#### Braiins Report Object Export Import Documentation

20 September 2012

#### Braiins Report Object (BRO) Import Export Purpose

The Braiins Admin Utils BROs Import and Export modules allow Braiins Report Objects to be maintained in a spreadsheet.

The current versions are for UK GAAP Taxonomy based Bros.

The current export file or Bros Spreadsheet (SS) forms the “Master” definition of Bros.

#### Braiins Report Objects

Braiins Report Objects (BROs or Bros) are the means by which Braiins entity data is stored for up to 4 years, accessed, summed, and output. Bro information is also used to perform iXBRL tagging.

Bros may be taxonomy based or not. Taxonomy based Bros encompass taxonomy element, dimension, and tuple information.

Bros are referenced by name, optionally with dimension member references and a tuple reference following the name. Full details are given in the “Bro Reference” section below.

Bros are organised in a multi-branch tree structure using Sets (or nodes) and Elements. An element is always a member of a set i.e. it always has a parent. A member of a set can be another set and this can continue to any depth or level, though currently Bros export and import only allow for up to 9 levels.

A Bro holds one value per year for the so called ‘Base’ Bro (the Bro referenced without any dimension or tuple references), and additional values for each dimension member reference value that has been posted.

Slave Bros may be used to replicate Bro information to another branch of the Bro Tree to enable natural and complete tree construction. Slave Bros are described in detail in a separate “Slave Bros” section.

Money values are stored as debit (+ve) or credit (-ve) balances.

Both sets and elements hold data, with a set often but not necessarily holding the sum of its children.

*Posting Bro Uniqueness*

Any possible posting must be unambiguous, thus posting Bros must be unique. (Slave Bros are not unique as they are a copy of another Bro but they are always non-posting.) Usually this is achieved by a Bro having a unique Hypercube, Taxonomy Element, and Tuple combination, but this can be extended as discussed further in the “Duplicate Hypercube and Taxonomy Element Bros”.

*Set and Set Member Posting Mutual Exclusivity*

Sets and Set Members are mutually exclusive re posting i.e. a Bro (whether set or element) cannot be posted to if any ancestor has been posted to, and a Set cannot be posted to if any ancestor or descendant has been posted to.

*Dimension Member Posting Mutual Exclusivity*

Posting to a Bro with and without dimension use are mutually exclusive.

*Automatic Summing*

Automatic summing is performed for Summing Type Bros (those of type Money, Integer, Decimal, and Share), both within a given Bro where the Base value for the Bro is the sum of the values with a dimension reference, and then up the Bro tree via the Sets (for both base and dimensional values, subject to allowable dimension limitations) unless stopped via a ‘No Sum Up’ property.

#### Terms to Describe Bros

Bros are described using various terms from the following list, sometimes prefixed with ‘non-‘. More detail on some of these terms is given in the Spreadsheet Columns table below.

| **Term** | **Comment** |
| --- | --- |
| Set | A Bro which is a Bro tree node. Sets usually, but not necessarily, have children which can be Sets or Elements. |
| Element or Ele | A stand-alone Bro which is a member of a Set. |
| Taxonomy Based  (or Tx based) | A Bro which is based on (directly linked to) a Taxonomy concrete element. Tx based Bros generate iXBRL tags when used via the report Generator. |
| Non-Taxonomy Based  (or Non-Tx based) | A Bro which is not taxonomy based i.e. which has no taxonomy link. Such Bros may take part in automatic summing and be used by the Report Generator but will not generate an iXBRL tag. |
| By Data Type | Bros have a Data Type and sometimes the data type is used to describe a Bro e.g. a Money Bro. |
| Summing | Bros with Data Types of Money, Integer, Decimal, and Share are automatically summed by Braiins, and so are called “Summing Bros”. |
| RO or Report Only | An RO or Report Only Bro can only be used for reporting purposes and cannot be posted to i.e. it is a Non-Posting Bro. Element Bros which are not Slaves cannot be RO as otherwise there would be no way for to get data into them. |
| Non-Posting | Another way of saying ‘Report Only’ |
| Posting | A Bro which can be used for Posting, and is not Report Only |
| Instant | A Taxonomy based Bro with Instant period. |
| Duration | A Taxonomy based Bro with Duration period. |
| StartEnd | A Taxonomy based Instant Bro which has StartEnd Period values. |
| Tuple | Taxonomy based Bros can be a member of a tuple as per the [Tuples List](http://admin.braiins.com/utils/TuplesList.php) and the Bro TupId property. Tuple TxIds are never used directly as the TxId of a Bro. |
| Standard | A standard or normal Bro which is not a Braiins Dimension Map. |
| Main | The term Main Bro is used to refer to a Standard Bro which has Braiins Dimension Map “children”. |
| Braiins Dimension Map or BD Map | A Braiins Dimension Map or BD Map provides mapping to a different taxonomy element or TxId for Braiins Dimension Members used with a Main Bro.  Braiins Dimension Members may also be used without mapping being involved either with standard (non-Main) Bros, or with Main Bros for a different Braiins Dimension Member from those with a Map specified.  A Bro may use only one Braiins Dimension i.e. Maps for a given Main Bro must be for just one of the Braiins Dimensions. If the hypercube for a Bro includes two Braiins Dimensions e.g. hypercube 13, OpActivities, then the Braiins Dimension which is not required must be excluded from the Bro.  Maps cannot include a TxId that is a StartEnd type  Maps cannot be a Slave. They can be a Master.  The hypercube (HyId), Excl Dims, DiMes, Check, and Descr properties default to the Main Bro values but can be varied.  The Name 0-8 segments should be the same as for the Main Bro.  All other properties (columns) have no significance to BD Maps, and are not used by Import, apart from generating errors in some egregious cases e.g. StartEnd used with a map.  BD Maps must come immediately after their Main Bro in the SS. |
| Slave | A Slave Bro is a Tx based Bro which replicates the values (potentially for an HyId subset) of another Bro, known as its Master Bro, in a different location in the Bro tree, to facilitate natural tree structures, and summing. See the following “Slave Bros” section for more about Slaves. |
| Master | A Master Bro is a Tx based Bro which has one or more Slave Bros replicating its values. It can be a Set, an Element, a Main Bro, or a BD Map. |

Bros may also be described in terms of any of their other properties, as listed in the Spreadsheet Columns table below.

