IBM® DB2® OLAP Server

Version 1.0.1

Quick Technical Reference

SC26-9239-02

US Patent Number 5,359,724

Notice: The information contained in this document is subject to change without notice. Arbor Software Corporation shall not be liable for errors contained herein or consequential damages in connection with the furnishing, performance, or use of this material.

Arbor Software Corporation
1344 Crossman Avenue
Sunnyvale, CA 94089

Publication Number: SC26-9239-02

10 9 8 7 6 5 4 3 2 **R1**

Table of Contents

Essbase Functions

Calculation Commands

Report Writer Commands

ESSCMD Commands

ESSBASE.CFG File Settings

Essbase Functions

Essbase is designed with powerful calculation features. You can define complex formulas for use in calc scripts or in a database outline. A suite of functions is available to assist you in developing these functions.

This chapter includes the following:

- Tables of Calculation Operators
    - Mathematical Operators
    - Conditional and Logical Operators
    - Cross-Dimensional Operator
- Table of Function and Macro Parameters
- An alphabetic listing of functions

Table 1, Mathematical Operators

+	Addition operator.
-	Subtraction operator.
*****	Multiplication operator.
/	Division operator.
%	*Member1%Member2* evaluates *Member1* as a percentage of *Member2*.
()	Controls the order of calculations and nests equations and formulas.

Table 2, Conditional and Logical Operators

IF | ELSE | ELSEIF | ENDIF	Tests conditions and calculates a formula based on the success or failure of the test.
>	Data value is greater than.
>=	Data value is greater than or equal to.
<	Data value is less than.
<=	Data value is less than or equal to.
= =	Data value is equal to.
< >	Data value is not equal to.
AND	Logical AND linking operator for multiple value tests.
OR	Logical OR linking operator for multiple value tests.
NOT	Logical NOT operator.

Table 3, Cross-dimensional Operators

->	Points to data values of specific member combinations. The cross-dimensional operator is created with a dash (-) and right angle bracket (>). Do not leave spaces between the cross-dimensional operator and member names in quotes. For example, type `"New York"->Sales` and *not* `"New York" -> Sales`.

Table 4, Function and Macro Parameters

expression	Any mathematical or numeric expression valid within Essbase which, when calculated, returns a numeric value. This definition also includes the *numDigits*, *generation*, *level*, and similar parameters for the financial group of functions such as *rateMbrConst* and *lifeMbrConst*.
	`@ABS(Actual -Budget)`
	`@ROUND(Sales / 10.0 + 100)`

mbrName	Any valid single member name. This definition also includes similar parameters for the financial group of functions such as *balanceMbrName*, and *costMbr*, *cashflowMbr*.
	`@GEN(Actual)`
	`@CHILD(Sales)`

Table 4, Function and Macro Parameters

rangeList	Any valid single member name or a comma-delimited list of members, macro functions, and range functions. This definition also includes *mbrList*.
	`@ACCUM(Q189:Q491)`
	`@MAXRANGE(Mar,Apr,May,Jun)`

expList	A comma-delimited list of members, variable names, functions, and expressions, all of which return numeric values.
	`@MAX(Jan,Feb,100,Apr-May)`
	`@MIN(Oct:Dec)`

dimName	A single dimension name specification.
	`@CURLEV(Accounts)`
	`@CURGEN(Year)`

n	A positive or negative integer value.
	`@NEXT(2,Jan:Dec)`
	`@SHIFT(3)`

discountFlag	Any valid mathematical or numeric expression valid within Essbase which, when calculated, returns a 1 or 0. `@IRR(Cash,0,Q1:Q4,1991:1993)`
timePeriodList	A single range of members. `@PTD(Jan:May)`
tag	Any valid account tag defined in the current database including FIRST, LAST, AVERAGE, EXPENSE, and TWOPASS. `@ISACCOUNT(EXPENSE)`

@ABS (*expression*)

Returns the absolute value of *expression*. The absolute value of a number is that number less its sign. A negative number becomes positive, while a positive number remains positive.

expression

Member name or mathematical expression that generates a numeric value.

@ACCUM (*mbrName* [,*rangeList*])

Accumulates the values of *mbrName* within *rangeList*, up to the current member in the dimension of which *rangeList* is a part.

mbrName

Single member specification whose value is to be accumulated.

rangeList

Optional list of members, macro functions, and range functions across which the accumulation is to occur.

