write, or read/write access. This operation is most likely to "guarantee" a write to secondary storage.
file extension	A trailing sequence of characters at the end of a filename, appearing after a period. A file extension tends to identify the type of information in a file.
file format specification	A document describing the format of a file. This type of document is conceptual; it does not necessarily describe the document object model.
file format version	A particular rendition of a file format of a particular specification. Version differences between file formats can profoundly impact the document object model.
file mapping	The process of associating portions of a file with BAR components, as in the creation of nodes.
file modes of operation	Types of file access. These include reading and writing. File access restrictions may prevent one or both modes from being used.
file offset	An absolute position into a file (generally a zero-based byte position).
file open	An operation involving preparing a file in secondary storage for reading, writing, or both.
file read	An operation involving reading data from an open file.
file scope	During compilation, this is the prevailing scope when not inside any braces.
file size	The size of a file (generally in bytes). The largest absolute position of a file is generally this value minus 1.
file write	An operation involving writing data to an open file.
filename (file name)	A string identifying a file.
finalization	The act of finishing use.
fixed-length	A portion of data whose size is consistent. Compare "variable-length."
flag	A value (generally Boolean) that signals a condition. Flags with related functionality tend to be grouped together in a datum word.
floating-point	Describes a number that has a certain extent of precision, but no fixed decimal point. Such a number has a sign, a mantissa, and an exponent.
floating-point arithmetic	Mathematical operations performed on floating-point numbers.
forward tweak	BAR terminology for manipulation of the parse cursor in the forward (positive) direction.

Formatted

Deleted: ,

Deleted: a

Term	Definition
forward-chain	A linked list reference to the next object in a sequence.
fragmented	Not contiguous in memory.
function	A series of programming statements that compose a routine with possible inputs and a return value.
function definition (BAR implementation file)	The statements composing a function body.
function scope	During compilation, this is the prevailing scope just inside of a function definition's braces.
general construct	A construct in a BAR implementation file with no further breakdown in type. Includes blocks, lists, and data structures.
generation	See "creation."
global function	A function that can be called from any context.
global scope	During compilation, this is the prevailing scope when no other scope prevails. Any functions or variables declared at file scope without an explicit namespace indicating otherwise are considered to be at global scope.
global variable	A variable whose value is "visible" at any scope in a BAR implementation file.
handle	An object representing the identification for a particular resource. A handle is usually integral in nature, such as an address of resource information or an index into an array of resource information.
hardware	A device that physically exists (tangible).
header structure	A data structure that appears as the first child of an Organized Block, and is fundamentally tied to the existence of the block. If an Organized Block has a header structure, a header structure is created and initialized with default values whenever the Organized Block is created, and it is removed when the Organized Block is removed. It is never possible to remove a header structure while keeping its parent block intact.
heuristic	The steps and logic used in computations.
hexadecimal	A base-16 numeric system based on the 10 Arabic numerals (0-9) and 6 Roman characters (A-F).
hierarchy	A graded or ranked series.
hierarchy level	A number that characterizes hierarchy among linked nodes. Zero is reserved for the top-level node. One number is added to hierarchy level for each child of a parent.
high-level	This term refers to general concepts and large, complex sets of data. Compare “low-level.”
hosting environment	A combination of the target hardware, the operating system, and the programming language used to support the BAR protocol.
I.F.	(BAR) implementation file.
IEEE	Institute of Electrical and Electronics Engineers.
IEEE 754 floating-point	A floating point number (4-byte single precision or 8-byte double precision) whose format is controlled by the IEEE.
implementation	The actual manifestation of a particular algorithm or process; the "way it's done."
implementation chaos	The fundamental inability to standardize knowledge and processes, with the consequence of generating a large number of disparate implementations to support a proportionally smaller number of formats.
implementation file	See "BAR implementation file."
implicit sizing	Sizing via subtraction of the sizes of all nodes up to, but not including, the node to be sized, from the last theoretical point of the parent of the node to be sized. See also "explicit sizing" and "termination sizing."
incremental deserialization	A deserialization process in which the client repeatedly invokes “resume deserialization” steps, which are interspersed with the client’s own data processing.

