I-beams is a monadic operator that provides a range of system-related services.

Syntax: \( R \leftarrow \{X\} \left(A \right\) Y \)

where:

- \( A \) is an integer that specifies the type of operation to be performed
- \( X \) (optionally) and \( Y \) are described in the following table
- \( R \) is the result of the derived function

### A | Derived Function | Notes |
---|---|---|
8 | Inverted Table Index-of | \( X \) and \( Y \) are inverted tables. |
85 | Execute Expression | \( X \) is a scalar specifying whether to retain the shy result obtained by executing expression \( Y \) (a character vector). Possible values are:  
  - 0: retain shy results  
  - 1: discard shy results (default) |
127 | Overwrite Free Pockets | Overwrites all unused data pockets in the workspace, thereby removing any remnants of potentially secure data. Returns 1 when successful. \( Y \) is any empty array, preferably \( \emptyset \). |
180 | Canonical Representation | Similar to monadic \( \Box CR \) but enables the canonical representation to be obtained for methods in classes as well as functions and operators. \( Y \) is a simple character scalar or vector comprising the name of a defined, system or primitive function or operator or the class.method name. |
181 | Unsqueezed Type | Similar to monadic \( \Box DR \) but does not squeeze the data. Returns an integer indicating the array type. \( Y \) is any array. |
200 | Syntax Colouring Descriptions | Returns a 4-column matrix (token type, value, specific token and TTY colour number) of syntax colouring descriptions. \( Y \) is \( \emptyset \). |
201 | Syntax Colouring Descriptions | Returns syntax colouring information for the APL code specified in \( Y \) (a vector of character vectors containing the \( \Box NR \) representation of a function or operator). The output of 201\( \mathcal{X} \) can be used to interpret the returned information. |
219 | Compress/Decompress Vector of Short Integers | \( X \) is a scalar or 1-item (optionally, 2-item) vector specifying the compression library to use. Possible values are:  
  - 1: use the LZ4 compression library  
  - 2: use the zlib compression library  
  - 3: use the gzip compression library  

  If \( X[1] \) is positive, then compress. In this situation:  
  - \( X[2] \) specifies the compression level in the range 0 to 9 (only if \( X[1] \) is not 1)  
  - \( Y \) must be a simple integer vector of input data with items in the range -128 to 127  

  If \( X[1] \) is negative, then decompress. In this situation:  
  - \( X[2] \) specifies the length of the uncompressed data  
  - \( Y \) must be a simple integer vector of compressed data with items in the range -128 to 127  