@ALLANCESTORS (*"mbrName"*)

Expands the selection to include all ancestors of the specified member, including ancestors of any occurrences of the specified member as a shared member. This macro excludes the specified member.

You can use **@ALLANCESTORS** as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@ANCESTORS (*mbrName* [, *genLevNum* | *genLevName*])

Expands the selection to include either all ancestors of the specified member or those up to a specified generation or level. You can use this macro function as a parameter of another function, where the acquired parameter is a list of members.

mbrName

A single member specification.

genLevNum

Optional. An integer value that defines the generation or level number up to which to select the members. A positive integer defines a generation number. A value of 0 or a negative integer defines a level number.

genLevName

Optional. Level name or generation name up to which to select the members.

@ANCESTVAL (*dimension*, *genlevel*, *mbrName*)

Returns the ancestor values of a specified member combination.

dimension

Defines the focus dimension of ancestor values.

genlevel

Integer value that defines the generation or level number from which the ancestor values are to be returned. A positive integer defines a generation reference. A negative number or value of 0 defines a level reference.

mbrName

Defines a specific member or member combination.

@**AVG** (SKIPNONE | SKIPMISSING | SKIPZERO | SKIPBOTH, *expList*)

Returns the average of all the values of *expList*.

SKIPNONE

> Includes all cells specified in the average operation regardless of their content.

SKIPMISSING

> Excludes all values that are `#MISSING` in the average operation.

SKIPZERO

> Excludes values of zero from the average calculation.

SKIPBOTH

> Excludes all values of zero or `#MISSING` from the average calculation.

expList

> Comma-delimited list of members, macro functions, or range functions.

@AVGRANGE (SKIPNONE | SKIPMISSING | SKIPZERO | SKIPBOTH, *mbrName* [,*rangeList*])

Returns the average value of the specified member across a specified range.

SKIPNONE

Includes all cells specified in the average operation regardless of their content.

SKIPMISSING

Excludes all values that are `#MISSING` in the average operation.

SKIPZERO

Excludes values of zero from the average calculation.

SKIPBOTH

Excludes all values of zero or `#MISSING` from the average calculation.

mbrName

Single member specification.

rangeList

Optional comma-delimited list of members, macro functions, or range functions.

@CHILDREN (*mbrName*)

Expands to include all children of the specified member. This macro function can be used as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@COMPOUND (*balanceMbr*, *rateMbrConst* [,*rangeList*])

Compiles the proceeds of a compounded interest calculation based on the balances of a specified member at a specified rate across a specified range.

balanceMbr

Single member specification representing the beginning balance across a range of periods. The input can be either one deposit or a series of deposits. If *balanceMbr* is a constant, then Essbase assumes *balanceMbr* to be a single deposit in the first member of *rangeList*. (This is equivalent to entering the constant value in the first member in the *rangeList* followed by zeros. The function keeps track of each deposit separately, but returns a composite value.) If *balanceMbr* is a member or a range, then it is assumed to be a series of deposits.

rateMbrConst

Single member specification, variable name, or numeric expression in decimal form. Represents the interest rate per time period specified in the *rangeList*. If your interest is compounded monthly, this value would be the annual interest rate divided by 12.

rangeList

Optional comma-delimited list of members, macro functions, and range functions. Represents the range over which the interest is compounded. The last value in the range is the total compounded interest for that range. The range must be from a Time dimension.

@COMPOUNDGROWTH (*principalMbr*, *rateMbrConst* [,*rangeList*])

Calculates a series of values that represent a compound growth of the first non-zero value in the specified member across a range of members. The growth factor is calculated by multiplying the growth rate in the current time period by the previous period's result, yielding a compounded value. You can change the growth rate from period to period if you place a non-zero value in the current period's *rateMbrConst* cell.

principalMbr

Member specification representing the initial value of the range to be compounded. The input line must be a single deposit.

rateMbrConst

Single member specification, variable name, or expression which provides a constant value. This value can change across *rangeList*, making the new value be the new compound rate. If the value in the current period is zero, the compound rate is equal to zero, and the principal does not change.

rangeList

Optional comma-delimited list of members, macro functions, and range functions. The range must be from a Time dimension.

@CURGEN (*dimName*)

Returns the generation number of the current member combination for the specified dimension. The generation number is the number of members that separates the current member, or the top-most member in the dimension.

dimName

> Single dimension-name specification. This must be a dimension member name (the name of the dimension). It cannot be another member name from within the dimension.

