

MagicLogic Optimization Inc.
Cube-IQ 5.0 & BlackBox 4.0
XML Input & Output Specification

Synopsis	This document specifies the format of the XML data files which are used and / or created by both Cube-IQ 5.0 and the BlackBoxTCP 4.0 Load Planning software.
Version	5.0.0
Date	2015/06/23
Author	T.N. Smith / A. Jonker / R. Jonker

Revision Information

Last update	12 July 2015
Status	Released
Internal Document Location	T:\Documents\XML\

Contents

Revision Information	2
Contents	3
Introduction.....	5
Glossary of Terms.....	5
Inputs.....	7
Output	8
Input Data: Static Data.....	9
Input Data: Volatile Data (or “Load”)	10
Output Data: Loading Results.....	11
Data Description	12
Static Data: Parameters	13
Static Data: Container Settings	15
Static Data: Product Data.....	19
Static Data: Container Data	23
Static Data: Zones	27
Static Data: Lost Spaces (Unusable Spaces).....	29
Static Data: Orientation Rules	30
Static Data: Configurations.....	31
Static Data: Stacking.....	33
Static Data: Customer Data (Cube-IQ 4.0 GUI only).....	34
Volatile Data: Product to Load	35
Volatile Data: Fixed Boxes.....	37
Volatile Data: Container to Load.....	38
Volatile Data: Stage	39
Output: Loading Results	41
System Information.....	41
Load	41
Loaded Container.....	43

Block Information.....	46
Sample Data: Input	49
Input: Static Data	49
Input: Volatile Data (Loading Case).....	51
Sample Data: Output.....	52

Introduction

This document describes the XML data format required by the Cube-IQ “BlackBox” Load Optimizer. The BlackBox uses XML for both input (to define container and product data), and for output (to describe loading solutions).

Glossary of Terms

- **Load** – this is a collective term to describe all the data required for the BlackBox to perform optimization. In other words, one or more containers and one or more products.
- **Container** – the BlackBox uses the term “container” to refer to anything which is to be used to hold products. Containers include: pallets, trucks, railcars, shipping containers, ULDs etc.
- **Product** – is the internal term for items which are to be placed into a container, e.g. products, SKUs, items or collie.
- **Orientation** – an optional attribute, applied per product, that controls the stacking and positioning of products in any of its six possible orientations. (Flat, Turned, On Left, On Right, On Front, On Back)
- **Configuration** – an optional attribute, applied per product, that not only controls orientations, but also how multiples of the same product may be loaded. Sometimes abbreviated to 'Config'.
- **ContainerToLoad** – defines the availability of a Container within a Load. Through the ContainerId, it refers to the ‘container’ for the physical data, like dims and weight capacity.
- **ProductToLoad** – defines the product (and its quantity, sequence, ..) that is to be loaded as part of a Load. For the product’s physical data, it refers to Product using the ProductId.
- **LoadedContainer** – is the term for a container which has been utilized by the BlackBox and now contains one or more products.

- **Block** – a “block” is a group of products placed within a container. A Block only contain one product type, and the products are all in the same orientation.
- **Static Data** – data that does not tend to change much: physical container and product dimensions, system settings and parameters.
- **Volatile Data** – data which changes for every load: lists of products which are to be loaded, and the list of containers to be used for this particular instance.

Inputs

Inputs are provided to the BlackBox in two forms:

1. “Static Data” represents data items which are loaded by the BlackBox once, during initialisation when the program starts up. This is useful for data which are likely to be required for a number of different loading cases. For example, commonly used container and product types should be stored in the system as static data for increased efficiency.

Configuration and settings information for the BlackBox system, which is used to tune the optimizer, can also be included in the static data.

Static data is stored in a single XML or CSV file, which can be located in any folder that the BlackBox program can access both at startup and during runtime. You can use any name for this file – and you must then specify the same name in the BlackBoxTCP.ini startup file as described in the Installation Guide.

2. “Volatile Data” represents a single loading case to be solved by the optimizer. Volatile data can also include new container and product definitions. Any such container and/or product data will be retained in the BlackBox until the program exits, and can be referenced by any future loading cases while the program is still running. Please note however that once the BlackBox program terminates, this data will *not* be preserved.

Please refer to the following sections for further information on the type and format of the data which can be provided to the BlackBox.

Output

Every loading is created by the BlackBox as a single XML file containing a number of XML elements. Please refer to the section “Solution Data” for details on the output file created by the system. Output files are named based on the “LoadID” tag specified in the loading case input data.

So for example a load named “Pallet-2008-07-07” would be written to an output file called “Pallet-2008-07-07.xml”

Input Data: Static Data

Static data is optional, as long as you then provide all relevant container and product data as part of the volatile data (or “loading case,” see Volatile Data section below). Where static data is not used, the BlackBox operates using a set of default configuration values which are appropriate for most loading situations. Please refer to technical support if you have specific configuration or tuning requirements for this system or if you would like assistance in creating a static data file containing your existing container and / or product data.

Static data can be considered to be the “database” for the BlackBox system, consisting of settings, containers and products which may need to be loaded by the system many times.

Static Data typically includes the following:

- **Settings** which can be used to control how the loading algorithm works, general loading rules etc.
- **Container settings** which control how the algorithm will treat the loading of each particular type of container. (e.g. Palletizing rules, center of gravity etc.)
- **Container** data which defines the size and shape of all the containers you may wish to load.
- **Product** data defines the size and shape of items which may need to be placed into the containers.
- **Orientation / Configuration** data which can be used to define any special positional loading rules which should be applied to certain product types.

It is possible to update the static data during the operation of the BlackBox if any changes are required. The system can be triggered to re-load the static data file after any updates have been made, or can reload automatically at pre-set times. Please refer to the BlackBox installation and user guide for further information.

Note that additional static data items, containers and/or products, can also be provided as part of the loading case if required on an ad hoc basis.

Input Data: Volatile Data (or “Load”)

Volatile data represents a single loading case, and includes at least the following:

- A “Load” line which simply names the loading case and optionally operational and/or business rules.
- A number of “Containers to be loaded” which refer back to containers defined in the static data.
- A number of “Products to be loaded” which refer back to products defined in the static data.

You can also optionally provide static data type items, such as containers and products embedded in the volatile data file in order to create self-contained orders which describe all the containers and products that form the load.

Output Data: Loading Results

Once the BlackBox optimizer has produced a loading solution, the system writes a load file in XML format. The output file contains the following:

- A “Load” line which summarises the load which has been created – including the volume and weight loaded, the number of containers used, number of blocks (groups of like products) created etc.
- A number of “loaded container” records, identifying which containers have been used in this load.
- A number of “blocks” which represent groups of products in the load.
- Additionally, the BlackBox will list any products which have not been loaded successfully.
- Optionally, the BlackBox can also output all the input data, for instance if you wish to be able to handle the results as a self-contained Load.

Data Description

Please refer to the example data at the end of the document for further information. There are also example files available as downloads from the MagicLogic website, or can be created using the Cube-IQ system itself (File->Export Data).

The XML schema is shown in the appendix.

*Please contact MagicLogic for access to test data to help you set up and configure the BlackBox system.
MagicLogic will also always help you transform your own data into the required format.*

Data item	Type	Range	Required?	Information
This field column shows the data item name, exactly as it must be specified in the XML file.	This column defines the data type of the item	The range of possible values that this data item may hold	States whether this data item MUST be specified in the XML, or whether it can be omitted. If omitted, a safe default value is used (see “Information” column)	Brief description of the purpose and function of this data item, and its default value.

Static Data: Parameters

Note that a large proportion of the complete Cube-IQ settings element is not relevant to the BlackBox system. The entire family of Cube-IQ products share the same schema and content in order to support cross-product compatibility.

Note: **only those data items which are used by the BlackBox system are shown here.** Any other items are optional and can will be ignored by the BlackBox, and omitted altogether from your input, since they are only of relevance to other Cube-IQ products.