  If \( X \) is 0, then decompress. In this situation:  
  - \( Y \) must be the 2-item vector of vectors produced by a previous 219\( \mathcal{X} \) compression |
220 | Serialise/Deserialise Array | \( X \) specifies whether \( Y \) is to be serialised or deserialised. Possible values are:  
  - 1: \( Y \) can be any array – this array is then serialised  
  - 0: \( Y \) must be a simple integer vector with items in the range -128 to 127 that must have been serialised using this I-beam – this array is then deserialised |
400 | Compiler Control | Controls the actions of the Compiler. The nature of \( Y \) and \( R \) depend on the value of \( X \). Possible values for \( X \) are:  
  - 0: set automatic compilation options (default)  
  - If \( Y \) is 0, disable automatic compilation (initial setting)  
  - If \( Y \) is 1, compile functions when they are fixed (with \( \mathcal{F} X \) or from the function editor)
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| **600** Disable Traps | Controls whether the trapping mechanism is active. Y is an integer whose possible values are:  
  * 0: all traps are enabled  
  * 1: all traps are disabled  
  * 2: when in suspended functions, errors generated by expressions typed in the Session do not fire traps (default) |
| **739** Temporary Directory | Returns the name of a temporary system directory suitable for user files (no trailing separator is included). Y is 0. |
| **900** Called Monadically? | When included within a tradfn/tradop, returns a Boolean value indicating whether the function/operator was called monadically (1) or not (0). Y is any array (ignored). |
| **950** List Loaded Libraries | Lists the dynamic link libraries that have been loaded by ⎕NA and are still loaded. Y is the empty vector ⍬. |
| **1010** Set Shell Script Debug Options | Sets options for debugging APL "shell scripts". Y is an integer whose possible values are:  
  * 1: lines in the script are echoed to stderr prior to execution  
  * 2: behaves as if ⍵TRACE is set for every line of every function in the script  
  * 3: a combination of the other two options  
Y optionally specifies a character scalar/vector that prefixes each line of output (the default is 't'). If Y is not specified, the previous value of Y is returned. |
| **1111** Number of Threads | Y is an integer specifying one of the following:  
  * the number of threads to be used for parallel execution (the previous value is returned)  
  * the number of virtual processors in the machine is returned |
| **1112** Parallel Execution Threshold | Y is an integer specifying the array size threshold at which parallel execution takes place (the previous value is returned). |
| **1159** Update Function Time and User Stamp | X is an array of function attributes in same format as the output of ⎕AT  
Y is an array of function names in same format as the right argument of ⎕AT |
| **1200** Format Date-Time | X is a character scalar or vector specifying the formatting to apply to the elements in Y  
Y is a numeric array of any shape, where every element contains a Dyalog Date Number that represents a date between 1 January 0001 and 28 February 4000 |
| **1500** Hash Array | Y is any array. R is dependent on X. Possible values of X are:  
  * 1: R is an integer reporting on the hash status of Y. Possible values of R are:  
    * 0: Y has not been marked for hashing  
    * 1: Y has been marked for hashing but does not yet have a hash table  
    * 2: Y has a hash table  
  * 2: R is the unhashed form of Y  
If X is not specified, R is a copy of array Y that has been marked for hashing (the hash table will be created the first time the array is used as an argument to 1 or other set functions). |
<table>
<thead>
<tr>
<th>Year</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| 2000 | Memory Manager Statistics | Y is an integer or vector of integers specifying the statistics to be displayed (if X is not specified) or set (if X is specified). Possible values are:  
- 0: workspace available  
- 1: workspace used  
- 2: number of compactions since loading workspace  
- 3: number of garbage collections that found garbage  
- 4: number of garbage pockets currently in workspace  
- 12: sediment size  
- 13: amount of memory currently being used in workspace  
- 14: maximum amount of memory used since workspace was loaded  
- 15: limit on minimum workspace allocation  
- 16: limit on maximum workspace allocation  
- 19: number of calls to DWA or 2002z since the last time 2000z was called (or since the process started if this is the first call to 2000z).  
Optionally, X is an integer or vector of integers of the same length as Y specifying the value to change the specified Y item to. Possible values are:  
- for Y is 2, X must be 0 (resets the compaction count)  
- for Y is 3, X must be 0 (resets the garbage collection count)  
- for Y is 14, X must be 0 (resets the amount of memory used since ws was loaded)  
- for Y is 15, X must be between 0 and the current workspace allocation (sets the minimum workspace allocation)  
- for Y is 16, X must be between the current workspace allocation and MAXWS (sets the maximum workspace allocation) |
| 2002 | Specify Workspace Available | Similar to DWA but allows the memory allocation to be specified explicitly. Returns an integer indicating the size (in bytes) of the memory committed for the workspace. Y is an integer specifying the size (in bytes) of the extra memory that is added to the compacted workspace before de-committing the remainder. |
| 2007 | Disable Global Triggers | Controls whether global triggers are active (useful when databinding) – only active in the APL thread in which it is called. Y is an integer whose possible values are:  
- 0: all global triggers are enabled (default)  
- 1: all global triggers are disabled |
| 2010 | Update DataTable | NOTE: Not supported when using .NET Core.  
X is a Boolean vector with same number of items as the instance of System.Data.DataTable matrix has columns (a 1 indicates that the corresponding column contains strings that must be passed to the Parse method of the data type).  
Y is a 2, 3 or 4-item array comprising (in this order):  
- a reference to the instance of System.Data.DataTable  
- a matrix with the same number of columns as the instance of System.Data.DataTable  
- a vector with the same number of items as the instance of System.Data.DataTable matrix has columns, with each item specifying the value to map to DBNull when this column is written to the instance of System.Data.DataTable  
- Row indices (zero origin) of the rows to be updated; if omitted, then data will be appended to the instance of System.Data.DataTable |
| 2011 | Read DataTable | NOTE: Not supported when using .NET Core.  
Y is a 1 or 2-item array (scalar or vector) comprising (in this order):  
- a reference to the instance of System.Data.DataTable  
- a vector with the same number of items as the instance of System.Data.DataTable has columns, with each item specifying the value that a DBNull in the column should be mapped to when this column is read  
The Invert variant option (default = 0) determines R:  
- 0: R is a matrix with the same shape as the DataTable referenced by \( \succ \)Y  
- 1: R is a vector whose length is the same as the number of columns in the DataTable referenced by \( \succ \)Y  
X is a numeric vector whose length is the same as the number of columns in the DataTable referenced by \( \succ \)Y (if X has fewer elements than there are columns then the missing values are assumed to be 0 and those columns are not transformed):  
- 1: Specifies that the corresponding column of the result should be converted to a string using the ToString method of the data type of the column.  
- 2: Specifies that numbers of type System.Int64 in the corresponding column of...