#### Slave Bros

A Slave Bro is a Bro which replicates or mirrors (except for a Summing Set Slave – see below) the values (potentially for an HyId subset) of another Bro, known as its Master Bro, in a different location in the Bro tree, to facilitate natural tree structures, and summing.

The replication may be for the values of a prior year - 1, 2, or 3 years backwards in time from the current year. This allows cross year figures to be introduced into Bros for start/end and movement purposes. Such Bros are called Prior Year Slaves. A Money Prior Year Slave must have a Post Type of Sch as such cross year figures cannot be DE. Bros Import sets Money Prior Year Slave to Sch if not input as Sch.

If a Slave Bro has a TxId (it doesn’t have to), it must be the same as the TxId of its Master. If no TxId is input for a Slave, it inherits its Master’s TxId, if any.

If a Slave Bro has an HyId (it doesn’t have to), it must be either the same as the HyId of its Master, or a subset of the Master’s HyId. If no HyId is input for a Slave, it inherits its Master’s HyId, if any.

If a Slave Bro has a TupId (it doesn’t have to), it must be the same as the TupId of its Master. If no TupId is input for a Slave, it inherits its Master’s TupId, if any.

A Slave Bro is always RO as slaves cannot be posted to directly.

A Slave cannot also be a Master, though a Master can have multiple Slaves.

If the Master is a Start/End Bro only the End values are replicated to the Slave. If Start values are needed these may be obtained in another Bro defined as a Prior Year Slave.

Master values are not copied to a Summing Set Slave. Instead the set is summed as for any other set and the result of the summing is checked for equivalence with the Master via an ‘Equal To, Either Master Bro Name’ Check which is automatically added to the Bro by Bros Import.

Slave properties which can differ from those of the Master are:

* Level
* Type (Set or Ele)
* Name (must be different)
* ShortName (must be different)
* Ref
* HyId (can be a subset of the Master’s)
* Acct Type
* Post Type (cannot be DE if the Master is Sch but can be Sch for a DE Master. Must be Sch for a Money Prior Year Slave.)
* RO (always set for the Slave)
* SumUp
* Check
* Order
* Zones
* Descr
* Comment

Scratch

Unused properties for Slaves are:

* ExclDims
* InclDims
* DiMes

Start/End (Slaves replicate End values)

All other properties inherit the Master’s values, if set for the Master:

* Data Type
* TxId
* TupId
* SignN
* Except
* Amort

Context

During Bros Import Slave Bros can be tied or linked to their Master Bros either explicitly by naming the Master Bro, or they may be matched automatically by Import on the basis of TxId.HyId{.TupId} matching.

Prior Year Slaves and Slaves that are Sets with multiple HyIds must use the explicit naming of the Master method.

#### Duplicate Hypercube and Taxonomy Element Bros

Two posting taxonomy based Bros can have the same Hypercube, Taxonomy Element, and Tuple (HyId, TxId, TupId) combination if some further property serves to distinguish them such that postings to the two Bros cannot be the same. This is the case if one of the Bros has a mandatory dimension reference, whilst the other Bro does not allow use of that dimension reference.

The possible ways are:

|  |  |
| --- | --- |
| Mandatory Dimension | Bro A has a mandatory dimension courtesy of its hypercube including one of the dimensions which require an ‘M#’ or mandatory dimension member, as shown in the Dimensions Map, These dimensions are IFAclasses, TFAclasses, FAIholdings, Officers (with exceptions), and TPAtype or 9, 10, 12, 29, 34.  Bro B excludes use of that dimension via its Excl Dims property. |
| Mandatory Dimension Members | Bro A has one or more mandatory dimension members defined in its DiMes property. These can be but do not need to be ‘M#’ type dimension members.  Bro B excludes use of those dimension members via either:   * its Excl Dims property * having different mandatory dimension members defined in its DiMes property * having Bro A’s mandatory dimension members excluded in its DiMes property |

#### Bro Reference

In formats and when importing/posting, Bros are referenced by name, optionally with dimension member references, end/start property, and a tuple reference following the name, as follows:

BroName{:DimRef...}{:<end|start>}{:T{.#}}

where

BroName is a full Bro name, including dots between level sections, or a Bro Shortname

The optional dimension member references, end/start property, and tuple reference {:DimRef...}{:<end|start>}{:T{.#}} are colon separated and may come in any order i.e. even with {:<end|start>} or {:T{.#}} in the middle of a series of dimension member references.

DimRef is a dimension reference as per the [Dimensions Map](http://admin.braiins.com/utils/DimensionsMap.php) "Dimension{.Dimension Member} Reference" column, which is DimensionShortName{.DimensionMemberShortName} with the ".DimensionMemberShortName" optional for the default.  
  
The DimensionShortName part can be omitted if there is a ".DimensionMemberShortName" section and if the Bro is a member of a Set and any Set level name (in left to right order) matches a DimensionShortName e.g.  
Officers.Properties:.Director1:.Chairman  
rather than  
Officers.Properties:Officers.Director1:Officers.Chairman  
  
There can be any number of colon separated DimRefs in any order.   
  
Duplicate dimension references are silently ignored when parsing a Bro Reference.  
  
Dimension references are converted to DiMeIds for DB storage and processing.  
  
Validity checks are applied to DimRef uses e.g. versus the Bro's allowable dimensions, potentially as modified for global options, etc.  
  
Mux checks also apply.

:<start|end>  
:start or :end can be used to specify start or end (opening or closing) period for SumEnd, PostEnd, or Acc type StartEnd Instant Bros, and only for such Bros.  
  
The default is end, so use of :end is optional.  
  
start for posting is only applicable for the first year of data in Braiins i.e. for a year without a prior year for Braiins to obtain the start value from. If used when posting to a year which does have prior year values, the value is ignored. Start for reading via the RG can be used in any year.   
  
end is RO for SumEnd Bros

:T{.#}  
is an optional tuple instance where

* {.#} is an optional instance number from 1 upwards to refer to a specific tuple instance.
* A tuple instance should not be used in an import unless to update a previous tuple in an Update type import, as a new sequential instance number is allocated automatically otherwise.  
    