XML syntax:

```
<parameters><parameter ... /></parameters>
```

Data item	Type	Range	Default	Required?	Information
donotsplitintosmallerconfigs	boolean	True/false	False	N	Use only the largest Config that meets the quantity, never multiples of smaller Configs.
combineposonlyinlaststage	boolean				Do not use
configpriorityisstage	boolean				Do not use
featurescode					Do not use
groupnumberisprimarysequence	boolean				Client special
maxnumdecimals	integer	1-4	2	Y	Accuracy to which the optimizer will work. Useful if you are working in large units such as cubic metres.
minbottomonlyweight	float	0.0..999999.9	999999.9	N	Minimum weight an item must be if it is to be loaded on the bottom of the container.
outputloadedcntnrstartnumber	integer	0..99999	0 (no offset)	N	Offset applied to the container sequence number when the load is written back to the output.
productsuprightonpallets	boolean	true/false	False	N	Forces products to be loaded upright only when palletized. Default: False
palletsalwaysstraightup	boolean	true/false	False	N	Controls whether pallets should always be loaded into a container upright or whether they can be tipped on their side.
runtimeunit	string	“sec” or “msec”	“sec”	N	Defines the units that are used in the field ‘maxruntime’ (see Container Settings section below)
sequencemeansdrop	boolean	True/false	false	N	If true, products will be loaded into the container in such a way that

					they can be unloaded again in the sequence specified by the 'sequence' value of each productload.
standarddoorwaypallet	string		""	N	This is a client-specific feature and should be omitted
useonlylargestpossibleconfig	boolean	True/False	False	N	If true, products which have associated config elements will be loaded only by using the largest config.
usesequencewithingroups	boolean	True/false	False	N	This is a client-specific feature and should be omitted or set to "False"

Static Data: Container Settings

Container settings are used by the optimizer to gain information on how to handle and load individual containers of each discrete type. Generally speaking, once these settings have been created they would remain unchanged.

Mandatory fields are **highlighted**.

XML Syntax:

```
<settings><setting ... /></settings>
```

Data item	Type	Range	Default	Required?	Information
settingsid	string	n/a	'default'	Y	Key Field. This string is used in the container record to identify which of these settings records to use when the optimizer processes this container type.
absmaxunitoverhang	float	0..99999	0	N	How much overhang is allowed for a product when it is stacked on top of another product. This is an absolute value, specified using your current units.
action	string	'delete', 'overwrite'			If 'delete', the setting will be deleted from the database, if 'overwrite' the setting data will be overwritten without confirmation being asked
braceunitsatback	boolean	true/false	False	N	Sets whether loaded units must have some other unit close enough to its back, for bracing purposes.
checkfeasibilityonly	boolean	true/false	False	N	Stop optimizing as soon as everything fits
combinesamecodeonly	boolean	true/false	False	N	Products can only be combined in one loaded container if both have the same Combination Code, or one that is blank.
dblenthunit					Unit of measurement
defaultclampspace					Client special
estimatepercentage					Only used for estimate of how many containers are required for a load.
keepitemsinonecontainer	boolean	true/false	False	N	If specified, controls whether all products with the same Id should be in one container (where possible)
lengthcogbackmargin	Float (percentag	0..100.0	0.0	N	Attempt to position the Center of Gravity between the front and back margin

lengthcogfrontmargin	Float (percentag	0..100.0	100.0	N	
loadbynetweight	boolean	true/false	False	N	Client special
loadingmargin	float	0..99999	0	N	Distance to allow around the load for padding and bracing
loadproductsconsecutively	boolean	true/false	False	N	All products with the same Id are to be loaded
loadproducttogether	boolean	true/false	False	N	All products with the same Id should be grouped together (where possible)
maxblocklengthinloadplans	float	1..Any positive number	1000000	N	The maximum distance that the loaders can reach over existing products in the container, to place more products.
maximumdoorwayunderhang					Client special
maxlengthgap	float	Any positive number	0	N	The largest gap allowable in the length direction
maxnonimproveiters	integer	1..n	100	Y	The number of times the BlackBox should “look ahead” for an improved load before stopping.
maxpalletoverhang	float	0..99999	0 (no overhang)	N	Maximum distance a product can overhang any edge of the pallet.
maxreach	float	1..Any positive number	1000000	N	Maximum size that a “block” of products may be. (See definition of blocks in the glossary)
maxruntime	integer	1..n	60	Y	The maximum amount of time the optimizer will spend, in seconds, optimizing the load for each container
maxsupportheightdiff	float	0..99999	0	N	Maximum difference in heights between two supporting products that can still have product loaded that rests on both.
maxsupportheightdiffatop	float	0..99999	0	N	Maximum difference in heights between two supporting products that can still have product loaded that rests on both.
maxwidthgap	float	0..99999	0	N	The largest gap allowable in the width direction
minlengthgap	float	0..99999	0 (no gap)	N	The smallest gap allowable in the length direction
minsupportrequired	float	0..1	0.7 (70%)	N	Minimum area of the product which must be supported by other products or the container floor.
minunitsperblock	integer	1..99999	1	N	Smallest number of identical products to be grouped (see glossary)
minunitspercontainer	integer	1..99999	1	N	Target minimum number of products to be placed in the

					container.
minutildiffboxtarget	Integer (percentage)	0..100	80 (%)	N	Container utilization to achieve when loading different products in the same container. (see "targetnumdiffboxes")
minwidthgap	float	0..99999	0	N	The smallest gap allowable in the width direction
neversplitboxes	boolean	true/false	false	N	Controls loading of all boxes of the same type in one container.
nocrossstacking	boolean	true/false	false	N	Controls cross-stack when building the load.
onesetpercontainer	boolean	true/false	false	N	Client special
overallmaxstack	integer	0..9999	9999	N	Limits the stacking count of all products (e.g. "no box may stack more than "x" high")
pinwheelspreferred	boolean	True/false	false	N	When building pallets, try to give boxes rotational symmetry for extra stability
pinwheelsnotallowed	boolean	True/false	false	N	When building pallets, avoid placing complete pinwheels of boxes
productsuprightonly					all boxes to be loaded straight up if in pallet container
rollloadingrule	Integer	1..4			This rule sets how rolls may be loaded, eg only in pyramids, or can the walls be used for support.(1) = Free, (2) Pyramid, (3) Straight allowed, (4) straight only
rowlengthtolerance					Client special
rowminbracingheight					Client special
rowminwidth					Client special
rownumtolookahead					Client special
seqseparatorthickness	float	0..99999	0.0	N	Thickness of padding required between differently sequenced products (e.g. bulkhead, bracing etc)
sequencemixok	boolean	true/false	True	N	Controls whether products with different sequence numbers can be mixed together.
singlestackonly	boolean	true/false	False	N	Load products in stacks only
singleunitblocks	boolean	true/false	False	N	Load products in stacks only
stacksamecodeonly	boolean	True/false	False	N	Only products that have the same stack code may be stacked
stacksamesequenceonly	boolean	True/False	False	N	Only products with the same sequence number may be stacked (see ProductsToLoad)
standarddoorwaypallet	string	n/a	''	N	Client special
straightstackonly	boolean	true/false	False	N	When building pallets, build "chimneys"
targetnumdiffboxes	integer	0..99999	0	N	Number of different products the optimizer should attempt to

					load in the same container.
unitheightmargin	float	0..99999	0	N	Amount of margin to allow around each product in the height direction. (e.g. for packing material, padding etc)
unitlengthmargin	float	0..99999	0	N	Amount of margin to allow around each product in the length direction. (e.g. for packing material, padding etc)
unitwidthmargin	float	0..99999	0	N	Amount of margin to allow around each product in the width direction. (e.g. for packing material, padding etc)
unloadedunitstonextstage	boolean				If True, whatever does not get loaded in a stage, will be available for loading in the next stage. If False, such items will will be dropped from the Load
widthcogleftmargin	Float (percentag	0..100.0	0.0 (%)	N	Attempt to position the Center of Gravity between the left and right margins
widthcogrightmargin	Float (percentag	0..100.0	100.0 (%)	N	

Static Data: Product Data

Mandatory fields are **highlighted**. All other fields may safely be omitted from your XML.