@CURLEV (*dimName*)

Returns the level number of the current member combination for the specified dimension. The level number is the number of members that separates the current member from its bottom-most descendant.

dimName

> Single dimension name specification. This must be a dimension member name (the name of a dimension). It cannot be another member from within the dimension.

@CURRMBRRANGE (*dimName*, {GEN|LEV}, *genLevNum*, [*Start*], [*End*])

Returns a member range list based on the relative position of the current member being calculated.

dimName

> Name of the dimension for which you want to return the range list.

GEN | LEV

> Defines whether the range list to be returned is based on a generation or level within the dimension.

genLevNum

Integer value defining the generation or level number of the range list to be returned.

Start

Defines the first member in the range to be returned.

- A null value returns the first member of the specified *genLevNum*.
- An integer value returns the member name relative to the current member being calculated.
- A negative value specifies a member prior to the current member being calculated in the dimension.
- A value of 0 returns the name of the member currently being calculated.
- A positive value specifies a member after the current member being calculated in the dimension.

End

Defines the last member in the range to be returned.

- A null value returns the last member of the specified *genLevNum*.
- An integer value returns the member name relative to the current member being calculated.
- A negative value specifies a member prior to the current member being calculated in the dimension.
- A value of 0 returns the name of the member currently being calculated.
- A positive value specifies a member after the current member being calculated in the dimension.

@DECLINE (*costMbr*, *salvageMbrConst*, *lifeMbrConst*, *factorMbrConst* [,*rangeList*])

Calculates the depreciation of an asset for a specific period using the declining balance method. The factor by which the declining balance depreciates the assets is specified using *factorMbrConst*. For example, to calculate a double declining balance, set *factorMbrConst* to 2.

costMbr

> Single member specification representing the starting values of the assets. More than one asset can be input and depreciated across the range. The function calculates each asset separately.

salvageMbrConst

> Single member specification, variable name, or numeric expression which provides a constant value representing the value of the asset at the end of the depreciation.

lifeMbrConst

> Single member specification, variable name, or numeric expression which provides a constant value. The number represents the number of periods over which the asset is being depreciated.

factorMbrConst

> Single member specification, variable name, or numeric expression which provides a constant value. The value represents the factor by which the asset is depreciated.

rangeList

Optional comma-delimited list of member names, macro functions, and range functions. The range represents the periods over which the function is calculated. More than one asset can be depreciated. The range must be from a Time dimension.

@DESCENDANTS (*mbrName* [, *genLevNum* | *genLevName*])

Expands to include either all descendants of the specified member, or those down to a specified generation or level. You can use this macro function as a parameter of another function, where that parameter is a list of members.

mbrName

A single member specification.

genLevNum

Optional. An integer value that defines the generation or level number down to which to select the members. A positive integer defines a generation number. A value of 0 or a negative integer defines a level number.

genLevName

Optional. Level name or generation name down to which to select the members.

@DISCOUNT (*cashMbr*, *rateMbrConst* [,*rangeList*])

Calculates a value discounted by the specified rate, from the first period of the range to the period in which the amount to discount is found. The answer is returned in the same period. More than one value can be simultaneously discounted in this manner.

cashMbr

Member specification representing the value you want to discount from the last period in *rangeList* to the current period.

rateMbrConst

Member specification, variable name, or numeric expression which provides a constant value. The value represents the rate per period which *cashMbr* is discounted. It is a decimal value, not a percent.

rangeList

Optional comma-delimited list of members and macro and range functions.

ELSE *statement* ; [...*statement* ;] ENDIF

Command used to designate a conditional action to be performed in an **IF** statement. The **ELSE** command can only be used in conjunction with an **IF** command. All actions placed after the **ELSE** in an **IF** statement are performed only if the test in the **IF** statement generates a value of FALSE.

statement

Those operations that are to be performed in the event that the **IF** test including the **ELSE** command produces a FALSE, or 0, result.

ELSEIF (*condition*) *statement* ; [...*statement* ;] ELSEIF | ELSE | ENDIF

Designates a conditional test and conditions that are performed if the preceding **IF** test generates a value of FALSE. For this reason, multiple **ELSEIF** commands are allowed following a single **IF**. The **ELSEIF** command can only be used in conjunction with an **IF** command.

condition

Formula or function which returns a Boolean value of TRUE (a non-zero value) or FALSE (a zero value)

statement

Those operations that are to be performed in the event that the **IF** test including the **ELSE** command produces a TRUE, or 1, result.