  (When importing/posting, a BroRef for a Tuple Bro without a tuple instance means: 'use the next instance #'.)
* In reports, a BroRef for a summing Tuple Bro without a tuple instance means: 'the sum of the tuple instances'. For a non-summing (string) Tuple Bro it gives an empty (or false) result. Or should it mean a list of all the tuple instances? djh??
* The combination of a Bro's TxId and its TupId identify a specific Tuple Member as per the Tuples List, with a unique TuMeId that is used internally by Braiins. The TuMeId never needs to be specified for Bros Import or Data Importing/Posting.

#### Import

The Braiins Admin Utils Bros Import module takes data from a Bros spreadsheet (while ignoring ‘I’ or Info columns), to the Bros database tables, replacing all previous database entries.

Bros are imported in the order in which they appear in the SS.

Import first checks the SS data for validity, and does not rebuild the Bros database tables unless no errors are found.

The Braiins Admin Utils module “Build Bro & Zone Structs” (BuildStructs.php) should be run after an Import to create the Bro and Zone “Structs” used internally by Braiins during Data Importing and generation of accounts. It is not necessary to run Build Structs if just doing a series of Bro Imports and Exports.

*Hint:* When importing it is only necessary to upload SS columns as far as Scratch since the following I columns are ignored by Import. Doing this saves some bandwidth and upload time.

#### Export

The Braiins Admin Utils Bros Export module creates a tab delimited text file Admin/www/utils/tmp/BrosUK-Gaap-yyy-mm-dd\_HH\_MM.txt, with a link to the file, of all Bros in the Braiins UK-GAAP database, for all the columns listed in the Spreadsheet Columns table below.

Export cleans up the SS by leaving columns blank where Import can deduce a value from previous row values.

The export file may be imported into Excel using just Ctrl A, Copy, Paste.

#### Editing

All columns in the SS may be edited subject to the restrictions described in the following table of the spreadsheet columns, though editing ‘I’ or Info columns without a background colour achieves nothing as Import ignores these columns.

Bros may be deleted just by removing the row, though if the Bro is a Set, all Set members (children) will also need to be deleted.

Bros may be moved if due attention is paid to Sets, both children and parents.

A new Bro may be added by inserting a row into the SS and completing just the relevant columns without bothering about Info columns.

For Braiins Dimension Maps it is OK to fill in fields where the main Bro value is used and which are blank after an export e.g. when copying changes across rows, as Import skips these fields. They will revert to blank on the next export.

#### The Spreadsheet Columns

The Columns in a Bros SS used for Editing and then Importing are described in this table, with comments for inserting/editing guidance.

Empty SS rows are skipped, and are preserved for a subsequent Export.

Rows may be part complete if sufficient information is provided.

Anything after the Scratch column is ignored by Import. (But will be re-generated by an Export.)

Whole rows after the first headings row may be commented out by starting them with ‘#’, ‘;’ or ‘//’. Import will skip such rows but preserve them (with column detail kept) for a subsequent Export. This can be useful to add documentation about the SS, and during development to skip a Bro if something is missing e.g. if a Check Bro or Slave Master hasn’t been defined yet. (Any Bro can also have comments at5tached to it via the Comments and Scratch columns.)

The Type Characters and background colours mean:

E **E**ditable as described in Comments for non-BD Map Bros, used by Import

B As for E but also applicable to **B**D Maps, possibly with some special conditions

D As for E but only applicable to B**D** Maps

T As for E for **T**axonomy based Bros or Main Bros, not applicable to BD Maps

S Applicable only to a **S**lave Bro

I **I**nformation purposes only, not used by Import. These columns have no background colour.