XML Syntax:

```
<products>
<product ... />
</products>
```

Data item	Type	Range	Default	Required?	Information
productid	string	n/a	"	Y	Key Field. Unique identifier for this product type
action	string	'delete', 'overwrite'			If 'delete', the product will be deleted from the database, if 'overwrite' the product data will be overwritten without confirmation being asked
attribute1					these text strings can be used to 'filter' the products if the user wants to see just a subset of them
attribute2					
attribute3					
attribute4					
backdepth	float	0..99999	0	N	Relevant only for products of type "sofa"
bfvolume					Volume in Boardfeet (lumber)
bottomonly	boolean	true/false	N	N	Specifies whether this product must be on the floor of the container
bundlequantity					– if > 1, the product will only be loaded in bundles of this quantity (stacked in a single pile in the height direction), with the exception of a single bundle that can have a lower quantity in order to reach the quantity that must be loaded. So, if BundleQuantity is 12, a loadable quantity of 40 will be loaded as three blocks/bundles of 12, and one of 4.
carton					Client special
category					Client special
cntnrmatchingcode	string	Up to 32 chars	n/a	N	If specified, this product will only be loaded into a container that has the same value in its 'pckmatchingcode' field.
color	string	n/a	Random	N	Color to be used for this product if it is displayed graphically. Color is specified as 3x byte values representing Red/Green/Blue: #123#456#789#000
combinecode	string	Up to 32	n/a	N	A reference to the combination matrix that controls which other

		chars			products this product can be combined with (i.e. loaded into the same container)
configunitisperc		True/false	N		Client special
cost	float	0..99999	0	N	Value of this product, used for cost reports
dblenthunit					Unit of measurement
dbvolumeunit					Unit of measurement
dbweightunit					Unit of measurement
defaultbatch					Assigns this value as a sequence /batch index to this product, when it gets added to a Load.
defaultgroup	integer	0..99999	0	N	Assigns this value as a group index to this product, when it gets added to a Load. Groups of products can be loaded together.
deletewithload	boolean	true/false	N	N	If set to "true" this product will be automatically deleted from the BB's internal database after the load has been created. This helps to keep the BB's memory footprint small, and can improve performance.
description					Description
displaystring					If not an empty string, this will be used to identify products in the various Load Plans and Diagrams.
drawaspallet					If true, the product will get drawn as a pallet in the load Plans. But it will still get loaded as 'Box'.
endupok	boolean	true/false	Y	N	Product can be loaded on end (i.e. tipped 'forward' by 90 degrees; Length becomes height);
flatok	boolean	true/false	Y	N	Can be stacked in its default orientation (height = height)
foundationheight	float	0..99999	0	N	Use only on advice from MagicLogic.
height	float	0.01..99999	0	Y	Product height in the current units
length	float	0.01..99999	0	Y	Product length in the current units
lengthextrapadding	float	0..99999	0	N	Added to the product's length dimension when loading, to account for packaging or padding.
lineal					A separate field to add a length to a product, only used when loading lumber. Usually the 'Length' is in Inch, and used for loading, and the Lineal is in feet, and used for tallies.
loadableinownbox					If True, this item is assumed to be shippable in 'as is'. It will only get added to a container (typically a carton) when space is available anyway, without adding to the number of cartons, or to their size.
loadonoutside	boolean	true/false	N	N	Must be loaded at one of the outer edges of the container
loadseparately	boolean	true/false	N	N	Will only be loaded if this does not lead to using an extra, or even a larger container

masterid	string	'			If two products have the same 'masterid', they will be combined during the optimization, and, for example, kept together if there is a 'keep products together' rule
maxinlayer	integer	0..99999	0	N	Maximum number of products which can be stacked (0 means unlimited stacking)
maxnumbercontainer					Maximum number of units of this product in any container.
maxsinglesidedlengthoverhang					Used for lumber loading only
maxsinglesidedwidthoverhang					Used for lumber loading only
maxsupportweight	float	0..99999	99999	N	The maximum weight this product can bear. Can be overridden by either Orientations or Configs for more precise modelling.
minimumsupportfactor	float	0..1.0	0.7	N	The part of a loaded product that needs to be supported by boxes underneath
netweight	float	0..99999	0	N	Net weight of this product
neverindoorway	boolean	true/false	N	N	Client specific
nostickingout	boolean	true/false	Y	N	Must be supported for its entire area
numitems					The number of items in the product, used only in the Manifest.
numlayers					(client specific) The number of layers on a pallet.
numpercarton					(client specific)
numperlayer					(client specific)
onfront	boolean	true/false	Y	N	Relevant only for non-rectangular products. May be loaded on its front.
onright	boolean	true/false	Y	N	Relevant only for non-rectangular products. May be loaded on its right.
pallet	boolean	true/false	false	N	Use only on advice from MagicLogic.
passthroughonly	boolean	true/false	false	N	If set to true, this product will not be loaded.
picklocation					A pass-through text string (not used by Cube-IQ, only imported and exported)
pickquantity					The quantity that needs to be kept together during loading
seatheight	float	0..99999	0	N	Relevant only for products of type "sofa"
shortsidelength	float	0..99999	0	N	Used when defining a Trapezoidal product. Please consult with MagicLogic when modelling such objects.
shortsideoffset					
sideupok	boolean	true/false	Y	N	Product can be loaded on its right side. (Width becomes height.)
specialmarking	boolean	true/false			If set true, the top of this product will be drawn with an extra marking in the Load Plans
stackcode	string	n/a	'	N	Reference to the Stacking Matrix
stackheight	float	0..99999	0	N	If not zero or blank, this value indicates that this product nests when

					stacked on itself, where each extra unit in the stack increase the overall stack height by this amount. So, the first product may have a height 10, where nesting of a next product increases the height by for example 2.5, instead of 10.
stackindex					The value of this field is only used if 'Orientations' are not imported separately
supportonlysamefootprint	boolean	true/false	N	N	Only products with exact length and width may be stacked UPON this product.
toplength	float	0.99999	0	N	Length at the top of a 'tapered' product
toplengthedgethickness	float	0.99999	0	N	The usable width of an edge in the length direction of an open top product
toponly	boolean	true/false	N	N	Specifies whether this product can have anything placed on top of it
toptype	string	Full/ Tapered/ Perimeter/ Lengthbars /Widthbars/ Corners	''	N	The type of an 'open top' product
topwidth	float	0.99999	0	N	Width at the top of a 'tapered' product
topwidthedgethickness	float	0.99999	0	N	The usable length of an edge in the width direction of an open top product
tshapelegwithoffset	float	0..99999	0	N	Used when defining a T-shaped product. Please consult with MagicLogic when modelling such objects.
tshapelegwidth					
tshapebarlength					
turnable	boolean	true/false	Y	N	Part of the group of orientation rules for the product
type	string	n/a	"box"	N	Can be "box" "cylinder" "sofa", "trapezoid", "pallet"
upsideup	boolean	true/false	Y	N	Relevant only for non-rectangular products. May be loaded upside down.
useconfigs	boolean	true/false	N	N	Overrides "useorientations" field. Set to true if you are providing configuration data for this product.
useorientations	boolean	True/false	N	N	Set to true only if you intend to use orientation data (see below)
weight	float	0.01..99999	0	Y	Product weight in the current weight units
width	float	0.01..99999	0	Y	Product width in the current units
widthextrapadding					Added to the product's width dimension when loading, to account for packaging or padding.

Static Data: Container Data

Mandatory fields are **highlighted**.

XML Syntax:

```
<containers><container ... /></containers>
```

Data item	Type	Range	Default	Required?	Information
containerid	string	n/a	"	Y	Key Field. The name of this container (e.g. 40' High Cube)
action					If 'delete', the container will be deleted from the database, if 'overwrite' the container data will be overwritten without confirmation being asked
addshelveswhenneeded	boolean				Client specific
bottomtotoploading	boolean	true/false	N	Y	Load this container from the bottom to the top. Typical for pallets.
bulkheaddepth					The size of the bulkhead in the length direction of the rail car.
bulkheadminimumdistane					The minimum distance between two 'split' bulkheads.
bulkheadpositionstep					The step between possible placement points of a bulkhead.
bulkheadsplitted					'Y' if there are two centered bulkheads, in between which nothing can be loaded, and with a variable distance.
bulkheadstartposition					The first spot for (the center of) a bulkhead from the end of the railcar.
centerload					If the container is of type 'Pallet' the whole load will be centered on the pallet.
containertype	string	Fixed	"Rectangular"	Y	Must be "Rectangular"
dblenthunit					Unit of measurement
dbvolumeunit					Unit of measurement
dbweightunit					Unit of measurement
defaultnum	integer	999	1	N	Number of this type of container available, if not overridden in the volatile data.
depth	float	0.1..99999	0	Y	Internal depth (i.e. "front to back" length) of this container (see also width and height fields)
doorheight	float	9999999	0	N	Height of entrance to container. Defaults to the maximum relevant size.