<table>
<thead>
<tr>
<th>Year</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>Remove Data Binding</td>
<td>Not supported when using .NET Core. Disassociates a data-bound variable from its data binding source. Returns 1 when successful. Y must be a character vector containing the name of the data-bound variable to be disassociated (otherwise all databinding is removed from the workspace).</td>
</tr>
<tr>
<td>2015</td>
<td>Create Data Binding Source</td>
<td>X is optional; if omitted, then default binding types are used. If included, its structure is dependent on the content of Y. Y is a character vector comprising the name of one of the following: a matrix: X is a two-column matrix specifying the name and binding type for each column in matrix Y a variable: X is a single .NET type specifying the binding type for variable Y a namespace containing variables(s): X is a two-column matrix specifying the name and binding type for each variable in namespace Y a variable containing vector of refs to namespaces containing variables(s): X is a two-column matrix specifying the name and binding type for each variable in each namespace.</td>
</tr>
<tr>
<td>2016</td>
<td>Create .NET Delegate</td>
<td>NOTE: Not supported when using .NET Core. Returns an instance of the .NET type specified in Y[1] that can be used to invoke the function in Y[2]. Y is a vector comprising: [1] is a .NET type that derives from System.Delegate, for example, System.EventHandler [2] is either the name or the ⎕ OR of a function to be used as a callback.</td>
</tr>
<tr>
<td>2017</td>
<td>Identify .NET Type</td>
<td>NOTE: Not supported when using .NET Core. Returns the .NET type of Y for types that are located in any assembly that has been loaded into the current AppDomain by calling ⎕USING or using (the assembly-qualified name is required by System.Type.GetType). Y is a character string containing the name of a .NET object (namespace names can be omitted if they are provided in elements of ⎕USING).</td>
</tr>
<tr>
<td>2022</td>
<td>Flush Session Caption</td>
<td>Updates the Session caption. Y is any array (ignored).</td>
</tr>
<tr>
<td>2023</td>
<td>Close all Windows</td>
<td>Closes all open Edit/Trace windows without resetting the state indicator. Returns 1 when successful. Y is any array (ignored).</td>
</tr>
<tr>
<td>2035</td>
<td>Set Dyalog Pixel Type</td>
<td>Specifies how Coord 'Pixel' is interpreted by all GUI operations. Y is a character vector whose possible values are: 'ScaledPixel' 'RealPixel'</td>
</tr>
<tr>
<td>2041</td>
<td>Override COM Default Value</td>
<td>By default, if a COM object is in error or is of a type that cannot be represented in APL, then an error is generated in the Session; if the COM object is of type VT_EMPTY then ⎕NULL is returned. Y is an integer whose possible values are: 1: X specifies the value that is returned instead of ⎕NULL when the COM object is of type VT_EMPTY 2: X specifies the value that is returned when the COM object is in error or is of a type that cannot be represented in APL Omitting X restores the default behaviour.</td>
</tr>
<tr>
<td>2100</td>
<td>Export to Memory</td>
<td>Exports the active workspace as an in-memory .NET assembly. Returns 1 when successful. Y is any array (ignored).</td>
</tr>
<tr>
<td>2101</td>
<td>Close .NET AppDomain</td>
<td>Close the current .NET AppDomain (started by the current APL process). Returns 0 when successful, otherwise returns a non-zero integer. Y is any array (ignored).</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Function</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| 2250  | Verify .NET Interface                                                      | Provides information about the Dyalog-.NET interface. Y must be 0. R is a vector of vectors in which [1] indicates .NET support, [2] indicates failure (0) or success (1) in loading, and [3] is a text vector containing error messages generated during load. Possible values of R[1] are:  
  - 1 : .NET is not supported  
  - 0 : .NET is supported but not configured  
  - 1 : Configured to use .NET Core  
  - 2 : Configured to use .NET Framework (Microsoft Windows only) |
| 2400  | Set Workspace Save Options (workspace specific)                          | Specifies whether Trace, Stop and Monitor settings are cleared when workspace is saved. Y is an integer whose possible values are:  
  - 0 : settings are not cleared on saving (default)  
  - 1 : settings are cleared on saving |
| 2401  | Expose Root Properties                                                    | Specifies whether Root Properties, Events and Methods are exposed. Y is an integer whose possible values are:  
  - 0 : no further Root Properties, Events and Methods are exposed  
  - 1 : Root Properties, Events and Methods are exposed (default) |
| 2501  | Discard Thread on Exit                                                    | Specifies whether threads created to serve incoming .NET requests are destroyed or persist (the default) on completion of their task. Y is an integer singleton; when set to 0 on the current thread, that thread is destroyed on termination rather than persisting. |
| 2502  | Discard Parked Threads                                                    | Destroys all persistent threads in the workspace. Y is any array (ignored). |
| 2503  | Mark Thread as Uninterruptible                                           | Specifies whether a thread and/or its children respond to a weak interrupt. Y is an integer whose possible values are:  
  - 0 : the thread and its children are interruptible (default)  
  - 1 : mark thread as uninterruptible  
  - 2 : mark children of the thread as uninterruptible  
  - 3 : mark thread and its children as uninterruptible |
| 2520  | Use Separate Thread for .NET                                             | NOTE: Not supported when using .NET Core. Specifies whether .NET code run on APL thread 0 runs on the same operating system thread as the session. Y is an integer whose possible values are:  
  - 0 : .NET code runs on the same thread as the session (default)  
  - 1 : .NET code called from APL thread 0 runs on its own thread |
| 2704  | Continue Autosave                                                         | Enables or disables the automatic saving of a CONTINUE workspace when Dyalog exits. Y is an integer whose possible values are:  
  - 0 : disable the automatic saving of a CONTINUE workspace  
  - 1 : enable the automatic saving of a CONTINUE workspace |
| 3002  | Disable Component Checksum Validation (system wide)                      | Specifies whether checksums are validated by \[F\READ. Y is an integer whose possible values are:  
  - 0 : \[F\READ will not validate checksums  
  - 1 : \[F\READ will validate checksums (default) |
| 3500  | Send text to RIDE-embedded browser                                       | Optionally, X is a simple character vector, the contents of which are used as the caption for the tab in the RIDE client that contains the embedded browser. If omitted, then the default is "HTML". Y is a simple character vector the contents of which are displayed in the embedded browser tab. To include SVG content, the HTML text in Y must include <meta http-equiv="X-UA-Compatible" content="IE=9">. R identifies whether the write to the RIDE was successful. Possible values are:  
  - 0 : the write to the RIDE client was successful  
  - 1 : the RIDE client is not enabled  
  any other value : contact support@dyalog.com |
| 3501  | Connected to the RIDE?                                                    | X and Y are any value (ignored). R identifies whether the Session is running through the RIDE. Possible values are:  
  - 0 : the Session is not running through the RIDE  
  - 1 : the Session is running through the RIDE |
### Manage RIDE Connections