| **Name** | **Type** | **Comments** |
| --- | --- | --- |
| Id | I E | The Id in this column produced by Bros Export is the Id of the Bro in the DB table BROs.  The Id column is not normally used by Bros Import, so mostly this is an ‘Info’ column.  However, the column can optionally be used as an ‘E’ column to Set or start a new Id sequence or range. This is achieved by entering the desired Id into the column prefaced by ‘=’ e.g.  =3000  This is interpreted as a command to Set the Id for the Bro on this row to the number given, as the start of a new grouping within the Bro Tree. Subsequent Ids will increment from this number.  **To use this feature** t**he ‘=’ must be visible in the SS** i.e. the cell value needs to be text. Bros Export achieves this by outputting a space before the ‘=’. The number for the Id must be 4 digits long in the range 1000 to 9950.  Changing the Id in this manner is purely for human legibility purposes. It serves no DB or summing purpose.  Data is stored within the Braiins DB using Bro Ids. Thus once live data is being stored, Bro Ids can only be changed with a DB update.  Other than if the ‘=’ option is being used, there is no need to edit Ids, or keep them consistent. If a Bro is deleted there is no need to renumber the Ids of following Bros.  The Ids of new Bros created in the SS can be left blank, or any ‘code’ can be used. |
| Type | E | The possible Types are:   * Set * Ele (meaning Element) * Set Main * Ele Main * Set Map * Ele Map * Set Slave * Ele Slave * Set Master * Ele Master * Set Map Master   Ele Map Master  Export fills in complete values here, but when creating/editing Bros the only ones which really matter are ‘Set’ and ‘Ele; as Import works out all the others, or there is another way of specifying them. Even ‘Set’ is not critical, as if the names of following rows indicate that an ‘Ele’ should really be a Set, Import converts it from Ele to Set.  A Set may hold Sets and Elements. An Element must always be a member of a Set. All Level 0 Bros must be Sets.  The type of a Bro may be changed if the Name is adjusted appropriately. |
| Level | I | Level starts at 0 for a top level Bro and increases by 1 for each Set level.  Level corresponds to the number of “.”s in the Bro’s full name or the number of segments in the name less 1.  Level is an Info item, as Import uses the Bro name as the primary source of “level” information. |
| Bro Name | I | The Bro name column shows the full name of a Bro for Info purposes. Bro names consist of dot separated segments or sections corresponding to set or level names. Bro names also appear in the following columns, split into segments. Any edits should be performed there.  A Bro’s name defines a Bro’s place in the Bro tree. It is used in formats and imports, but is not intended to be seen or used via the public Braiins site.  Export prefixes Bro names with 2 spaces per level for visual purposes. These spaces are not part of the actual names.  Bro names must be unique.  Bro names must start with a letter. After that any alphanumeric character plus \_ (underscore) may be used. Segment names after the first may start with a digit or be purely numeric. The size limit is 300 characters including dots.  In the case of BD Maps, Export follows the name by the dimension reference e.g. PL.Expenses.BadDebts:Function.Admin for information purposes. |
| Name 0-8 | B | Bro Name segments by level from level 0 to level 8.  Bro names may be edited via these columns, provided care is taken to change all set member names appropriately if a set name is changed.  Optionally a b blank value may be used to repeat the value of the column in an ancestor row if defined. Using this feature makes trees easier to read and visualise in the SS. Export outputs name segments in this form, even if they were specified in full in the import.  A Bro may be moved in the Bro tree by moving it in the SS and editing its name accordingly. If such a move is performed, the Id need not be changed, but can be.  For Taxonomy based Bros the "magic' word 'Tx' (case insensitive) may be used as the last name segment. This tells Bros Import to generate a Bro name segment from the taxonomy name programmatically, using the function BroNameFromTxName() in Admin/www/utils/UtilsFuncs.inc. This strips invalid characters and shortens it by removing noise words and abbreviating common words or phrases, as well as removing camel case words which appear in prior segments of the name.  ‘Tx’ may also be used for sections of name before the last one, which correspond to a previous Set final name segment that was created via use of ‘Tx’.  An example of the use of Tx for name creation is:  Bro1 name: My0.My1.My2.Tx  Bro2 name: My0.My1.My2.Tx.Tx  Bro3 name: My0.My1.My2.Tx.MyLevel4SetName  Bro4 name: My0.My1.My2.Tx.MyLevel4SetName.Tx  where Bros 1, 2, and 4 are taxonomy based. Bro 3 could be but need not be as far as name generation is concerned.  In this example Bro 1 must be a set for the Bro 2 and Bro 4 usage of two ‘Tx’s. For Bros 2 and 4 the first Tx becomes the Tx generated final segment name from Bro 1, with the second Tx being generated from the Bro 2 and 4 Tx Names. Bro 3 does not involve any new Tx name generation. |
| BD | D | Braiins Dimension member short name used with Braiins Dimension Maps. Allowable values are:   |  |  | | --- | --- | | Admin | For Braiins Dimension Function | | CoS | | Distrib | | All | For Braiins Dimension Ageing | | <1 | | >1 | | 1-5 | | 1-2 | | 2-5 | | >5 |   Maps need only be created for the dimension members required i.e. unwanted ones can be skipped. Those that are used should be in the above order which corresponds to the DiMeId order of the Dimension Map.  Maps for a given Bro must be for just one of the Braiins Dimensions, as a Bro can use only one Braiins Dimension. |
| ShortName | B | The ShortName for a Bro is an optional shorter form of the Bro name to be used everywhere a Bro name can be is referenced i.e. in posting selection screens, in imports, in trial balances and data trail reports, and in formats.  A ShortName is not a level name. It is a replacement for the full Bro name.  If used in a Bro Map, the ShortName is a replacement for both the Bro name and the Braiins Dimension reference.  No parent.child (set.member) rules are enforced in ShortNames so a ShortName for a Bro need not reflect the Bro tree structure. Clearly care should be taken with this to avoid causing confusion, especially when Bros are presented in a tree structure, where it could appear as if the full name was the short names joined together, giving a name longer than the full name.  The maximum length of a ShortName is 48 characters.  If no ShortName is defined for a Bro, the full name, including dots between level section names, applies.  Bro ShortNames must start with a letter. After that any alphanumeric character plus ‘\_’ (underscore) but not ‘.’ may be used.  Bro ShortNames must be unique vs all other short and full Bro names. |