doorlength	float	9999999	0	N	Length of doorway or entrance to container. Defaults to the maximum relevant size.
doorwidth	float	9999999	0	N	Width of doorway or entrance to container. Defaults to the maximum relevant size.
dualrobotized	boolean	true/false	false	N	Customer special. Please contact MagicLogic if you are building pallets using robots.
flexibledims	boolean	true/false	false	N	Special feature. Please contact MagicLogic if you would like Cube-IQ to optimize container design.
flexzonevariationfactor	float	0..99999	0	N	Please see Zones section.
fliplayers	boolean	true/false	false	N	Alternate the layers for stability.
fulllayersonly					(Pallet) If true, Cube-IQ will only create complete layers, no partial layers
hardminfillpercentage	boolean	true/false	false	N	Minimum volume to be used in order to consider this container to be loaded
height	float	0.1..99999	Container height	Y	Internal height of this container
irregularinfrontview	boolean	true/false	false	N	Reserved.
layerslastpallet	boolean	true/false	N	N	Force even the top layer to be built as a layer, not mixed.
loadbottomfirst	boolean	true/false	N	N	If the container is not full, compact the load in the height direction. Typical for pallets.
loadingmode	Enum.	Manual PackStation Robot DualRobots MLSSpider MCSGripper			(Pallet only) The mode in which the pallet is loaded
loadinlayers	boolean	true/false	N	N	Load in a layer formation. Typically used when building pallets.
maxbulkheadaligndiff	float	0..99999	0	N	The target distance of loaded product from the closest bulkhead.
maxheight	float	0..999999	0	N	Used when estimating optimal container dimensions. Please contact MagicLogic if you would like Cube-IQ to optimize container design.
maxlength					
maxwidth					
minheight					
minlength					
minwidth					
maxlineal	float	0..999999	N	N	Maximally loadable lineal feet (lumber)

maxlengthoverhang	float	0..999999	0	N	Used when palletizing. The amount a product can overhang the front/rear edge of the pallet.
maxwidthoverhang	float	0..999999	0	N	Used when palletizing. The amount a product can overhang the left/right edge of the pallet.
maxspreadgap	float	0..999999	99999	N	(Pallet) The maximum gap that can get created between two boxes when spreading takes place
maxtotalcost	float	99999999	0	N	Total cost limit for this container when creating a load
maxweight	float	0.1..999999	999999	999999	Maximum weight capacity for this container
minbridgingsupport	float	0..99999	1.0	N	(Pallet) When a box gets loaded bridging two other boxes, the minimum distance it must be supported on each side
minfillpercentage	float	0.0	0	N	Minimum volume to be used in order to consider this container to be loaded
minutilforlayers	float	0..9999	0	N	Minimum number of products that constitute a layer.
minvolumetoload	float	0..99999	0	N	Minimum volume that must be loaded
minweighttoload	float	0..99999	0	N	Minimum weight that must be loaded
mixedlayers	boolean	true/false	Y	N	Allow mixing of products when creating layers.
nextstagestackcode	string			N	In multi-stage loading, a loaded container in a stage becomes loadable product in the next stage, and this will be its stack code
numberofbulkheads	integer	0..999	0	N	The number of bulkheads in a railcar.
outsideextraheight	float	0..99999	0	N	Any additional height for this container, such as for a lid, freezer unit etc.
outsideheight	float	0..99999	Container height	N	Outer dimensions for this container.
outsidelength	float	0..99999	Container depth	N	Outer dimensions for this container.
outsidewidth	float	0..99999	Container width	N	Outer dimensions for this container.
partialloadonfloor	boolean	true/false	N	N	If true, a partial load will be compacted in height. If false, it will be compacted in length.
pckmatchingcode	string	n/a	'	N	If specified, this container will only load products that have the identical string in their 'entnrmatchingcode' field.
robotized	boolean	true/false	N	N	Customer special. Please contact MagicLogic if you are building pallets using robots.
scac	string				SCAC
sequenceonlyfordepalletizin	boolean	true/false	false	N	If true, the product loading sequence is not taken into account in

g					the final order of the loaded products
settingsid	string	n/a	"	Y	Used to reference the "Container Settings" record (see above)
shippingcost	float	0..999999	0	N	Used if optimizing by cost.
showcentergravity	boolean	true/false	false	N	If true, the center of gravity is shown in the load plan graphics
spreadblocks	boolean	true/false	N	N	Spread products across the entire area of the container for stability.
stackingwithinlayersok	boolean	true/false	false	N	If true, layers may be built up out of stacked and non-stacked products
tareweight	float	0..99999	0	N	The weight of the empty container.
topspace	float	0..99999	0	N	Space to be left clear at the top of the container, e.g. for loading access.
treatblankasmatchingcode	boolean	true/false	N	N	If set to true, allows this container to load Products that have one or more cntnrmatchingcodes set.
twosidedloading	boolean	true/false	N	N	This container can be loaded from both sides (e.g. two-sided railcar)
type	string				Rectangular, Airline (ULD), Truck or Pallet.
useonlyifrequired	boolean	True/false	N	N	If true, the container will only be used if absolutely necessary for at least one product in the load list
volume	float	n/a	0	N	Internal volume of this container (re-calculated by the BlackBox)
volumecapperc	float	100.0	0	N	Actual volume of this container which can be used to load product (expressed as a percentage)
volumecapperciffinal	float	100.0	0	N	Actual volume of this container which can be used to load product if this is the last container which can be used. (Expressed as a percentage)
width	float	0.1..99999	n/a	Y	Internal width of this container

Static Data: Zones

Zones are used to define individual areas within a container that can have their own loading rules. Please refer to the Cube-IQ User Guide for more information. Note: Zones are the new name for Load Spaces.

Mandatory fields are **highlighted**.

XML Syntax:

```
<zones><zone ... /></zones>
```

Data item	Type	Range	Default	Required?	Information
containerid	string	n/a		Y	Key field. Must match the id of the container which holds this zone
depth	float	0.01..n		Y	overall depth (front to back)
width	float	0.01..n		Y	overall width
height	float	0.01..n		Y	overall height
depthcrd	float	0..n		Y	start location of this zone, relative to the back of the container
widthcrd	float	0..n		Y	start location, relative to the left wall of the container
heightcrd	float	0..n		Y	start location, relative to the floor
description					Description
flexible	boolean	true/false		N	Zone is allowed to shrink from its overall dimensions
maxlineal	Integer	0..99999		N	(Lumber) Maximum allowed total footage inside the zone
minimumfillperc	integer	9..99999		N	Loading in the zone will be skipped if not at least this fill percentage is reached
minnumunits	integer	0..n		N	Zone must contain at least 'n' units. Omit this field to remove the restriction.
maxnumunits	integer	minnumunit s..n		N	Zone cannot container more than 'n' units. Omit this field to remove the restriction.
maxnumproducts	integer	0 or 1		N	Client specific feature. Only one product type will be placed in this zone.
pckmatchingcode	string	string		N	A list of one more more matching codes that allows only products with the same code to be loaded in this zone.
weightlimit	float	0..n		N	Max weight capacity for this Zone

loadstart	integer	1..8		N	Loading direction for this zone, relative to the container. Values are 1 – back to front, starting at the left wall 2 – back to front, start at the right wall 3 – right to left, starting at the back 4 – right to left, starting at the front 5 – front to back, starting at the left 6 – front to back, starting at the right 7 – left to right, starting at the back 8 – left to right, starting at the front
sequence	integer	1..n		N	If the container has multiple zones, they are loading by sequence, starting with the lowest number.

Static Data: Lost Spaces (Unusable Spaces)

Lost Spaces are rectangular areas of the container which must not be used. They typically represent physical impediments such as freezer units, internal bracing etc.

All fields are mandatory.

XML Syntax:

```
<lostspaces><lostspace ... /></lostspaces>
```

Data item	Type	Range	Default	Required?	Information
containerid	string	n/a		Y	Key Field. Must match the id of the container which holds this lost space
length	float	0.01..n		Y	overall length (front to back)
width	float	0.01..n		Y	overall width
height	float	0.01..n		Y	overall height
depthcrd	float	0..n		Y	start location of this zone, relative to the back of the container
widthcrd	float	0..n		Y	start location, relative to the left wall of the container
heightcrd	float	0..n		Y	start location, relative to the floor

Static Data: Orientation Rules

Orientation rules are optional. They are used when very specific orientations are permitted / disallowed for products, allowing complex loading rules to be defined. If they are specified, there are six possible orientations per product, which should be defined as follows:

- 1 – box “as defined”
- 2 – box turned through 90 as viewed from above
- 3 – box placed on its front
- 4 - box placed on its front, and turned through 90 as viewed from above
- 5 – box placed on its side
- 6 – box placed on its side, and turned through 90 as viewed from above

The BlackBox will use any orientation values as if they had been defined as above, so it is essential that the orientations, if used, as specified correctly.