Formatted

Deleted: being generated

Term	Definition
incremental loading	See “incremental deserialization.”
incremental saving	See “incremental serialization.”
incremental serialization	A serialization process in which the client repeatedly invokes “resume serialization” steps, which are interspersed with the client’s own data processing.
index (offset to array or pointer)	A number used to identify a specific element in a set of data. Generally, indices are zero-based.
index (registry)	A zero-based number used to identify a specific BAR implementation file embedded within a registry.
infinite cycle	A condition in which a procedure executes a set of steps endlessly; this is usually an error.
initial value	The value of data whenthey are first used.
Initialization	A method called during deserialization that prepares an object for future calls to methods used in deserialization.
initialization	The act of preparing for use.
initializer	A value or set of values that serve as the initial quantity of data upon initialization of the data.
instance	A portion of memory that composes an object of a particular type.
instruction	A combination of bits or bytes that constitute a low-level task for a computer. These bits or bytes are often called "opcodes."
instruction pointer	A reference to a memory location where the next object-coded or machine-coded instruction shall be interpreted and executed. An instruction pointer generally moves forward in memory for each instruction it processes, although some instructions can move this pointer in a nonlinear fashion.
integer	A positive, negative, or zero number that has no fractional component.
integer arithmetic	Mathematical operations performed on integers. Integer arithmetic differs from floating-point arithmetic in several important ways: two's complement integers can automatically wrap around during addition and subtraction operations (which is not considered an error), multiplication and division operations have separate signed and unsigned variations, multiplication overflows are not considered errors, and the fractional portion of a quotient after division is discarded. There are many operations that can be performed on integers that cannot be performed on floating-point numbers, such as bitwise, modular, and shift operations.
internal function	A function that can be called only within a confined scope (i.e. not an interfacial function).
internal object	An object used within a confined scope (i.e. not an interfacial object).
invoke	Call a function or execute a procedure.
iteration	Within a sequence, an iteration constitutes one unit. In BAR, means list iteration.
Java	An object-oriented programming language developed by Sun Microsystems that has many similarities to C and C++ programming languages. Java is intended to run on a wide variety of platforms and deliver consistent results regardless of platform. Unlike C and C++, Java aims to introduce type and pointer safety, as well as removal of global data and functions.
keyword	A combination of characters (usually alphanumeric) that identifies a built-in symbol in a programming language.
last iteration condition	BAR terminology for the condition for stopping repetition of list iterations. This is indicated by a true return value from a Termination function.
least-significant bit	A bit of a word that equates to the lowest binary value. In an integer, the least significant bit indicates a value of 0 or 1.
least-significant byte	A byte of a multi-byte word that equates to the lowest binary value. In a 16-bit unsigned integer, the least significant byte indicates values anywhere from 0 (all bits 0) to 255 (all

Formatted

Deleted: it is

Deleted:

Term	Definition
	bits 1).
legacy	Refers to dated and/or disused formats.
length	See "size."
library	A bank of compiled data and functions. A library is an intermediate file that can be used to create executable code in a program.
linear file organization	A file whose specification biases reading in a forward-only direction. Most files are linearly organized.
link	The process of taking object-coded data from a compiled program or library and assembling them into native CPU opcodes.
linked list	A sequence of objects with pointers or references to each other. Generally, objects in a linked list are not co-located in memory per the order of the list.
list	An ordered set of items. In BAR, a list is a general construct containing definitions of dynamically-sized portions of a file, which controls the selection and repetition of blocks and data structures.
list item	A reference to a construct in the body of a list.
list iteration	A set of nodes that, during deserialization, compose a single deserialized instance of a set of node list members.
list scope	During compilation, this is the prevailing scope just inside of a list definition's braces.
little endian	Used to describe "little end first" for byte or bit order; the least significant portion of a word is stored first, while the most significant portion of a word is stored last. Compare "big endian."
loading	Synonymous with file reading.
locking	Restricting access to a resource, especially a synchronization resource.
long integer (BAR implementation file)	A 32-bit signed integer.
low-level	This term refers to simple concepts and small, empirical data types. Compare “high-level.”
macro definition	A definition in an implementation file that replaces one token with another token, making one token the equivalent of another.
manual termination	A node list feature that causes the number of list iterations to be determined by a directly-set termination condition while deserializing the last list iteration. Compare "automatic termination."
maxblocksize	A built-in global variable that identifies the maximum allowable block size that a node can take up.
member	A sub-unit of an object.
member function	A function associated with the scope of a particular object. A member function has as an implicit parameter the pointer to an object instance, which can be used, directly or indirectly, to reference and manipulate the object instance.
member scope	During compilation of a member function body, this is the prevailing scope if parameter scope does not prevail.
memory	A form of hardware that stores data for later usage. Computer memory generally stores information in the form of a series of bytes.
memory allocation	The act of reserving memory from a free portion of memory, deducting from the overall supply of free memory.
memory deallocation	The act of relinquishing previously allocated memory, restoring the portion to free status, and adding back to the overall supply of free memory.
memory file	A file whose "secondary storage" status is conceptual only. A memory file can exist partially or entirely within RAM.