| Master | S | The Master column is only relevant to Slave Bros. The format of the column is: <Match | {Year#}{ MasterId} Master BroName> where:  The single word Match can be entered as an instruction to Import to search for a Master based on a TxId, HyId, and if applicable, TupId match. Any other Bros marked as ‘Slave either by having ‘ Slave’ as part of their type column, or by also having a value in their Master column, will not be considered when searching for a master match. The only time that automatic matching cannot be used is for a Prior Year Slave or a Set Slave with multiple Hypercubes. For such Slaves the Master must be explicitly defined.  Appending ‘ Slave’ to a Bro Type and leaving this column blank achieves the same thing as putting ‘Match’ here i.e. it tells Import to search for a matching Master.  The HyId used for matching can be a subset HyId of the Master’s HyId.  The alternative form for this column of: {Year#}{ MasterId} Master BroName is used to explicitly define a Master, and is the form generated by Bros Export.  The optional Year# argument defines the Bro as a Prior Year Slave. # must be 1, 2, or 3 meaning 1, 2, or 3 years before the current year.  The Master BroId argument is inserted into the column by Bros Export for info purposes. It is ignored during Import.  The Master Bro Name argument is inserted by Bros Export. It is used for explicit Master matching on Import. |
| Ref | B | Ref is an optional free form text field that an Agent is to be able to edit, which is intended to serve as a hint or guide (or reference) as to which NL code or code range, or other item e.g. Memo code or ReportPad name in the case of SAPA, would or could apply to the Bro.  This information will be used in import mapping, and in directing a user to the appropriate part of the Bro Tree during on screen posting.  Initially the field is to be used for SAPA values, but a master option by Agent will be added when support for other originating AP systems is added.  Initially the field will consist of just one section that can contain anything, like the Comment field.  However, the field could be extended to containing multiple sections re dimensions, as well as being specific about NL codes versus Memo codes versus ReportPad names (in the SAPA case), if it becomes clear that this would be useful.  The maximum length of a Ref field is 48 characters. This is arbitrary and can be increased if needed. |
| TxId | B | The TxId column can contain either the Id of a concrete element for a taxonomy based Bro, or nothing for a non-taxonomy based Bro.  If a TxId is specified, it must:   * be the Id of a concrete element in the DB table Elements with substitution group of Item * be unique, or: * the same as the TxId of another Bro if it has hypercube that is not a sub set of the other Bro’s hypercube. * For a Slave Bro the TxId is the same as its Master’s TxId   If TxId is specified, this sets the value of many other columns. |
| HyId | B | HyId is the hypercube Id or, in the case of a Set it can be a comma separated list of more than one HyId. It is a required field for a taxonomy based Bro, but can also be used with Non-Tx Bros.  For an Element Bro, the HyId must be one of the Ids in the ‘I tx Hys’ column. (A taxonomy element can have multiple hypercubes.)  For a Set the HyId can be one or all of the I Tx Hys HyIds e.g. 17,18 for a Bro based on TxId 506 is OK. Braiins uses all of such hypercubes with subsets eliminated, to determine the allowable dims for the Bro.  The HyId(s) of a Bro define the full list dimensions that can potentially be used with the Bro, as per the Braiins Admin Utils [Hypercubes List](http://admin.braiins.com/utils/HypercubesList.php), though that list can be amended for a particular Bro via the Excl Dims, Incl Dims, and DiMes Bro properties.  The allowable Dims for a Bro derived from its HyId, Excl Dims, and Incl Dims options are shown in the I Allowable Dims column.  HyId for a BD Map can differ from the main Bro value. If the same it may be left blank. |
| TupId | E | TupId is the Tuple Id as per TuplesList, if the Bro is Tx based and its TxId is a tuple member.  The TupId of a Bro is used during data importing/posting and by the RG when generating iXBRL tuple tags.  The TupId also plays a role in defining the uniqueness of a Bro. (Only one Bro can have a given TxId.HyId.TupId combination unless it is a Slave Bro.) These three items are used when matching Slave and Master Bros.  TupId does not need to be supplied for Bros where the TxId is a member of only one Tuple, which is the case for most Tuple members. Bros Import fills in the TupId property automatically for such Bros. If a TupId is supplied for such a Bro, Import checks that it is correct.  Three tuple members that are members of multiple tuples (TxIds 3237, 3382, 5186) can be linked to a specific tuple (TupId) on the basis of the HyId in use. Import processes these automatically as for the single tuple member case above.  Six other tuple members that are members of multiple tuples (TxIds 1324, 1448, 1563, 1608, 3878, 4617) cannot be linked to a specific tuple by Import so for these the required TupId must be supplied. It isn’t, Import reports an error. |
| Data Type | E | All Standard Bros have a Data Type, one of:   * None which is only applicable to a Set Bro that is just a holder or tree node, which will not ever be output itself. Such Bros cannot hold data, cannot be posted to, and cannot have TxId, HyId or related properties, other than Context. * String * Integer * Money * Decimal * Date * Boolean * Enum = one of a set of integer values * Share * PerShare * Percent   The Data Type specifies how values are stored in the DB, whether the Bro is a Summing Bro, whether the Money properties apply or not, how the Bro may be used in RG expressions, how values are formatted for inclusion in reports, and how the iXBRL tag (If any) is constructed.  The taxonomy uses more types than are listed above, but other taxonomy types are mapped to one of the above as described for the ‘I Tx Type' column.  The Data Type of a Tx based Bro is normally set by the taxonomy type, but to allow for exceptional cases, or errors, a Bro’s Data Type can be set to be different from the value derived from the taxonomy. Import issues a warning if the Bro’s Data Type is different from that derived from the taxonomy. |
| Sign | E | Sign applies to Money Bros. Allowable values are ‘Debit’, ‘Credit’, and ‘Either’. ‘Either’ applies when a Bro is just as likely to have a Credit balance as a Debit one, and corresponds to the Taxonomy ‘Not Set’ value, with the difference that a specific choice of ‘Either’ has been made rather than nothing.  For taxonomy based Bros with a sign in the ‘I Tx Sign’ column, this sign will usually be the same, but it can be different. Import issues a warning if the taxonomy value is set i.e. is not ‘Not set’ and the Bro value is different.  This sign property is used by the report generator in conjunction with the sign of the balance, any Zone sign setting, report generator row/cell sign settings, and brackets in title text, to decide on whether balances are to be bracketed and whether any iXBRL tag is to include a sign attribute. See ‘Table Balance Sign Handling’ in RGlanguage.txt for details. |
| Account Types | E | All Standard Bros must have at least one Account Type, and can have up to 4 Accounts Types in a comma separated list, chosen from:   * BS = Balance Sheet * CF = Cash Flow * PL = P&L * DetailedPL = Detailed P&L * HistoricalPL = Historical P&L * STRGL * Notes * Info * Other   The Account Type or Types define the accounting use or uses of the Bro, though as of 22 Nov 2011 no use is made of this property. It could potentially be used in conjunction with DataType, Report Only, Post Type, Allowable Dims, and Zones to control posting and Bro use. Though does it add anything not covered by Zones and Post Type?? |