Controls connections between the RIDE and an interpreter. Returns 0 if successful or a positive or negative integer if unsuccessful. Y has the following possible values:
- 0: disable any active RIDE connections – only valid when the RIDE is enabled
- 1: enable the RIDE using the initialisation string defined in the RIDE_INIT configuration parameter – only valid when the RIDE is not enabled
- a simple character vector: specifies an initialisation string that replaces the RIDE_INIT configuration parameter – only valid when the RIDE is not enabled

On a run-time interpreter, 350211 is the only way to enable the RIDE.

### Fork New Task

Initiates a new APL process with the same execution stack and runs the task in both processes. Returns 0 in the child process and the child’s process ID in the parent process. Y is a simple empty vector (ignored).

### Change User (system wide)

Should only be run as root. Changes the effective user ID for the process. Runs the user name specified in Y (a character vector specifying a valid UNIX name) if successful.

### Reap Forked Tasks

Returns a matrix of newly-terminated child processes along with information about each of those processes (including whether the process terminated normally or as a result of a signal). The first three of the 18 columns indicate:
- R[;1] is the process ID of the terminated child
- R[;2] is the signal number that caused the child process to terminate (¯1 if the process terminated normally)
- R[;3] is the exit code of the child process (¯1 if the process terminated as the result of a signal)

Y is a simple empty vector (ignored).

### Signal Counts

Returns an integer vector of signal counts whose length corresponds to the number of signals supported by the operating system. Elements are the counts of SIGHUP, SIGINT, SIGQUIT, SIGTERM and SIGWINCH signals (others are 0).

Y is a simple empty vector (ignored).

### List Loaded Files

Returns a list of all of the files that are associated with objects in the active workspace, together with information about those files. Y is any array (ignored).

### List Loaded File Objects

Returns details of all the objects in the active workspace that are associated with a file. Y is an empty array (ignored).

### Remove Loaded File Object Info

Removes file/script information about single workspace object Y from the workspace.

### Load File Object Info

Returns file/script information about single workspace object Y.

### Sample Probability Distribution

Generates an array of random numbers from a named probability distribution. Y is a 2-item vector specifying the name of the probability distribution and the shape of the result. X is a scalar or 1- or 2-item numeric vector that specifies parameters for the named distribution.

### Line Count

Restricts the number of calls to ⎕LC, thereby potentially improving performance. Y is any positive integer; R returns at most the first Y elements of ⎕LC.