Formatted

Deleted: it

Deleted: s

Deleted:

Deleted: .

Term	Definition
memory freeing	See "memory deallocation."
memory leak	A failure to deallocate a portion of memory when finished using the memory. Memory leaks often do not cause problems immediately, but as they can accumulate over time, they can cause gradual destabilization of a system.
method	See "member function."
minblocksize	A built-in global variable that identifies the minimum allowable block size that a node can take.
mixed text and binary file	An implementation file composed of a binary portion and followed by a text portion.
mod bytes	Bytes that influence the actions of opcodes.
most-significant bit	A bit of a word that equates to the highest binary value. In a 16-bit unsigned integer, the most significant bit indicates a value of 0 or 32768.
most-significant byte	A byte of a multi-byte word that equates to the highest binary value. In a 16-bit unsigned integer, the most significant byte indicates values anywhere from 0 (all bits 0) to 65280 (all bits 1).
multithreaded	Refers to an application or process composed of multitasked threads that run concurrently. Compare "single-threaded."
name	Characters (generally alphanumeric) that identify an object, type, or other concept.
native	Refers to capabilities that a platform naturally supports without requiring extra effort to implement.
network	A communication system between individual computers.
network packet	A unit of data sent over a network.
node	A fundamental discrete location in a hierarchical linked list which potentially has a parent, a previous sibling, a next sibling, and child nodes. In a deserialized file in BAR, a node represents an absolute location in the file.
node bookmark	A saved location of a node, to which the client programmer can refer at a later time.
node cursor	A reference to a particular node maintained by a BAR interfacial object.
node handle	A compilation of either an entire node or a thoroughly descriptive subset of the node components that uniquely identifies the node.
Node List	A BAR specific construct that defines a list of one or more constructs to compose a block's body. A Node List does not persist as a node following deserialization.
node navigation	The process of moving the node cursor from one node to another.
nonlinear file organization	A file whose specification contains data that are designed to be read in a non-forward-only order.
nonrepeating list	A Node List whose contents are designed to have only one iteration.
null	A state indicating zero or empty. A null byte has a value of zero; a null pointer or reference points to nowhere; a null object is an invalid object. Many types have a single dedicated "null" value.
object	A high-level conceptual representation of any sort of unique instance. An object as used in programming often refers to specialized data in memory resembling a particular type.
object-coded	A type of encoding for instructions that is binary in format, but more symbolic than machine-specific. Object-coded instructions are not assembly language instructions.
object-oriented	Referring to a programming language or protocol that directly facilitates the creation, maintenance, and destruction of objects. An object-oriented language facilitates the use of member functions for most, if not all, objects.
offset	A zero-based numerical position from a certain point of reference.

Formatted

Deleted: up

Deleted:

Deleted: is

Term	Definition
offset tweaking	The act of modifying a parse cursor used to serialize or deserialize data in a file in a nonlinear fashion.
OffsetTweak	A method called during serialization and deserialization that performs offset-tweaking after the rest of the serialization or deserialization operation has completed. The function return value is the new absolute file offset.
one-based	Referring to an index value or offset value whose first value is one, next value is two, next value is three, etc. A one-based index or offset can have valid values from one to the total number of valid items. Compare "zero-based."
opcode	An operational code (synonymous with instruction).
operating system (OS)	A combination of programs and drivers that manage system resources.
operator	A symbol that performs an operation (especially mathematical) on one or more operands.
optimization	The concept of making an operation more efficient, faster, and/or smaller.
opt-in path	A path stem that describes portions of paths that can be used to access files in secondary storage.
opt-out path	A path stem that describes portions of paths that cannot be used to access files in secondary storage.
Organized Block	A BAR specific construct that represents a variable-length portion of a file with non-uniform children. This type of block contains an optional header data structure and a mandatory block body.
output buffer	A buffer that is returned to a calling function or calling program.
overhead	The combination of tasks that must be completed as a means to the end of a more general task. Overhead slows down a more general task and generally has a bad connotation.
overwrite	The act of replacing a previously written data location with other data, which causes the previously written data to become lost.
packing	The process of formatting data into a compact, serialized form.
padding	Unused memory whose purpose is to align data on a particular boundary. For example, a three-byte data structure would need a fourth "padding" byte if the structure must be aligned on a four-byte boundary.
parameter	A variable that receives an argument passed to a function during a function call.
parameter scope	During compilation of function parameters, this is the prevailing scope.
parent	A conceptual object in a hierarchy that owns "child" objects. A parent object often contains references to the memory locations of the children, or at least one child if the children are linked to each other.
parsing	The picking apart of complex data with the intent of converting the data to a more usable form.
parse cursor	The cursor used to identify the start of the absolute position in the file used in the next serialize or deserialize operation.
partial byte	The contents of the last unfilled byte from the last serialization of a Bit Field Structure.
partial iteration	A condition in which there is not enough memory to store all the data necessary for a complete iteration.
path stem	A line of text that composes the first part a pathname (or else the entire path).
pathname (path name)	A line of text that instructs an operating system how to reference a file in secondary storage.
pause condition	A condition in which the BAR protocol's serialization or deserialization process is momentarily halted.
pause point	A location in a file during deserialization or serialization that causes the process to pause.
persist	The act of staying in memory during and after a particular operation.

Formatted

Deleted: where

Deleted: where

Term	Definition
physical location in file	A reference to a particular location in a file. Generall, the y, absolute position (byte offset) references it).
platform	A combination of hosting environment and configuration-specific information, which composes the basis for the implementation of a software engine, application, driver, etc.
platform dependence	Relating to different implementations across platforms.
platform independence	Relating to a common implementation across platforms.
pointer	Synonymous with reference. Many programming languages (especially those derived from C language) use pointers.
poorly formed	A format that is poorly designed, is not meant to be consistent, or is not easily interpreted or populated by a computer.
population	Synonymous with writing of data.
procedural manifestations	Portion of a software unit that defines the "how to do it" of the system. Procedural manifestations appear in scripts, but not in data definitions.
procedure	A routine that accomplishes a task. A procedure is generally a function, but is not necessarily required to return a value.
program	A combination of code and data that compose a principal unit of software, designed to accomplish a specific task.
programming language	A protocol that defines the general text-based architecture of a program's source code.
property	A high-level encapsulation of a feature of an object, usually represented by private data, and with a very specific form of public access.
protocol	A set of rules and guidelines that define a system.
query	A search of data that employs a specific algorithm to identify data. A query can "find" one or more than one instances of data that match the search criteria.
RAM	Random Access Memory. This is the primary storage medium for system memory.
reference	A reference specifies the memory location of an object.
registry	A bank of registered information. A file format registry contains compiled file formats.
regular expression	A symbolic expression that characterizes a set of possible conforming sequences of data (matches) to the expression.
relative position	A number describing a position relative to any arbitrary point of reference. A relative position can be positive, zero, or negative.
relocation	The movement of an object from one point in memory to another (this is both conceptual and physical).
repeating list	A Node List whose contents are designed to repeat indefinitely (for any number of iterations).
resume	The act of continuing deserialization or serialization after a pause.
return	The act of exiting from a function call. When a function returns, it often has a return value.
return type	The type of a return value.
return value	A value passed back to a calling function or calling program upon completion of a function call.
robust	Strongly formed or constructed.
routine	A combination of statements to be executed during a function call.
run	Synonymous with execution.
safe pointer	A pointer that can point to only a limited number of memory locations. This type of pointer cannot point to invalid locations except when there is a means to "catch" an attempted reference to an invalid location before it can occur.