| Post Type | E | Post Type applies to Money Bros.  Allowable values are:   * Nothing defaults to DE on import * DE meaning Double Entry or ”Chart of Accounts” type posting required * Sch meaning that Single Entry Schedule type posting is required |
| RO | E | RO or Report Only applies to standard and Slave Bros of any data type.  Allowable values are:   * Nothing meaning Posting is allowed   RO meaning Report Only or Non-Posting Bro  Set Bros with multiple hypercubes are automatically set to be RO. (They are expected to be used only for summing and tree structure purposes, never posting.)  Slave Bros are automatically set to be RO. (Slaves mirror their Master’s values so cannot be posted to directly.)  Standard Element Bros cannot be set to RO as otherwise there would be no way of getting data into the Bro. Slave Element Bros can be RO. |
| Excl Dims | B | For Bros or Maps, whether Tx based or not, but with an HyId, Excl Dims is an optional comma separated list of Dimension Ids as per the Braiins Admin Utils [Dimensions List](http://admin.braiins.com/utils/DimensionsList.php) to be excluded from use with the Bro i.e. that are not allowable when posting to the Bro, or, for a Set Bro, when balances from other Bros are being summed to the Bro.  Individual dimension members of an excluded Dim can be brought back via the DiMes Allow list option.  Excl Dims for a BD Map can differ from the main Bro value. If it is the same it may be left blank. |
| Incl Dims | E | For non-Tx based Bros, Incl Dims is an optional comma separated list of Dimension Ids to be included as allowable for use with the Bro. In such cases the Incl Dims list becomes the Allowable Dims list.  Incl Dims cannot be used with Tx based Bro.  Incl Dims is mutually exclusive with Excl Dims. |
| DiMes | B | The DiMes or Dimension Members column allows for four different types of special dimension member handling for Bros or Maps. The first three types apply only to Bros which have Allowable Dims, either via an HyId or Incl Dims for a non-Tx based Bro:  **Mandatory** (m): DiMes which are Mandatory for the Bro, one per Dim. The DiMes must be in the Bro’s Allow Dims list, or the DiMe must be one of the Allow DiMes list below. This list is Mux with the Default and Exclude lists. When posting, any of the Mandatory DiMe(s) not included in a BroRef are added automatically.  **Default** (d): DiMes which are Defaults for the Bro, one per Dim. The DiMes must be in the Bro’s Allow Dims list, or the DiMe must be one of the Allow DiMes list below. This list is Mux with the Mandatory and Exclude lists. When posting, any of the Default DiMe(s) not included in the BroRef and for which there is no other DiMe from the same Dim as the Default DiMe, are added automatically.  **Exclude** (x): DiMes which are Excluded from use with the Bro. The DiMes must be in the Bro’s Allow Dims list. This list is Mux with the Mandatory, Default, and Allow lists.  The fourth type, **Allow** (a), has several functions depending on the Bro type/properties, as follows:  **Bro with Excl Dims**: Allow can be used with a Bro with Dim(s) excluded from use with the Bro via Excl Dims to re-enable just one or a few of the DiMes in that or those dimensions. The same result could be achieved by leaving the Dim(s) as allowable and excluding all the unwanted DiMes, but that could result in a long Exclude list.  **Bro with no Allowable Dims**: Allow, can also be used with non-Tx based Bros that do not have Allowable Dims, in a similar way to how Incl Dims can be used with such Bros. This provides the minimum and most specific definition of a Bro – just allowable dimension members.  **Main Bro i.e. a Bro with BD Maps**: Allow can be used to permit use of a BD DiMe with a Main Bro when no BD Map is provided for it i.e. no mapping to a different TxId is involved. (Without this, any BD DiMe for which a BD Map is not supplied is excluded from use with the Main Bro.)  **Source Format**  DiMes column values consist of a CS list of DIMeIds or DiMe ShortNames with each DiMeId/ShortName prefaced by the code letter shown above after the name of list type i.e. <m | d | x | a> and a colon. For backwards compatibility purposes ‘-‘ is also accepted for ‘x’. The <m | d | x | a> DiMe types can be intermingled in the DiMes column, and the entries can be in any order apart from in one rare case mentioned below in Summing Use of Mandatory/Default DiMes.  Any DiMes which cannot be used when posting i.e. ‘R’ or ‘Z’ types are ignored.  Examples of DiMes use are:  m:145 or m:NegativeGoodwill to make dimension member IFAclasses.NegativeGoodwill mandatory with negative goodwill related Bros.  m:148 or m:DevelopmentCosts to make dimension member IFAclasses.DevelopmentCosts mandatory for use with Asset development cost related Bros.  d:1288 or d:CoS to make dimension member Function.CoS the default for purchases related Bros  x:423 or x:CoSec to exclude the the Officers.CoSec dimension member from Director and PartnerLLP Bros.  x:1293,x:1295,x:1296,x:1297 to exclude >1 year details from Ageing for a debtor Bro. If the Bro was specifically for <1 year ageing then m1292 could be used instead.  a:1184,a:1284 to Allow English and Welsh for a Bro from which Languages had been excluded via Excl Dims 42.   * a:1292,a:1293 for a non-Tx based Bro to allow <1 or >1 Ageing The same end could be achieved via use of Incl Dims 44 and then excluding the unwanted DiMes i.e. x:1295,x:1296,x:1297. (There is no need to exclude 1294 as that is an ‘R’ or non-posting DiMe.) The a:1292,a:1293 method is shorter and clearer.   For a BD Map without its own DiMes column, DiMes are inherited from the Main Bro with any BD DiMes removed.  **Summing Use of Mandatory/Default DiMes**  When Bros are either SumEnd or Set summed, if the sum is for a mixture of Bros with and without DiMes in use, then a difference arises between the sum of the Primary DiMe values, and the Base Sum which includes both the with and without DiMe values. The difference is allocated to a DiMe so that after summing is finished the Base Sum for a Bro will equal the sum of its Primary DiMe values.  If the target Bro of the sum has a Mandatory DiMe list then the first DiMe in the Mandatory list is used for allocating the difference. This is the only time that the order of Mandatory DiMes in the DiMes column has meaning, though the most usual case is for the list to contain only one DiMe, in which case the order of DiMes in the DiMes column still does not matter.  If the target Bro of the sum has no Mandatory DiMe list but does have a Default DiMes list, then the first DiMe in the Default list is used for allocating the difference. This is the only time that the order of Default DiMes in the DiMes column has meaning, though the most usual case is for the list to contain only one DiMe, in which case the order of DiMes in the DiMes column still does not matter.  If the target Bro of the sum has neither a Mandatory DiMes list nor a Default DiMes list, then the Unallocated pseudo DiMe (9999) is used for allocating the difference. |