XML Syntax:

```
<orientations><orientation ... /></orientations>
```

Data item	Type	Range	Required?	Information
allowed	boolean	true/false	Y	This orientation is allowed for this product
bottomonly	boolean	true/false	Y	This orientation must be on the floor of the container
cntnrmatchingcode	string	Up to 32 chars	N	Locks this orientation to containers that have a matching code in their ‘pckmatchingcode’ field
maxinlayer	integer	0..99999	Y	Maximum number of products which can be stacked <i>in this orientation</i>
maxsupportedweight	float	0.0..9999.9	Y	Maximum weight that this orientation can support.
orientationnum	integer	1..6	Y	See orientation list above for meaning of values 1 through 6.
productid	string	n/a	Y	Identifier to tie this orientation record to a specific product type
length	float	0.01..99999	Y	overall length of this product, <i>in this orientation</i>
width	float	0.01..99999	Y	overall width of this product, <i>in this orientation</i>
height	float	0.01..99999	Y	overall height of this product, <i>in this orientation</i>
stackindex	integer	0..9999999	N	Index that refers to the appropriate stack code to use for this orientation (see stacks)
toponly	boolean	true/false	Y	This orientation must not have anything else stacked on it

Static Data: Configurations

Configurations are optional. They are used as ‘super orientations’ giving complete control over how products should be placed in containers, and allowing different positioning and containerization rules for multiples. MagicLogic can help you create configurations for your product data if required.

XML Syntax:

```
<configurations> tag,  
followed by multiple <configuration ... /> tags  
Terminated by </configurations> tag
```

Data item	Type	Range	Required?	Information
productid	string	32 char string	Y	Key field. Identifier to tie this orientation record to a specific product type
bottomonly	boolean	true/false	Y	Must be on floor. However, the product itself may get stacked up to the value of the ‘maxinlayer’ field
clampspace	float	0..99999	N	Reserved.
cntnrmatchingcode	string	32 char string	Y	Locks this config to a container which has the same string in its ‘pckmatchingcode’ field.
color				Color of the configuration. Will be given a random color if left blank, or given value 0.
configindex	integer	1..99999	Y	Unique index of this config.
flatok	boolean	true/false	Y	This config may be loaded flat (i.e. in its ‘natural’ orientation)
height	float	0.1..99999	Y	Height of the config
indoorwayonly	boolean	true/false	N	Reserved.
length	float	0.1..99999	Y	Length of the config
lengthandwidthswapped	boolean	True/false	N	If true, this configuration will only be loaded turned
lineal	float	0..99999	N	(Lumber) Lineal length in feet
maxinlayer	integer	0..99999	Y	May be stacked to a maximum of ‘n’ layers
maxsupportedweight	float	0..99999	Y	How much weight this config can support
mirroring	Integer	0,1,2,3,4	N	If the configuration represents a layer, this field represents if rotating/mirroring creates a different layout. Values are 0..4 represent: 0-No mirroring 1-Lengthwise mirroring 2-Widthwise mirroring

				3-Mirroring in both length and width, and rotate by 90 degrees 4-Rotate by 90 degrees
name	string	32 char string	Y	Unique id for this config
notonfloor	boolean	true/false	N	Must not be loaded on the floor of the container/pallet.
numlayers	integer	0..99999	N	Reserved.
numunits	integer	1..99999	Y	How many individual products are in this config
onendok	boolean	true/false	Y	May be loaded on its end
onsideok	boolean	true/false	Y	May be loaded on its side
palletized	boolean	true/false	N	Reserved.
priority	integer	0..99999	Y	Configs with a higher priority are loaded first, if possible
showpdrawnum	boolean			(client specific) Leave blank
singlestageuse	Integer	1..number of stages	N	In multi-stage loading this configuration can only be used in the stage that corresponds to the value of this field.
splitafterloading	boolean	true/false	Y	Split this config into a block of 'n' products when loaded
stackcode	string	32 char string	Y	Stack which maps to the stack matrix (see 'Stack Matrix')
stackqty	integer		Y	Reserved.
toponly	boolean	true/false	Y	Must be on the top of the load. However, the product itself may get stacked up to the value of the 'maxinlayer' field
turnable	boolean	true/false	Y	Controls whether this config can be turned 90 degrees in its vertical axis
weight	float	0.1..99999	Y	Weight of the config
width	float	0.1..99999	Y	Width of the config

Static Data: Stacking

Stacking can be controlled in several ways in Cube-IQ / BlackBox, depending upon your requirements. The first method is to use the stacking index, which can be defined either for the overall product, or for each orientation. This rule is simple: lower stack indexes stack on top of higher ones.

Secondly, as discussed here, a stacking code can be allocated to each product and/or its configs.

These codes have to be defined first in the Stacking Codes record shown below.

Data item	Type	Range	Required?	Information
seq	integer	1..999999	Y	Must be a simple incremental number for each stack code
code	string	Up to 32 character string	Y	Unique identifier for this stack code

Once you have defined your stack codes, you can create a matrix which controls how the stack codes interact. Each entry is a pair of codes, which define a positive stacking rule: topcode may be stacked on bottomcode. In this way you have total control over how every product can be stacked (or not) with every other product.

Data item	Type	Range	Required?	Information
topcode	string	32 char string	Y	These two codes form a 'stacking pair'
bottomcode	string	32 char string	Y	

Static Data: Customer Data (Cube-IQ GUI only)

Customer information is only relevant to Cube-IQ. The BlackBox system does not process this data in any way – it would not trigger an error, but will be ignored if sent as part of the input.

Data item	Type	Range	Required?	Information
name	string		Y	
address1	string		N	
address2	string		N	
address3	string		N	
city	string		N	
state	string		N	
country	string		N	
zip	string		N	
phone	string		N	
fax	string		N	
email	string		N	
notes	string		N	
contact	string		N	
www	string		N	

Volatile Data: Product to Load

Data item	Type	Range	Required?	Information
autoid	integer	integer	N	Key field. Unique identifier for this product-to-load, so that a direct database link can be made with any loaded blocks for this product (see the Block Import section).
batch	integer	1..99999	N	Sequence number. Lower batch numbers are put into the container first. Default is to give all products an equal value. Can be used to ensure that products are loaded in pick sequence for example. Reverse the values to use as a way to control the unloading sequence.
cntnrmatchingcode	string	n/a	N	Forces the optimizer to place this product ONLY in a container that has the same matching code in its 'pckmatchingcode' field. Blank means this product can be loaded into any of the containers provided in the "containertoload" elements.
descr	string	n/a	N	Overrides the default description field at the Product level
groupindex				If two products have the same groupindex, they will be loaded together, and if their quantities are the same they will be loaded in the same quantity (possibly less than the requested quantity)
loadid	string	n/a	Y	Must match the "loadid" field in the load element.
maxquantity				The maximum quantity to be loaded of the product, with the system first loading the 'Quantities' of all products, before exceeding those quantities to load up to the MaxQuantity.
maxstack				Maximum stackable quantity for this product in this load (optional override of the product data value).
numloaded				The total number loaded in all containers in this case, which is useful only when Cube-IQ would export an optimized loading case. For a new case, the value is always 0.
originalqty	integer		N	This field is used to store the original quantity, so that at a later stage the quantity changes made by the user can still be traced.
overridecost				Overrides the cost of the Product
overrideheight				Overrides the standard height of the Product
overrideweight				Overrides the weight of the product as stored in the product data.
ponumber	string		N	A text string for the PO/order
productid	string	n/a	Y	Must match an existing product id in the static data (or product element in the volatile data)
ptlrgb				The color of this product in the current load.
quantity	integer	1..999999	Y	Number of this product available to be loaded
stage	integer	1	N	If used, set to "1" only
totitems				The total target number of items to be loaded of the product, converted automatically by Cube-IQ into a quantity based on the number of items per product.

totvolume				The total target volume to be loaded of the product, converted automatically by Cube-IQ into a quantity.
totweight				The total target weight to be loaded of the product, converted automatically by Cube-IQ into a quantity.

Volatile Data: Fixed Boxes

Fixed boxes are used to indicate to the BlackBox that the following product(s) must be placed in the container exactly as stated. The BlackBox will build the remainder of the load around the fixed boxes.

It is a requirement on the caller to ensure that fixed boxes do not overlap with each other.