Formatted

Deleted: This is g

Deleted: d by absolute position (byte offset

Term	Definition
sandboxed	This adjective refers to the isolation of a software component from other software components in such a way that undesired interaction between components is minimized (i.e. every component plays in its own "sandbox").
saving	Synonymous with file writing.
scope	The point of reference in a declaration; usually applies to compilation process.
script	A series of statements that are interpreted by a computer to act as instructions. A script is not necessarily synonymous with code; code is interpreted and executed literally by a machine's native processing capabilities, while a script often must undergo additional processing before the instructions are carried out.
search	An operation involving examining a set of data with the intent on finding a particular pattern.
secondary storage	Any device designed to act as long-term data storage.
serialization	The concept of creating a sequence of data in serialized form from data in a document object model.
Serialize	A method called during serialization that replaces a node with an entirely different substitute node which does not need to be the same size or type as the original node. The function return value is a pointer to the output buffer to the data in serialized form.
set-aside memory	This is system memory allocated when a BAR_Services object is created. For the duration of the BAR_Services object's operation, it will only use this memory; it will never request more from system memory. Set-aside memory is freed when the BAR_Services object is destroyed.
shared memory	Memory shared between different applications, processes, or machines.
sibling	A node at the same hierarchy level as another node; both have the same parent.
sign propagation	This describes the process of filling the leftmost bits of a right-shifted word with copies of the leftmost bit. Sign propagation characterizes a right-shifted two's complement signed integer. Compare "zero propagation."
signed	Containing a mathematical sign. In programming, a signed number is either zero, positive, or negative.
Simple Structure	BAR terminology for a data structure with a fixed number of named elements, with elements of Simple Type.
Simple Type	BAR terminology for the lowest-level unit type of data. A simple type can be a 1-byte, 2-byte, 4-byte, or 8-byte integer, or a 4-byte or 8-byte floating point number.
single-threaded	Refers to an application or process composed of only one thread. Compare "multi-threaded."
singleton object	An object designed to exist solo; only one instance can be created.
size	Computer terminology for the number of data units representing a type or portion of data. Synonymous with "length."
SOAP	Simple Object Access Protocol. SOAP is a protocol for exchanging XML-based messages over a network.
software requirement	A software-level requirement.
specific construct	A construct in a BAR implementation file with further breakdown in type than a general construct type.
stack	A portion of memory reserved for function return locations, argument storage locations, and local variable storage locations. The stack is designed to "grow" as more functions within functions are called, and "shrink" as functions return.
stack overflow	A condition in which the stack grows to the point where it exceeds its reserved memory.
stand-alone file	A file that lacks dependencies on any other file in order to accomplish a particular purpose.

Formatted

Deleted:

Term	Definition
static	Not changing. In reference to memory, static means the physical location of the memory is always the same.
static library	A type of library that is always loaded and operational when an application is running. Compare "dynamic library."
string	A variable-length buffer of characters.
structured type	A type that breaks down further into sub-fields.
substitute node	A node generated during a Deserialize or Serialize function call, which serves as a replacement for the original node in the hierarchy.
super-parents	Refers to the parent of the parent block, and the parent of the parent of the parent block, etc. all the way up to the top-level node.
support	This refers to providing services and documentation for a particular software component.
swap file	A temporary file created for the purpose of extending the overall amount of de-facto RAM.
synchronization services	Routines offered by an operating system or platform that allow coordination of multiple threads.
system memory	Computer memory that is stored in the short term. This memory generally resides in RAM, but can also reside in swap files.
system requirement	A system-level requirement (not tied specifically to hardware or software).
system resources	Computer resources that BAR uses, but does not own. These include things such as timers, low-level disk services, or thread identifiers.
tab stop	The maximum number of character positions occupied when a tab is inserted.
tabulation	Literally, "entering into a table." Tabulation in computer terms means placing data or references to data in something that can accommodate table-like data, like an array.
tailoring	The concept of modification of a system to meet specific needs of the target implementation.
Termination	A method called during deserialization that determines if a repeating node list should be terminated. The function return value is the termination condition.
termination condition	A condition indicating whether a repeating node list should be terminated. A true condition indicates a last iteration condition; a false condition indicates a continuation condition.
termination sizing	Sizing via termination condition per a Terminate method. See also "explicit sizing" and "implicit sizing."
Text Block	A BAR specific construct that is a subtype of Unorganized Block. A Text Block represents a variable-length portion of a file, composed of a sequence of text characters. The data inside a text block are not further broken down.
text editor	An application that allows one to edit a text file.
text file	A file composed of text characters. Text characters generally include alphabetical, numeric, and punctuation, as well as a few basic formatting characters like carriage returns, line feeds, spaces, and tabs. A text file is readable to most humans. Compare to "binary file."
third-party service	A service package used by the client programmer, but not built by the client programmer and not part of the formal development platform used by the client programmer.
this	An implicit parameter in a member function that refers to the current member scope.
thread	A task that runs on a computer with its own instruction pointer. If there is more than one thread running as part of an application or process, the application or process is multithreaded.
thread-reentrance	A case when more than one thread calls a function at the same time. Thread-reentrance poses problems to designers of applications and operating systems because local and member variable values can be momentarily incorrect at any one time.