| Except | E | Except applies to Posting Money Bros. Allowable values are ‘Except’ or nothing.  Except means that this Bro is for an Exceptional Item. Except can only be set if one of the allowable dimensions for the Bro is 5 - Exceptional items adjustments. ‘Except’ is used when performing summing for dimension 5 before and after any adjustments for exceptional items:  150 - Dimension - Exceptional Item Adjustments [5]  [A] 132 After any adjustments for exceptional items  [A] 409 Before any adjustments for exceptional items  Nothing means that the Bro is for a normal or non-exceptional item, and that Dimension 5 summing does not apply. |
| Amort | E | Amort applies to Posting Money Bros. Allowable values are ‘Amort’ or nothing.  Amort means that this Bro is for Amortisation and Impairment Adjustments. Amort can only be set if one of the allowable dimensions for the Bro is 6 - Amortisation and Impairment Adjustments. ‘Amort’ is used when performing summing for dimension 6 before and after any amortisation and impairment adjustments:  160 - Dimension - Amortisation and Impairment Adjustments [6]  [A] 131 After any adjustments for amortisation and impairment  [A] 408 Before any adjustments for amortisation and impairment  Nothing means that the Bro is not an amortisation and impairment adjustments Bro, and that Dimension 6 summing does not apply. |
| Sum Up | E | Sum Up applies to Summing Bros and defines what auto summing does with the Bro balances in regard to automatic summing up the Bro tree. All Summing Bros not at level 0 must have a Sum Up property. Allowable values are:   |  |  | | --- | --- | | Nothing | Level 0 Bro for which no summing ‘up’ can be performed. | | + | Specifies that the balances of the Bro sum to the parent Set balances.  This is the default on creation of a new Summing Bro for which no Sum Up value is supplied. | | No | Specifies that the balances of the Bro are not to be summed to the parent Set i.e. that summing up the tree stops at this point.  No is typically used with a Set at the head of a duplicate or analysis branch of the Bro tree, whose sum should not be summed up the tree any further e.g. with PL.Revenue.Analysis so that the restated Revenue figures for analysis purposes are not summed to the PL.Revenue or PL sets which would result in the revenue sums being doubled. | | NA | Setting when auto summing does not apply because either the Bro’s Data Type is different from the parent Set’s Data Type, or the Bro is a Total Bro. | |
| Check | B | The Check column provides a means of specifying auto summing checks for Summing Bros, included BD Map and Slave ones, by defining whether the Bro’s value should be equal to, or equal and opposite to, the value of a ‘Check’ Target Bro. The two Bros involved must have the same data type.  Either or both Bros involved can be BD Maps.  The check is applied to single balances only, the Base or BD Map value as applicable.  The Check Column takes the following format:  {TargetBroRef in internal numeric form }<Equal To | Equal & Opp To>{, <Either | Both | Check | Target>} TargetBroRef  TargetBroRef is just a Bro Name unless a BD Map is involved, in which case the BD Dimension Member is appended to the Name to form the TargetBroRef.  Bros Export prefaces the Check field with the Check’s TargetBroRef in internal numeric form, but this is ignored during Import. When creating a new Check property it is not necessary to include the prefix of the TargetBroRef in internal numeric form.  The optional {, <Either | Both | Check | Target>} field defines when the check is to be Performed, with the options meaning:   * Either if Either Bro(Ref) has a value * Both if Both Bro(Ref)s have a value * Check if the Check (this) Bro has a value   Target if the Target BroRef has a value  The default is Either if no {, <Either | Both | Check | Target>} field is defined.  A Check of ‘Equal To, Either Master Bro Name’ is automatically added to a Summing Set Slave as explained in the Slave Bros section. |
| StartEnd | E | The StartEnd column applies to Taxonomy based Summing Bros whose Tx element is an Instant period one. Effectively the StartEnd column defines the type of an Instant Bro. There are five options:  **No**, or nothing which defaults to ‘No’ on import Bro based on an Instant Tx element that is not Start End as per the Taxonomy, nor in an accounting sense i.e. a Bro that has just One instant value, such as average number of staff, with properties:   * No StartEnd Sum List (also called SE Sum List) * No Start value * End value is to be posted * TB includes the value if Post Type is DE * Tag: End type * Can use Braiins Dimensions   **SumEnd** Bro based on an Instant Tx element that is StartEnd as per the Taxonomy, in the list of StartEnd Taxonomy element TxIds (82, 145, 146, 222, 511, 542, 1110, 1170, 1172, 1242, 2002, 2004, 2120, 2132, 2155, 2167, 2223, 2467, 2805, 2815, 3019, 3090, 3092, 3198, 3278, 3377, 3379, 3387, 3388, 3389, 3465, 3642, 3696, 3697, 3984, 4062, 4239, 4265, 4267, 4269, 4397, 4403, 4536, 4546, 4551, 4654, 4655, 4810, 5022, 5078, 5092, 5114, 5121, 5230, 5234) derived as described in Doc/BrosAndTx/StartEndPeriodNotes.txt, with properties:   * Has a SE Sum List as defined in ‘SumEnd and PostEnd SE Sum Lists’ below * Start = previous period End and is automatically RO * End = Start + the Sum of the SE Sum List of Bros * End is automatically RO * TB includes: Start balance AND the SE Sum List Bros * Tag: Start or End according to the Bro Reference * Cannot use Braiins Dimensions   **PostEnd** Bro based on an Instant Tx element that is Start End as per the Taxonomy, whose TxId is in the list given above for SumEnd, with properties:   * Has a SE Sum List as defined in ‘SumEnd and PostEnd SE Sum Lists’ below * Start = previous period End and is automatically RO * End is to be posted. Zero if no posting. * The SE Sum List provides a check, or means of calculating missing values. * TB includes: End balance but NOT the SE List Bros * Tag: Start or End according to the Bro Reference * Cannot use Braiins Dimensions   **Stock** Similar to PostEnd with the additional property of being a Stock Bro. The differences from PostEnd are:   * TB includes: P&L: Start and End with sign reversed  BS: End * Movement to be taken into account when checking that a posting journal balances.   **Acc** Bro based on an Instant Tx element that is not Start End as per the Taxonomy i.e. whose TxId is not in the list given above for SumEnd, but which is Start End in an Accounting sense like 541 Cash at bank, 575 Cash in hand, with properties:   * No SE Sum List * Start = previous period End and is automatically RO * End is to be posted. * TB includes: End * Tag: Start or End according to the Bro Reference. (It is yet to be confirmed whether a start tag is legal re iXBRL verification for such elements.)   **StartEnd Instant Bros**  For Instant Bros with StartEnd values, BroRefs for the Start values include a Start pseudo DiMeId of 0 as the first DiMeId of the reference.  A Bro reference for a StartEnd Bro can include an optional {:<start|end>} value. If neither is used the defaults is end.  **SumEnd, PostEnd and Stock SE Sum Lists**  For SumEnd, PostEnd and Stock Bros, the StartEnd column takes the following form:  <SumEnd|PostEnd|Stock> SeSumList ...  Where SeSumList is a list of TxIds either ‘,’ or ‘+’ separated, space separated from <SumEnd|PostEnd>, and used as described above for SumEnd, PostEnd and Stock.  There must be at least one SeSumList.  SumEnd Bros can have only one SeSumList.  PostEnd and Stock Bros can have multiple lists, space separated e.g. for TxId 542  The list(s) cannot include the TxId of the Bro unless the Bro is a Tuple Bro in which case the self reference means the sum of the other instances in the tuple. djh?? Re 222.  The I{nfo} StartEnd Bros col below shows the list(s) using Bro Ids rather than TxIds.  **Examples**   |  |  |  | | --- | --- | --- | | **Bro TxId** | **Description** | **StartEnd Column** | | 541 | Cash at bank | Acc | | 542 | Cash at bank and in hand | PostEnd 541,575 2646,2647,2648,2649 | | 5114 | Advances and credits, directors | SumEnd 5115+5116 | |