ENDIF

The **ENDIF** function is used to mark the end of an **IF** command sequence. If the **ENDIF** function is used in a calc script, or is part of an embedded **IF** test (for example, more than one **IF** sequence contained within the same formula), the semicolon is required. If it is used as the end of a formula, the semicolon is not required.

@FACTORIAL (*expression*)

Returns the factorial of *expression*.

expression

Single member specification or numeric expression.

@GEN (*mbrName*)

Returns the generation number of the specific member.

mbrName

Single member specification.

@GENMBRS (*dimName*, *genName*)

Expands the selection to include all the members with the specified generation number or generation name in the specified dimension.

dimName

> Dimension name specification from the associated outline.

genName

> Generation name or generation number that you want to expand.

@GROWTH (*principalMbr*, *rateMbrConst* [*,rangeList*])

Calculates a series of values that represent a linear growth of the first non-zero value encountered in *principalMbr* across the range specified by *rangeList*. Growth is calculated by multiplying the growth rate in *rateMbrConst* by the original *principalMbr*. This value is then added to the previous time period's result, yielding the new value.

principalMbr

> Single member specification representing the initial value of the value to grow. The first non-zero value encountered is the initial value. Other *principalMbr* values after the first are ignored.

rateMbrConst

> Single member specification, variable name, or numeric expression providing a constant value representing the decimal growth rate to be applied (for example, 10% = .1).

rangeList

> Optional comma-delimited list of member specifications, macro functions, and range functions across which the function operates.

@IALLANCESTORS (*mbrName*)

Expands the selection to include the specified member and all its ancestors, including ancestors of any occurrences of the specified member as a shared member. You can use this macro function as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@IANCESTORS (*mbrName* [, *genLevNum* | *genLevName*])

Expands the selection to include the specified member and either all ancestors or only those up to a specified generation or level. You can use this macro function as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

genLevNum

Optional. An integer value that defines the generation or level number up to which to select the members. A positive integer defines a generation number. A value of 0 or a negative integer defines a level number.

genLevName

Optional. Level name or generation name up to which to select the members.

@ICHILDREN (*mbrName*)

Expands to include the specified member and all members in the level immediately below it. This macro function can be used as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@IDESCENDANTS (*mbrName* [, *genLevNum* | *genLevName*])

Expands to include the specified member, and either all its descendants or only those down to a specified generation or level. You can use this macro as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

genLevNum

Optional. An integer value that defines the generation or level number down to which to select the members. A positive integer defines a generation number. A value of 0 or a negative integer defines a level number.

genLevName

Optional. Level name or generation name down to which to select the members.

IF (*condition*) *statement* ; [*...statement* ;] [**ELSEIF***...statement* **|** ELSE*...statement*] **ENDIF**

The **IF** command is used to perform conditional tests within a formula. Using the **IF** test, users are allowed to define a Boolean test, as well as formulas to be calculated if the test creates either a TRUE or FALSE value. The related **ELSE**, **ELSEIF**, and **ENDIF** commands immediately follow this function.

condition

Formula or function which returns a Boolean value of TRUE (a non-zero value) or FALSE (a zero value).

statement

Those operations that are to be performed in the event that the test returns a TRUE value.

@ILSIBLINGS (*mbrName*)

Expands to include the specified member and all of the left siblings of the member. Left siblings are children which share the same parent as the member, and which precede the specified member in the database outline. This macro function can be used as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@INT (*expression*)

Returns the next lowest integer value of *expression*.

expression

Member name or mathematical expression which generates a numeric value.

@INTEREST (*balanceMbr*, *creditrateMbrConst*, *borrowrateMbrConst* [,*rangeList*])

Calculates the simple interest in *balanceMbr* at the rate specified by *creditrateMbrConst* if the value specified by *balanceMbr* is positive, or at the rate specified by *borrowrateMbrConst* if *balanceMbr* is negative. The interest is calculated for each time period specified by *rangeList*.

balanceMbr

Single member specification representing the balance at the time the interest is calculated.

creditrateMbrConst

Single member specification, variable name, or numeric expression providing a constant value. The value must be a decimal number that corresponds to a percentage. The value represents the per-period interest rate.

borrowrateMbrConst

Single member specification, variable name, or numeric expression providing a constant value. The value must be a decimal number corresponding to a percentage value. The value represents the per-period interest rate.

rangeList

Optional comma-delimited list of members, macro functions, and range functions representing the range of members over which the calculation is applied. The range must be from a `Time` dimension.