Formatted

Formatted

Deleted: is

Deleted: a

Deleted: a function is called by

Term	Definition
time-out	A finite amount of time to complete a task before the task is abandoned. Time-outs are often used with resource locking.
token	A conceptual object during compilation that represents a simple unique unit, such as a keyword, identifier, operator, etc. A token can always be identified by both a number and name.
top-level	The highest object in a hierarchy.
totalfilesize	A built-in global variable that contains the byte size of the entire file.
two's complement	This is a two-step sign conversion operation that works on integers (two's complement signed integers). The first step is to apply a logical NOT to every bit of the integer; the second step is to add the integer by one (wrapping to zero if necessary). Two's complement logic is found in the vast majority of signed integer implementation.
type	In a programming language, this refers to a unit of data with specific, unique characterization. Types generally have their own names, although such names are rarely consistent across programming languages.
type alignment	The required alignment boundary for a type. See "alignment boundary."
type look-up	The process of gathering information about a type.
type modifier	An attribute that changes the characterization of a type, but not so significantly that the type is unrecognizable. For example, "unsigned" modifies the signed integer type "int" to make an unsigned integer type.
typecast	A conversion of one type to another. A typecast can be explicitly written in code, or it can be implicitly implemented as part of a programming language's functionality.
Unicode	A standard for the consistent representation of text characters from languages around the world. Unicode character width is generally 2 bytes or greater.
union	A data structure that contains a single unit of many different types, but in such a manner that the types overlap the same memory. Unlike a variant type, it is not necessarily possible to distinguish type of the stored value upon request.
unique identifier (UID)	A zero-based integer that uniquely identifies a particular component in an implementation file. A unique identifier is only supposed to uniquely identify a component within a specific category of components; for example, a Node List construct and Enumerated Constant Block can have the same value for UID, but the component categories are different.
unit type	The type of a single unit of a larger sequence of data. In the BAR protocol, arrays and Unorganized Blocks require a unit type.
unitary reading	A read operation that reads a single Simple Type into a variable.
unitary writing	A write operation that writes a single Simple Type into a variable.
unload	BAR terminology for the removal of a previously loaded file from memory.
Unorganized Block	A BAR specific construct that represents a variable-length portion of a file, composed of a sequence of a unit type (Simple Type). The data inside an Unorganized Block are not further broken down.
unpacking	The process of formatting data into an uncompact, deserialized form.
unsafe pointer	A pointer that can potentially point to an invalid memory location, but without any reliable means to "catch" attempts to reference this memory location. An unsafe pointer generally consists of an address and no other information.
unsigned	Lacking a mathematical sign. In programming, an unsigned number is either zero or positive.
Validation	A method called during deserialization that determines if a portion of memory qualifies as a particular construct. The function return value is the validation condition.
validation	The process of determining whether or not an object is valid or can be valid if it existed.