Data item	Type	Range	Required?	Information
productid	string	n/a	Y	Key field. Must match an existing product id in the static data (or <product> element in the volatile data)
containerseqno	integer	1..	Y	The sequence number of the loaded container
depth ¹	double	0.01 - 99999.9	Y	Length (or depth) of this product AS PLACED
width	double	0.01 - 99999.9	Y	Width of this product AS PLACED
height	double	0.01 - 99999.9	Y	Height of this product AS PLACED
lcoord	double	0 - 99999.9	Y	Length (or depth) Coordinate of the lower left corner of the product
wcoord	double	0 - 99999.9	Y	Width coordinate of the lower left corner of the product
hcoord	double	0 - 99999.9	Y	Height coordinate of the lower left corner of the product
orientation	integer	[1 - 6]	N	Orientation index (optional)
rotation	integer	[1 - 4]	N	Rotation index (optional)

¹ Identifier "length" may also be used

Volatile Data: Container to Load

Data item	Type	Range	Required?	Information
autoid				Key field.
containerid	string	n/a	Y	Identifier for this container (e.g. 20' Dry Cargo)
containernum	integer	1..99999	Y	Number of this type of container available to be loaded
loadid	string	n/a	Y	Must match the loadid field used in the load element.
numlocked				The total number of this container type that is used for loaded containers that have their loads 'locked' (= frozen, so that only the remainder can be re-optimized).
numused				The total number used of this container type, which is useful only when Cube-IQ would export an optimized loading case. For a new case, the value is always 0.
overrideheight				
sequence	integer	0..99999	Y	Order of preference for this container. Lower sequence numbers will be used first.
shippingcost	float		N	The shipping cost attached to selecting this container for loading. If missing, the system will calculate simulated costs that will (1) minimize the total number of containers used and (2) minimize the total volume of the containers used. This field is only used if the container selection rule is "minimum overall cost"
stage	integer	1..	N	The stage in which this container is available for loading. For BlackBox use, the value must be 1.

Volatile Data: Stage

Data item	Type	Range	Information
loadid			Key field.
containerselectionrule			The Container Selection Rule sets in which order the various available containers are to be used in the case of a multi-container optimization.
groupsplitting	integer	1..4	Controls how grouped items are allowed to be split over containers. The four values are: 1: Allowed 2: Only allowed if it saves using an extra container 3: Only allowed if the group is too large for a single container 4: Never allowed.
loadedlineal	float		(Lumber) The total loaded lineal feet. Output.
loadedunits	float		The total number of loaded units/colli. Output.
loadedvalue	float		The total number of loaded value. Output.
loadedvolume	float		The total number of loaded volume. Output.
loadedweight	float		The total number of loaded weight. Output.
loadfullstackonly	Boolean	True/false	If true, the system will load full stacks only, where full stack is specified by the stack count is specified by the 'maxnuminlayer' of the product/configuration
loadproportionally	boolean	true/false	If set, the system will adapt the quantities to be loaded (more or less) proportionally, until the container is full
numstagingpositions	0.. integer	0 (no limit)	Number of staging positions that the optimizer will use when building a load.
onecontainerperseqno	boolean	True/false	For every sequence number in the load list, one container will be loaded
optimizedloadedvalue			if Y, Cube-IQ will optimize for maximum loaded value, rather than loaded volume/weight
overridesettings	string		If filled, the system will use these Settings, instead of the ones specified for the containers.
sequenceiscontainertype	boolean	True/false	The product's sequence number controls into which container it is placed. (1=first container, 2=second etc)
sequencemeanspriority	boolean	True/false	If products have a sequence number assigned, the sequence is treated as meaning give priority to loading lower sequence numbers.
stablemixedpalletload	boolean	True/false	If the loading takes place on pallets, the system will try to use brick laying to

			improve stability.
stage	integer	1..	The stage in which this container is available for loading. For BlackBox use, the value must be 1.
totallineal	float		(Lumber) The total number of lineal feet to be loaded (calculated)
totalunits	float		The total number of units to be loaded (calculated)
totalvolume	float		Total volume to be loaded (calculated)
totalweight	float		Total weight to be loaded (calculated)
treatgroupsasset	boolean	true/false	If true, and the full quantities cannot be loaded, the system will load an integer number of times the greatest common divisor of the specified quantities. So, a set/group of five tables and twenty chairs may get loaded as 4 tables and 16 chairs.
usesequenceovercontainers	boolean	True/false	Please consult MagicLogic before setting this value to “True”

Output: Loading Results

Loading results are stored in an XML file which is created in the loading results folder (please refer to the BlackBox installation guide for further information)

Results from each loading case are stored in a separate file. Each file contains the following information:

System Information

BlackBox optimization results are prefaced by Cube-IQ production information:

Data item	Type	Information
systeminformation		
product	string	The product identifier – this will always be “Cube-IQ BlackBox”
version	string	The version of the BlackBox system that produced this output file
date	string	The date when these loading results were created.

Example:

```
<systeminformation product="Cube-IQ BlackBox" version="1.4.16 (Build 117)" date="18/02/2004"/>
```

Load

Data item	Type	Range	Information
load			
containerselectionrule			The Container Selection Rule sets in which order the various available containers are to be used in the case of a multi-container optimization.
currentlengthunit			Unit of measurement
currentvolumeunit			Unit of measurement
currentweightunit			Unit of measurement

	customerid	string		Ignore
	date	string	any valid date	Date when this load was created
	DBAction	integer	0..2	0 means normal processing, 1 means 'overwrite without asking confirmation', 2 means overwrite but keep input data.
	groupsplitting			Can have values 1..4, indicating under which circumstances products with the same group number can be split over multiple containers. (1) is 'Allowed', (2) means 'only if that would save a whole container', (3) is 'only for groups that are too large for one container', and (4) is 'Never'.
	loadfullstackonly			If a product or config has a Stack Quantity (sometimes called Bundle Quantity), a value of 'Y' sets that the product can only be loaded in full stacks.
	loadid	string	n/a	Id for this loading case
	loadproportionally	boolean	true/false	If set, the system will adapt the quantities to be loaded (more or less) proportionally, until the container is full
	notes	string	empty string	user notes/comments
	numstages			If is greater than zero, the optimizer can place sequenced products temporarily in a staging area, effectively overruling their given sequence.
	numstaginpositions	0..integer	0 (no limit)	Number of staging positions that the optimizer will use when building a load.
	onecontainerperseqno	boolean	True/false	For every sequence number in the load list, one container will be loaded
	optimizeloadedvalue			If Y, Cube-IQ will optimize for maximum loaded value, rather than loaded volume/weight.
	overridesettings	string		If filled, the system will use these Settings, instead of the ones specified for the containers.
	productfilter	string		to be left blank
	sequenceiscontainertype	Boolean	True/false	If Y, the product sequence numbers will be interpreted as to match the sequence in which the container types are given.
	sequencemeanspriority	boolean	True/false	If products have a sequence number assigned, the sequence is treated as meaning give priority to loading lower sequence numbers.
	shipdate	string	any valid date	The date format must be 'yyyy-mm-dd', using the '-' as delimiter.
	stablemixedpalletload	boolean	True/false	If the loading takes place on pallets, the system will try to use brick laying to improve stability.
	treatgroupsassets	boolean	true/false	If true, and the full quantities cannot be loaded, the system will load an integer number of times the greatest common divisor of the specified quantities. So, a set/group of five tables and twenty chairs may get loaded as 4 tables and 16 chairs.
	userstring1	string		Two numerical and one string field that the user can use for any (display only)

	userstring2			purpose. It can be any user defined string, for example the PO Number.
	uservalue1			
	uservalue2			
	usesequenceovercontainers			The product sequence numbers can be used for two purposes: To load in drop sequence (First-In-Last-Out) (value Y), To load in picking sequence (value N). The difference is that for drop-sequence loading, the Box Sequence Numbers are to be taken into account over all available containers, as otherwise multiple drops to one location may result. When loading in picking sequence, the box sequence must be correct for each container (usually of type 'box' or 'pallet') separately.

Example:

```

=<loads>
<load id="test2" loadid="test2" containertype="" notes="Notes:" date="30/12/1899" shipdate="30/06/2005"
  userstring1="" noflodedproducts="972" stage="1" otherstageid="" customerid="" sequenceiscontainer="True"
  containersrequired="TBA" totalitems="TBA" totalvolume="0.440354499999999" totalweight="5394" loadproportionally=" "
  productfilter="" />
</loads>
    
```

Loaded Container

A loaded container record is written for every container used in this loading solution. It describes the usage made of the current container, including volume and weight utilization, the position of the Centre of Gravity etc.