| Context | T | The Context of Standard Taxonomy based Bros and Main Bros is used by the report generator when creating the XBRL context to be used with an iXBRL tag for the Bro or Map. Context values typically don’t change much going down the tree, so can be inherited from parent Set values often all the way back up to the Level 0 Set. Export leaves out lower level values if they haven’t changed.  The allowable values are:   * Period Default that applies to most Bros whether the Tx Period is Instant or Duration * Officer Officer related Bro e.g. Company Secretary, Director etc * AllOfficers Officer in aggregate form. ‘Officer’ can become ‘AllOfficers’ * TPA Third Party Agent – Accountants or Auditors * RegOffice Specific to Registered Office Bros * UkCountry Specific to Country of Incorporation Bro |
| Zones | E | Zones is a comma separated list by Ref of up to 10 zones which apply to the Bro.  See Braiins Adim Zones to list or edit Zones.  If a Bro that is being output in a report has zones defined, the first zone in the list which has a sign defines the zone sign, which is one of the factors taken into account when processing the sign of a balance. See also the Sign column above.  During compilation of a format, if allowable zones have been set via a [zones ....] statement, and a Bro is used which has zones defined, a check is performed to see if one of the Bro’s zones appears in the [zones ....] statement list. If not, a compilation error occurs as use of the Bro at that location in the format is deemed to be invalid. |
| Order | E | Order is an optional integer number intended to define the sort order for elements within a set if an order different from the Bros Id order is required for report List type output purposes. As of 10 October 2011 no RG statement makes use of this property. |
| Descr | B | Descr is the Description or title text of up to 400 characters in length that is output by the report generator when the Bro is used in a title column of a table, or when the Bro is used with a ‘descr’ prefix in an expression.  The default value for ‘Descr’ for a taxonomy based Bro is the taxonomy element’s standard label which is given in the next column, I Tx Std Label.  Only if a description that is different from the taxonomy standard label, or if a description for a non-taxonomy based Bro is required, should the Descr column have a value.  If the Descr property of a non-taxonomy based Bro that has not been given a Descr property is referenced in a format, the name of the Bro is output as a reminder to “give me a Descr or hard code the required description in the format.” |
| Comment | B | Optional free form comment text of up to 500 characters in length. Comments are included in Bro Exports, are shown by Bros Lookup, and are available as an optional column in Bros List.  Also, whole rows may be commented out by starting them with ‘#’, ‘;’ or ‘//’. Import will skip such rows but preserve them (with column detail kept) for a subsequent Export. |
| Scratch | B | Optional free form comment text of up to 500 characters in length. The difference from Comment above is that Scratch is not shown in Bros Lookup or Bros Listing. It is included in an Export. |
| I Tx Std Label | I | The standard label of a taxonomy based Bro. See the previous column Descr for the relevance of this. |
| I Allowable Dims | I | Entries in this column are Info entries generated by the Export.  Allowable Dims is a comma separated list of the allowable dimensions, by Dimension Id, for a Bro as derived from its HyId, Excl Dims, and Incl Dims options. DiMes dimension member exclusions are not shown here.  Bros without an HyId do not have Allowable Dims. |
| I M# DiMe Info | I | Entries in this column are Info entries generated by the Export.  The column shows ‘M# DiMes’ if the Bro includes a Dim in its Allowable Dims list which includes M# Type DiMes.  It further shows ‘ + Prop’ if the Bro needs a Property DiMe as well, unless the following ' (No M# DiMe OK)' is also present, in which case the Property DiMe is optional.  In the special case of a Money Officers Bro that excludes DiMe 423 (Cosec) the column further shows ' (No M# DiMe OK)'.  See the Dimension Map Notes for more information about M# Type DiMes. |
| I Tag | I | iXBRL tag a taxonomy based Bro’s taxonomy element in the form Name space:Tx Name. |
| I Tx Type | I | Type of a taxonomy based Bro’s taxonomy element. The possible values and their normal mappings to a Bro Data Type are:   |  |  | | --- | --- | | **Taxonomy Type** | **Bro Data Type** | | None | None | | Money | Money | | String | String | | Boolean | Boolean | | Fixed | Boolean | | Date | Date | | Decimal | Decimal | | Integer | Integer | | NonZeroDecimal | Decimal | | Share | Share | | Percent | Percent | | PerShare | PerShare | | Uri, Domain, EntityAccounts, EntityForm, ReportPeriod, Any, QName, Arc, Doc, Extended, Locator, resource, Simple, Title | String | |
| I Tx Period | I | Period of a taxonomy based Bro’s taxonomy element.  The values may be Instant or Duration.  The Bro’s Context will normally be ‘Period’ for both Instant and Duration Tx elements, unless the Bro is one of those that has a special Context such as the Officer ones. |
| I StartEnd Bros | I | This column is a repeat of the StartEnd column with the StartEnd Sum List(s) shown using Bro Ids rather than TxIds. Bros Lookup also gives both forms of the StartEnd Sum Lists. Bros List gives just the TxId form. |
| I Tx Sign | I | Sign of a Money taxonomy based Bro’s taxonomy element.  The sign of a Money taxonomy based Bro will usually be the same as this, but it can be different. |
| I Tx Hys | I | Hypercube or hypercube list of a taxonomy based Bro’s taxonomy element.  See also Columns TxId and HyId. |
| I Tuple | I | For taxonomy based Bros with a TupId, more information about the tuple as TupleId TupleTxId TuMeId TUC and Tuple Short Name as per the Tuples List |
| I Slave Ids | I | The BroId(s) of the Slaves of a Master Bro. |

#### 17 September 2012 Update

DiMeId column replaced by DiMes column with greater power and versatility.

‘I Man Dim’ and ‘I Prop Dim’ columns replaced by ‘I M# DiMe Info’ column.

“Duplicate Hypercube and Taxonomy Element Bros” section added on page 5.

Check property extended to allow specification of when the test is to be applied via the new {, <Either | Both | Check | Target>} option.

* Some minor changes in the preamble sections.

#### 20 September 2012 Update

BD Maps made invalid if TxId mapping is not involved. BD DiMes can still be used with Bros without being mapped.

Incl Dims made inapplicable to BD Maps

Allow DiMe uses extended to permit BD DiMe use with a BD Main Bro that does not include a Map for the DiMe in question.

New ‘Prior Year Slave’ option described in the Slave Bros and Master Column sections.

Slave Bros section expanded and errors corrected. (Previously this section said that a Slave Bro had to be Tx based. That was not correct.)