Formatted

Deleted: is

Term	Definition
validation condition	A condition indicating whether a construct is valid or not. A true condition indicates valid; a false condition indicates invalid.
variable	A programming and mathematical term for a value that can change. Programming languages treat variables as units of data of a particular type.
variable-length	A portion of data whose size is not consistent. Compare "fixed-length."
variant	A platform-dependent data structure that contains an indeterminate type. The value of the variant can be of any of a number of types, as long as the platform directly supports these types. If requested, the type of the value can always be distinguished.
versioning	The process of maintaining different versions or generations of a particular standard, format, or file.
Visual Basic (VB)	An event-driven programming language developed by Microsoft to provide high-level programming for many of its products and operating systems. This language is mostly derived from BASIC, but places a greater emphasis on events and objects. The functionality of VB is limited in scope and power when compared to other programming languages, such as those derived from C language.
void	A type that indicates an "absence" of type or data, found in many programming languages, especially those derived from the C language. A void return value means no value is returned; a void parameter list means there are no parameters; a void pointer means the pointer does not point to any particular type.
wait state	Synchronization term for a state in which one thread or object must suspend operations until a particular amount of time has passed, or until a condition is encountered that clears the wait state.
well formed	A format that is well designed, with consistency, readability, and ease of use being the primary virtues.
word	A number composed of a fixed number of bits. A word is usually a multiple of 8 bits in size. Many computers support a default word size (generally a multiple of 8 bits) that is the preferred size for calculations.
XML (eXtensible Markup Language)	A general-purpose, text-based markup language, designed to allow sharing of data across different platforms.
XML documents	A text file containing XML.
XML protocol	The open standard recommended by the World Wide Web Consortium for processing XML documents.
XML Schema	An XML Schema Definition is one way to define the format of an XML document. The schema document is itself a well-formed XML document.
XSD file	Standards for "XML Schema Definition." See "XML Schema."
zero propagation	This describes the process of filling the leftmost bits of a right-shifted word with a zero bit. Zero propagation characterizes a right-shifted unsigned integer. Compare "sign propagation."
zero-based	Referring to an index value or offset value whose first value is zero, next value is one, next value is two, etc. A zero-based index or offset can have valid values from zero to one minus the total number of valid items. Compare to "one-based."

Formatted

Deleted: are directly supported by the platform

Deleted: T

Deleted: when asked for

Regular Expression Terms

An EBNF definition is in bold face, with the syntax:

symbol ::= expression

Single Character Matches

The following text represents a character represented by decimal character number. The value of N represents the character number:

#N

The following text represents a character represented by hexadecimal character number. The value of N represents the character number:

#xN

The following text represents a single uppercase character match:

[A-Z]

The following text represents a single lowercase character match:

[a-z]

The following text represents a single numeric character match:

[0-9]

The following text represents characters numbered from 0 to 32:

[#0-#32]

Brackets can include multiple ranges or single character representations to show that a particular range of characters is possible. In the following case, the character can be uppercase, lowercase, numeric, a carriage return character, or a newline character:

Deleted: ,

[a-zA-Z0-9#10#13]

Multiple Character Matches

Sequences of multiple printable characters can appear in double or single quotes. The type used usually depends on whether the other quoted delimiter must appear (for example, "'" to represent a single quote and '"' to represent a double quote):

"string"
'string'

Grouping and Expression Rules

Any of the sequences listed under Single Character Matches or Multiple Character Matches can be used in combination to make more complex sequences. When combined with specific rules, virtually any syntax can be represented.

An expression can also be composed of sub-expressions in addition to character matches. Terms defined using '::=' can also be used in expressions.

If an expression appears inside parentheses, the contents are treated as a group:

(expression)

If multiple expressions appear alongside each other, and are *not* contained within brackets, these expressions must appear in left-to-right sequence. For example, 'S' 'A' 'Y' can only match the word "SAY".

expression1 expression2 expression3 …

If a question mark appears after an expression, the expression is optional. Either 0 or 1 instances can appear:

expression?

If a pipe symbol appears between expressions, a match of either expression is possible:

expression1 | expression2

If a hyphen symbol appears between expressions, the expression must match expression1 *except* for anything that might match expression2:

expression1 - expression2

If a plus symbol appears after an expression, at least one instance of the expression must appear. As few as 1 must appear; there is no upper limit to the number that can appear:

Expression+

If an asterisk symbol appears after an expression, the expression is optional, and any number of instances can appear. As few as 0 can appear; there is no upper limit to the number that can appear:

Expression*