@IRR (*cashflowMbr*, *discountFlag* [,*rangeList*])

Calculates the internal rate of return on a cash flow that must contain at least one investment (negative) and one income (positive) value.

cashflowMbr

Single member specification.

discountFlag

Member specification, variable name, or numeric expression providing a constant value of either 1 or 0. Indicates whether the function should discount from the first period. 1 means do not discount from the first period. 0 means discount from the first period.

rangeList

Optional comma-delimited list of members, macro functions, and range functions. The range must be from a Time dimension.

@IRSIBLINGS (*mbrName*)

Expands to include the member and all of its right siblings. Right siblings are children that share the same parent as the member and that follow it in the database outline. This macro function can be used as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@ISACCTYPE (*tag*)

Returns TRUE if the current member has the associated accounts *tag*.

tag

Valid account tag defined in the current database. Any of the following tags may be used: FIRST, LAST, AVERAGE, EXPENSE, and TWOPASS.

@ISANCEST (*mbrName*)

Returns TRUE if a current member is the specified member or an ancestor of the specified member.

mbrName

Single member specification.

@ISCHILD (*mbrName*)

Returns TRUE if a current member is a child of the specified member.

mbrName

Single member specification.

@ISDESC (*mbrName*)

Returns TRUE if a current member is a descendant of the specified member.

mbrName

Single member specification.

@ISGEN (*dimName*, *generation*)

Returns TRUE if the current member of the specified dimension is of a certain generation.

dimName

Dimension name specification. This must be a dimension member name.

generation

A generation name string or a non-negative number, representing the generation number.

@ISIANCEST (*mbrName*)

Returns TRUE if a current member is the specified member or an ancestor of the specified member.

mbrName

Single member specification.

@ISIBLINGS (*mbrName*)

Expands to include all siblings of the member and the specified member. This macro function can be used as a parameter of another function, where that parameter is a list of members.

This function excludes the specified member.

mbrName

Single member specification.

@ISICHILD (*mbrName*)

Returns TRUE if a current member is the specified member or a child of the specified member.

mbrName

Single member specification.

@ISIDESC (*mbrName*)

Returns TRUE if a current member is the specified member or a descendant of the specified member.

mbrName

Single member specification.

@ISIPARENT (*mbrName*)

Returns TRUE if a current member is the specified member or the parent of the specified member.

mbrName

Single member specification.

@ISISIBLING (*mbrName*)

Returns TRUE if a current member is the specified member or a sibling of the specified member.

mbrName

Single member specification.

@ISLEV (*dimName*, *level*)

Returns TRUE if the current member of the specified dimension is of the specified level.

dimName

Dimension member-name specification.

level

A level name string or a non-negative number, representing the level number.

@ISMBR (*mbrName* | *rangeList* | *mbrList*)

Returns TRUE if a current member matches any one of the specified members.

mbrName

Single member specification.

rangeList

Comma-delimited list of members, macro functions, and range functions.

mbrList

Comma-delimited list of members.

@ISPARENT (*mbrName*)

Returns TRUE if a current member is the specified member or the parent of the specified member.

mbrName

Single member specification.

@ISSAMEGEN (*mbrName*)

Returns TRUE if the current member in the same dimension is of the same generation as the specified member.

mbrName

Single member specification.

@ISSAMELEV (*mbrName*)

Returns TRUE if the current member in the same dimension is of the same level as the specified member.

mbrName

Single member specification.

@ISSIBLING (*mbrName*)

Returns TRUE if a current member is a sibling of the specified member.

mbrName

Single member specification.

@ISUDA (*dimName*, *UDAStr*)

Returns TRUE if the specified user-defined attribute (UDA) exists for the current member of the specified dimension at the time of the calc.

dimName

Name of a valid dimension in the associated outline which contains the member you are checking.

UDAStr

User-Defined Attribute name string.

@LEV (*mbrName*)

Returns the level number of the specified member.

mbrName

Single member specification.

@LEVMBRS (*dimName*, *levName*)

Expands the selection to include all the members with the specified level number or level name in the specified dimension.

dimName

Dimension name specification of a valid dimension in the associated outline.

levName

Level name or level number that you want to expand.

@LSIBLINGS (*mbrName*)

Expands to include the specified member and all of the left siblings of the member. Left siblings are children which share the same parent as the member, and which precede the specified member in the database outline. This macro function can be used as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@MATCH (*member|gen|level*, "*pattern*")

Allows you to do wildcard (* or ?) member selections.

member

The default or user-defined name of the member on which to base the search. Essbase searches the member names and alias names of the specified member and its descendants.

gen

The default or user-defined name of the generation to search. Essbase searches all member names and member alias names in the generation.

level

The default or user-defined name of the level to search. Essbase searches all member names and member alias names in the level.

pattern

The character pattern to search for, including a wildcard character (* or ?).