Identifier	Type	Range	Information
containerid	string	n/a	Key field. Identifying name of this container
axle1weightperc	float	0.0..100.0	Percentage of the load weight borne by axle 1
axle2weightperc	float	0.0..100.0	Percentage of the load weight borne by axle 2
axle3weightperc	float	0.0..100.0	Percentage of the load weight borne by axle 3
containerseqnr	integer	1..n	Index of the current container. Used to match the blocks (see below) to the container.

cuserstring1			A text string that is available to the user for any purpose, for examples, to attach on a message to a load. There's also a cuserstring2.
frequency	Integer	1..	The number of times that the loading of this particular container needs to be executed
heightgravitycentreperc	float		Position of this load's centre of gravity, as a percentage of the height of the container
lengthgravitycentreperc	float		Position of this load's centre of gravity, as a percentage of the length of the container
linealleft	float		(Lumber) The total number of lineal feet loaded on the left side of the container.
linealright	float		(Lumber) The total number of lineal feet loaded on the right side of the container.
loadeditems	Integer		The total number of items (inside products) loaded.
loadedlineal	float		(Lumber) The total number of loaded lineal feet.
loadedvalue	float		The total value of the products loaded in this container.
loadedvolume	double	0.0..n.n	Actual volume used by the SKUs in this container
loadedweight	double	0.0..n.n	Actual weight of all the SKUs in this container
loadid	string	n/a	Unique id of the current load
locked			'Y' indicating that the load should be left untouched when a subsequent optimization is done.
maxheight			Ignore
maxlength			Ignore
maxwidth			Ignore
numblocks	integer	1..n	Number of blocks loaded on this container
numfixed	integer	0..n	Number of products which were fixed in place before optimization
numproducts	integer	1..n	Total number of products in this container
numunits	integer	1..n	The number of units/colli loaded
physicalid			Some ID of the container that is actually used for this load.
seqno	integer	1..	The sequence number of the loaded container in the load.
stabilityindex	float	0.0..1.0	A value between 0.0 and 1.0, that gets calculated for loaded pallets. It is an indication of how stable the pallet is.
stage	integer	1..	The stage in which this container is available for loading. For

			BlackBox use, the value must be 1.
totalcontainervalue			(BlackBox only)
totalcontainerweight			(BlackBox only)
volumetricweight	float		The volumetric weight.
volumeutilization	double	0.0..100.0	Percentage of the container's volume used by the current load
weightutilization	double	0.0..100.0	Percentage of the container's weight capacity used by the current load
widthgravitycentreperc	float		Position of this load's centre of gravity, as a percentage of the width of the container

Example:

<loadedcontainers>

```
<loadedcontainer id="20' Dry Cargo" loadid="test2" containerseqnr="1" containerid="20' Dry Cargo"
  volumeutilization="1.33212113179214" weightutilization="24.8000258602711" loadedvolume="440354.499999999"
  loadedweight="5394" nofproducts="972" lengthgravitycentreperc="43.9787143446661"
  widthgravitycentreperc="6.82359068974118" heightgravitycentreperc="4.811563334303" maxlength="0" maxwidth="0"
  maxheight="0" nofblocks="28" nofdiffproducts="1" axle1weightperc="0" axle2weightperc="0" noffixed="0" idcontainer="" />
</loadedcontainers>
```

Block Information

A block is a group of one or more products of the same type, in the same orientation, and defines where and how the products are placed within the container, including position and orientation.

Identifier	Type	Range	Description
blocks			
autoid	integer	Always 0	A field that is used to link the block to the corresponding “producttoload.”
approachdepthcoord			The back-to-front co-ordinate in the container
approachheightcoord			The bottom-to-top co-ordinate in the container
approachwidthcoord			The left-to-right co-ordinate in the container
block	integer	0..n	Block index 0-based counted
cnt	integer	1..n	Block index 1-based counted
configindex			If not blank/null, the record number of the product configuration used
containerseqnr	integer	1..n	Indicates the container in which this block is to be placed
depthcoord	float	0.0..n.n	Coordinates of the front of the block, relative to the rear bottom left corner of the container
height	float	0.01..n.n	Height of the current block, measured in the system’s current units
heightcoord	float	0.0..n.n	Coordinates of the bottom of the block, relative to the rear bottom left corner of the container
layer			
length	float	0.01..n.n	Length of the current block, measured in the system’s current units
lengthspread	float	0.01..n.n	The space between units in the length direction
loadid	string	n/a	Unique name for this load
numdeep	integer	1..n	Number of SKUs in this block, counting from front to back
numhigh	integer	1..n	Number of SKUs in this block, counting from bottom to top. Therefore the total number of SKUs in the block is “numdeep * numwide *

			numhigh”
numwide	integer	1..n	Number of SKUs in this block, counting from left to right
orientationindex	integer		If orientation data has been used, indicates which of the 6 possible orientations have been used for ALL products in this block.
productid	string	n/a	Unique id of the product (SKU) which forms the current block
quantity			The number of units in the block (note that product configurations may contain multiple units)
rotation	integer	0..4	Used in conjunction with the orientation field above, specifically for L- or T-shaped objects, to indicate the angle at which this product has been turned in its current orientation. 0 – rotation not used 1 – product turned 90 clockwise 2 – product turned 180 clockwise 3 – product turned 270 clockwise 4 – product not turned
stage	integer	1..	The stage in which this container is available for loading. For BlackBox use, the value must be 1.
unitheight			Unit height as placed
unitlength			Unit length as placed
unitwidth			Unit width as placed
volume	float		The volume of the block
weight	float		The weight of the block
width	float	0.01..n.n	Width of the current block, measured in the system’s current units
widthcoord	float	0.0..n.n	Coordinates of the left edge of the block, relative to the rear bottom left corner of the container
widthspread	float	0.01..n.n	The space between units in the width direction

Example:

```

- <blocks>
- <block id="00369" loadid="test2" containerseqnr="1" block="0" productid="00369" numdeep="2" numwide="1" numhigh="8"
  length="14" width="3" height="72" depthcoord="0" widthcoord="0" heightcoord="0" topposition="2" lengthspread="0"
  widthspread="0" autoid="0" orientationindex="4" rotation="0" />

```

```
<block id="00139" loadid="test2" containerseqnr="1" block="27" productid="00139" numdeep="1" numwide="1"
  numhigh="1" length="11" width="4" height="8" depthcoord="330" widthcoord="19.25" heightcoord="17.75"
  topposition="2" lengthspread="0" widthspread="0" autoid="0" orientationindex="5" rotation="0" />
</blocks>
```


Sample Data: Input

Input: Static Data

In the following example, certain fields are **highlighted** in order to show their relationship; the container settings element has an id, which is referenced within the container element. This ensures the system uses the correct settings for each type of container.

Similarly, if orientations are to be used, the product id must match as appropriate in the product element and the orientations element. This is shown in the example **below**. Remember however that orientations are optional – they are implemented in order to support complex loading rules which may not be required. If not required, orientations can be omitted from the data.

```

_ <cubeiq>
_ <settings>
  <setting id="default" maxnumdecimals="2" maxblocksize="40" maxidlength="20" usetranslation="True"
    colorprinter="True" pagetype="" lengthunit="cm" volumeunit="m3" weightunit="kg" extraint2="0"
    usecasenotes="True" usesequenceovercontainers="True" logfile="" logfilestretched="False"
    minbottomonlyweight="0" companyname="MagicLogic Optimization Inc." registeroptionscount="0"
    reportproductsort="ByName" registrationcode="#?" featurescode=""
    manifestcode="11111111111111110000000000000000" drawasblockslimit="0" cntnrsequence="1"
    palletsupright="False" fillupdensity="0" allboxesinreoptimize="False" tippictureslocation="pics"
    showloadsetupgraphics="True" dbversion="27" donotshowtips="False" raizestyle="True" createproductid="False" />
_ </settings>
_ <containersettings>
  <containersetting id="Box" settingsid="Box" loadingmargin="0" productlengthmargin="0" productwidthmargin="0"
    maxruntime="2000" maxnonimproveiters="1000" mintoponlyspace="0.2" maxleaningdistance="0.2"
    minsupportrequired="0.6" maxsupportheightdiff="0.2" targetnumdiffboxes="0" minutildiffboxtarget="0"
    prioritylargeboxes="True" weightoptimizationfactor="0.6" lengthcogbackmargin="0" lengthcogfrontmargin="100"
    widthcogleftmargin="0" widthcogrightmargin="100" minimumblocknum="1" minproductspercontainer="1"
    unloadedproducttostage2="True" sequencemixok="True" estimatepercentage="85" overallmaxstack="9999" />
_ </containersettings>