- ? substitutes one occurrence of any character. You can use ? anywhere in the pattern.
- * substitutes any number of characters. You can use * only at the end of the pattern.
- To include spaces in the character pattern, enclose the pattern in double quotation marks ("").

@MAX (*expList*)

Returns the maximum value among the results of the expressions in the member list.

expList

Comma-delimited list of members, variable names, functions, and numeric expressions, all of which return numeric values.

@MAXRANGE (*mbrName* [*,rangeList*])

Returns the maximum value of *Mbomou* across a range of members.

mbrName

Single member specification.

rangeList

Optional comma-delimited list of members, macro functions, and range functions.

@MDANCESTVAL (*NumDim*, *DimName1*, *GenLevNum1*...*DimNameX*, *GenLevNumX* [,*Xmbr*])

Returns ancestor-level data from multiple dimensions based on the current member being calculated.

NumDim

Integer value that defines the number of dimensions from which ancestor values are being returned.

DimName1, ...DimNameX

Defines the dimension names from which the ancestor values are to be returned. You must specify a *GenLevNum* for every *DimName*.

GenLevNum, ...GenLevNumX

Integer value that defines the generation or level number from which the ancestor values are to be returned. A positive integer defines a generation reference. A negative number or value of 0 defines a level reference. You must specify a *DimName* for every *GenLevNum*.

Xmbr

Optional. Specific member for which the ancestor values are returned.

@MDPARENTVAL (*NumDim*, *DimName1*, ...*DimNameX* [,*Xmbr*])

Returns parent-level data from multiple dimensions based on the current member being calculated.

NumDim

Integer value that defines the number of dimensions from which parent values are to be returned.

DimName1, ...DimNameX

Defines the dimension names from which the parent values are to be returned.

Xmbr

Optional. Specific member for which the parent values are to be returned.

@MDSHIFT (*Xmbr*, *ShiftCnt1*, *DimName1*, [*Range1*|(*Range1*)], ...*ShiftCntX*, *DimNameX*, [*RangeX*|*RangeX*)])

Shifts a series of data values across multiple dimension ranges.

Xmbr

Member for which the values are to be shifted.

ShiftCnt1 ...ShiftCntX

Integer that defines the number of member positions to shift.

DimName1, ...DimNameX

Defines the dimension names in which the shift is to occur.

Range1 | (*Range1*) ...*RangeX* | *RangeX*)

Optional. Defines the member ranges over which the shift occurs. If the range list is comma-delimited, then the list must be enclosed in parenthesis. If this parameter is omitted, then the range is defined as all level 0 members.

@MIN (*expList*)

Returns the minimum value among the results of the expressions in *expList*.

expList

> Comma-delimited list of members, variable names, functions, and numeric expressions, all of which return numeric values.

@MINRANGE (*mbrName* [*,rangeList*])

Returns the minimum value of *mbrName* across the *rangeList*.

mbrName

> Single member specification.

rangeList

> Optional comma-delimited list of members, macro functions, and range functions.

@MOD (*member1*, *member2*)

Calculates the modulus of a division operation.

member1 and *member2*

> Members from the same dimension, whose modulus is to be calculated.

@NEXT (*mbrName* [, *n*, *rangeList*])

Returns the *n*th cell value in the sequence *rangeList* from *mbrName*, retaining all other members identical to the current member. If *n* is not given, then the default is set to 1, which provides the next member in the range. If *rangeList* is not given, then the tagged Time dimension from the database outline is used as a default. **@NEXT** cannot operate outside the given range.

mbrName

Single member specification.

n

Optional signed integer. Using a negative value for *n* has the same effect as using the matching positive value in an **@PRIOR** function.

rangeList

Optional comma-delimited list of members, macro functions, and range functions.

@NPV (*cashflowMbr*, *rateMbrConst*, *discountFlag* [,*rangeList*])

Calculates the Net Present Value of an investment based on the series of payments (negative values) and income (positive values).

cashflowMbr

Member specification providing a series of numeric values.

rateMbrConst

Single member specification, variable name, or numeric expression, providing a constant value.

discountFlag

Single member specification, variable name, or numeric expression set to 0 or 1 to indicate whether the function should discount from the first period. 1 means do not discount from the first period.

rangeList

Optional comma-delimited list of members, macro functions, and range functions.

@PARENTVAL (*dimension*, *Xmbr*)

Returns the parent values of a specified member combination.

dimension

Defines the focus dimension of parent values.

Xmbr

Defines a specific member or member combination.

@POWER (*expression*, *power*)

Returns the value of a specified member or expression raised to *power*.

expression

Single member specification, variable name, function, or other numeric expression.

power

Single member specification, variable name, function, or other numeric expression.

@PRIOR (*mbrName* [, *n*, *rangeList*])

Returns the *n*th previous cell member from *mbrName* in *rangeList*. All other dimensions assume the same members as the current member. If *n* is not given, then a default value of 1 is assumed, which returns the next prior member from the lowest level of the Time dimension in the database outline. If *rangeList* is not given, then the Time dimension is used as a default value. **@PRIOR** only works within the designated range.

mbrName

Single member specification.

n

Optional signed integer. Using a negative value for *n* has the same effect as using the matching positive value in an **@NEXT** function.

rangeList

Optional comma-delimited list of members, macro functions, and range functions.

@PTD (*timePeriodList*)

Calculates the period-to-date values of members in the Time dimension. By default, data is summed unless Accounts are tagged as “First” or “Last.”

timePeriodList

Range of members from the Time dimension.

@RELATIVE (*mbrName*, *genLevNum* | *genLevName*)

Selects all the members at a specified generation or level that are above or below a specified member.

mbrName

Single member specification.

genLevNum

An integer value that defines the generation or level number to select. A positive integer defines a generation number. A value of 0 or a negative integer defines a level number.

genLevName

Level name or generation name to select.

@REMAINDER (*expression*)

Returns the remainder value of *expression*.

expression

Single member specification, variable name, or other numeric expression.

@ROUND (*expression*, *numDigits*)

This function rounds *expression* to *numDigits* digits. When *numDigits* is 0, the number is rounded to the nearest integer. If *numDigits* is greater than 0, then *expression* is rounded to the specified number of decimal places. A negative value for *numDigits* rounds *expression* to a power of 10. The default value for *numDigits* is 0.

expression

Single member specification, variable name, or other numeric expression.

numDigits

Single member specification, variable name, or other numeric expression that provides an integer value.

@RSIBLINGS (*mbrName*)

Expands to include all of the right siblings of the member. Right siblings are children which share the same parent as the member, and which follow the member in the database outline. This macro function can be used as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@SANCESTVAL (*RootMbr*, *genLevNum* [,*Xmbr*])

Returns ancestor-level data based on the shared ancestor value of the current member being calculated.

RootMbr

Defines a member used to search for the nearest occurrence of an ancestor of a shared member.

genLevNum

Integer value that defines the generation or level number from which the ancestor values are to be returned. A positive integer defines a generation reference. A negative number or value of 0 defines a level reference.

Xmbr

Optional. Specific member for which the ancestor values are returned.

@SHIFT (*mbrName* [,*n*, *rangeList*])

Returns the *n*th previous or next cell member from *mbrName* in *rangeList*. The direction of shift (prior to *mbrName* or after *mbrName*) is based on *n*, with positive *n* values producing an effect equivalent to **@NEXT** and negative *n* values producing an equivalent to **@PRIOR**.

mbrName

Single member specification.

n

Optional signed integer. If a negative value is used, the effect is identical to using the matching positive value for *n* in an **@PRIOR** function. *n* must be a numeric value, not a reference such as a member name.

rangeList

Optional comma-delimited list of members, macro functions, and range functions.

@SIBLINGS (*mbrName*)

Expands to include all siblings of the member, excluding the specified member. This macro function can be used as a parameter of another function, where that parameter is a list of members.

mbrName

Single member specification.

@SLN (*costMbr*, *salvageMbrConst*, *lifeMbrConst* [,*rangeList*])

Calculates the amount per period that an asset in the current period may be depreciated, calculated across a range of periods. The depreciation method used is straight-line depreciation.

More than one asset may be depreciated over the range. The value is depreciated from its entry period to the last period in the range. The resulting value represents the sum of all the per-period depreciation values of each asset being depreciated.

costMbr

Single member specification representing an input asset for the current period.

salvageMbrConst

Single member specification, variable name, or numeric expression, providing a constant numeric value. This value represents the value of the asset in the current period at the end of the useful life of the asset.

lifeMbrConst

Single member specification, variable name, or numeric expression representing the useful life of the asset.

rangeList

Optional comma-delimited list of members, macro functions, and range functions that represent the range over which the function accepts input and returns depreciation values. The range must be from the Time dimension.