```

```
- <containers>
- <container id="20' Box" containerid="20' Box" containertype="Rectangular" depth="590" width="230" height="227"
  maxweight="22000" volume="30.8039" settingsid="Box" backtofront="True" righttoleft="False" tareweight="2200"
  bottomtotop="False" loadbottomfirst="False" outsidelength="0" outsidewidth="0" outsideheight="0"
  outsideextraheight="0" loadinlayers="False" spreadblocks="True" mixedlayers="True" layerslastpallet="False"
  minutilforlayers="75" fliplayers="True" errorinshape="False" />
- </containers>

- <products>
- <product id="2296661" productid="2296661" length="140" width="240" height="220" weight="500"
  turnable="True" sideupok="False" endupok="False" bottomonly="False" toponly="False" maxinlayer="999"
  colour="#102#77#193#0" nofitem="1" lighterontop="False" deletewithsolution="False" attribute1=""
  attribute2="" attribute3="" attribute4="" flatok="True" flatonlyontop="False" straightupunlessontop="False"
  nostickingout="False" producttype="Box" stackcode="" backdepth="0" seatheight="0" volume="7392000" />
- </products>

- <orientations>
- <orientation productid="2296661" orientationnum="1" length="140" width="240" height="220" allowed="True"
  bottomonly="False" toponly="False" lighterontop="False" maxinlayer="0" maxsupportedweight="0"
  stackindex="0" />
- <orientation productid="2296661" orientationnum="2" length="240" width="140" height="220" allowed="True"
  bottomonly="False" toponly="False" lighterontop="False" maxinlayer="0" maxsupportedweight="0"
  stackindex="0" />
- <orientation productid="2296661" orientationnum="3" length="220" width="240" height="140" allowed="False"
  bottomonly="False" toponly="False" lighterontop="False" maxinlayer="0" maxsupportedweight="0"
  stackindex="0" />
- <orientation productid="2296661" orientationnum="4" length="240" width="220" height="140" allowed="False"
  bottomonly="False" toponly="False" lighterontop="False" maxinlayer="0" maxsupportedweight="0"
  stackindex="0" />
- <orientation productid="2296661" orientationnum="5" length="140" width="220" height="240" allowed="False"
  bottomonly="False" toponly="False" lighterontop="False" maxinlayer="0" maxsupportedweight="0"
  stackindex="0" />
```

```

    <orientation productid="2296661" orientationnum="6" length="220" width="140" height="240" allowed="False"
      bottomonly="False" toponly="False" lighterontop="False" maxinlayer="0" maxsupportedweight="0"
      stackindex="0" />
  </orientations>
</cubeiq>

```

Input: Volatile Data (Loading Case)

In the simplest sense, a loading case for the BlackBox is merely the following three elements: a *load* element, a *containerstoload* element and a *productstoload* element. This is shown in the following example. Again please note how id's are used within the elements to tie the data together – in this situation the *loadid* field is used by the system to cross-reference the data.

```

<cubeiq>
  <loads>
    <load loadid="Rolls 01" solutionid="Rolls 01" containertype="" containernum="0" notes="Notes:"
      date="30/12/1899" loadsets="False" optimizeok="False" shipdate="14/04/2003" userstring1=""
      noffixedproducts="0" noflodedproducts="" nofnotloadedproducts="" stage="1" otherstageid="" customerid=""
      inuse="False" sequenceiscontainer="False" containersrequired="" totalitems="" totalvolume="0"
      totalweight="0" loadproportionally="False" containerfilter="" productfilter="" />
  </loads>

  <containerstoload>
    <containertoload id="20' Dry Cargo" loadid="Rolls 01" auto="0" containerid="20' Dry Cargo"
      containernum="6" sequence="1" numused="0" />
  </containerstoload>

  <productstoload>
    <producttoload id="Paper 01" loadid="Rolls 01" auto="0" productid="Paper 01" quantity="123" batch="1"
      numloaded="" inthiscontainer="" noffixed="0" quantityfixed="False" orderid="" descr="" />
  </productstoload>
</cubeiq>

```

Sample Data: Output

The following output file is a simple example of a BlackBox results file. Note that the layout has been spaced for readability.

```
<cubeiq>
  <systeminformation product="Cube-IQ Black Box" version="x.x.x (Build xx)" date="YYYY-MM-DD" />
</cubeiq>
E:\1 Loading\Documents\BlackBox\C:\Cube-IQ\iCube-IQ\output\tim_testbb.xml - %23 <loads>
  <load loadid="New Load" containertype="" containernum="0" notes="Notes:" date="30/12/1899"
  loadsets="False" optimizeok="False" shipdate="28/11/2004" userstring1="" nofixedproducts="0"
  nofloatedproducts="0" nofnotloadedproducts="0" stage="1" otherstageid="" customerid="" inuse="True"
  sequenceiscontainer="True" containersrequired="TBA" totalitems="TBA" totalvolume="29.837"
  totalweight="407.207112401343" loadproportionally="True" containerfilter="" productfilter="" />
</loads>
<loadedcontainers>
  <loadedcontainer id="20' Dry Cargo" loadid="New Load" containerseqnr="1" containerid="20' Dry Cargo"
  volumeutilization="70.1976949464873" weightutilization="0.386207299334402"
  loadedvolume="1416055.98153852" loadedweight="185.188300248" nofproducts="84"
  lengthgravitycentreperc="45.6469076803361" widthgravitycentreperc="45.8304187475177"
  heightgravitycentreperc="49.0879466106102" maxlength="226.377944946289"
  maxwidth="90.5511779785156" maxheight="88.5826740264893" nofblocks="6" nofdiffproducts="3"
  axle1weightperc="0" axle2weightperc="0" nofixed="0" loadedinlayers="0" idcontainer="" />
</loadedcontainers>
<loadedproducts>
  <block id="m03" loadid="New Load" containerseqnr="1" block="0" productid="m03" numdeep="9"
  numwide="2" numhigh="1" length="212.598419189453" width="90.5511779785156"
  height="23.6220474243164" depthcoord="0" widthcoord="0" heightcoord="0" topposition="2"
  lengthspread="0" widthspread="0" autoid="0" orientationindex="4" rotation="0" />
  <block id="m03" loadid="New Load" containerseqnr="1" block="1" productid="m03" numdeep="5"
  numwide="2" numhigh="1" length="226.377944946289" width="47.2440948486328"
  height="23.6220474243164" depthcoord="0" widthcoord="0" heightcoord="23.6220474243164"
  topposition="1" lengthspread="0" widthspread="0" autoid="0" orientationindex="3" rotation="0" />
</loadedproducts>
</loadedproducts>
```

```
<block id="m01" loadid="New Load" containerseqnr="1" block="2" productid="m01" numdeep="1"
numwide="2" numhigh="1" length="19.6850395202637" width="27.5590553283691"
height="49.2125968933105" depthcoord="0" widthcoord="49.2125968933105"
heightcoord="23.6220474243164" topposition="0" lengthspread="0" widthspread="0" autoid="0"
orientationindex="2" rotation="0" />
```

```
<block id="m01" loadid="New Load" containerseqnr="1" block="3" productid="m01" numdeep="11"
numwide="1" numhigh="2" length="216.535461425781" width="49.2125968933105"
height="27.5590553283691" depthcoord="0" widthcoord="0" heightcoord="47.2440948486328"
topposition="2" lengthspread="0" widthspread="0" autoid="0" orientationindex="4" rotation="0" />
```

```
<block id="m01" loadid="New Load" containerseqnr="1" block="4" productid="m01" numdeep="2"
numwide="2" numhigh="1" length="98.4251937866211" width="39.3700790405273"
height="13.7795276641846" depthcoord="0" widthcoord="0" heightcoord="74.8031463623047"
topposition="1" lengthspread="0" widthspread="0" autoid="0" orientationindex="3" rotation="0" />
```

```
<block id="m04" loadid="New Load" containerseqnr="1" block="5" productid="m04" numdeep="7"
numwide="2" numhigh="2" length="192.913375854492" width="27.5590553283691"
height="62.9921264648438" depthcoord="19.6850395202637" widthcoord="49.2125968933105"
heightcoord="23.6220474243164" topposition="0" lengthspread="0" widthspread="0" autoid="0"
orientationindex="2" rotation="0" />
```

```
</loadedproducts>
```

```
</cubeiq>
```