@SPARENTVAL (*RootMbr* [,*Xmbr*])

Returns parent-level data based on the shared parent value of the current member being calculated.

RootMbr

Defines a member that is used to search for the nearest occurrence of a parent of a shared member.

Xmbr

Optional. Specific member for which the parent values are returned.

@STDDEV (*expList*)

Returns the standard deviation of all the values of the member list.

expList

Comma-delimited list of member specifications, variable names, or other numeric expressions, all of which provide numeric values.

@STDDEVRANGE (*mbrName* [*,rangeList*])

Returns the standard deviation of all the values of the specified members across the specified range.

mbrName

Single member specification

rangeList

Optional comma-delimited list of members, macro functions, and range functions.

@SUM (*expList*)

Returns the summation of all the values in the member list.

expList

Comma-delimited list of member specifications, variable names, or other numeric expressions, all of which provide numeric values.

@SUMRANGE (*mbrName* [*,rangeList*])

Returns the summation of all the values of the specified member across the specified range.

mbrName

Single member specification.

rangeList

Optional comma-delimited list of members, macro functions, and range functions.

@SYD (*costMbr*, *salvageMbrConst*, *lifeMbrConst* [,*rangeList*])

Calculates across the range of periods, the amount per period that an asset in the current period may be depreciated. The depreciation method used is sum of the year's digits. More than one asset may be depreciated over the range. The value is depreciated from its entry period to the last period in the range. The resulting value represents the sum of all per-period depreciation values of each asset.

costMbr

Single member specification representing an input asset for the current period.

salvageMbrConst

Single member specification, variable name, or numeric expression, providing a constant numeric value. This value is the value of the asset in the current period after the useful life of the asset.

lifeMbrConst

Single member specification, variable name, or numeric expression representing the useful life of the asset.

rangeList

Optional comma-delimited list of members, macro functions, and range functions which represents the range over which2 the function accepts input and returns depreciation values. The range must be within a Time dimension.

@TRUNCATE (*expression*)

Removes the fractional part of *expression*, returning the integer. This is equivalent to using **@ROUND** with the number of digits set to 0.

expression

> Single member specification, function, variable name, or other numeric expression, which returns a numeric value.

@UDA (*dimName*, *UDAStr*)

Allows you to select members based on a common attribute, which you have defined as a User-Defined Attribute (UDA) on the Essbase server.

dimName

> Name of the dimension associated with *UDAStr*.

UDAStr

> Name of the user-defined attribute as it appears in the database outline.

@VAR (*member1*, *member2*)

Calculates the variance (difference) between two members. The variance calculation recognizes the difference between accounts that are tagged in the Database Outline as “Expense” or “No Expense” and calculates the variance accordingly.

member1 and *member2*

> Members from the same dimension whose variance results are to be calculated. The variance is derived by subtracting *member2* values from *member1*, unless an account is tagged as “Expense”, in which case *member1* values are subtracted from *member2*.

@VARPER (*member1*, *member2*)

Calculates the percent variance (difference) between two members. The variance calculation recognizes the difference between accounts that are tagged in the database outline as “Expense” or “No Expense” and calculates the variance accordingly.

member1 and *member2*

> Members from the same dimension whose variance results are to be calculated. The percent variance is derived by taking the percent variance of *member2* values from *member1*, unless an account is tagged as “Expense,” in which case *member1* values are taken as a percent variance of *member2*.

Calculation Commands

Calculation commands are defined within a calculation script to instruct Essbase in the exact calculation rules to be used. The Calculation Script Editor is used to create a script. Within the Editor, a dialog box is available, allowing you to paste functions while you develop formulas.

Note: Every formula and calc script command ends with a semicolon (;). For example, `CALC DIM(Product, Measures);`

This chapter includes an alphabetic listing of commands.

&(*variableName*)

Prefaces a substitution variable in your calc script.

variableName

The name of the substitution variable set on the database.

AGG (*dimList*) ;

Consolidates database values. An **AGG** command ignores all member formulas; it simply consolidates parent/child relationships.

dimList

Name of a single dimension or a comma-delimited list of dimensions.

ARRAY *arrayVariableName* [*dimName*] = {*constList* } ;

Declares one-dimensional array variables.

arrayVariableName

> Comma-delimited list of one or more array variable names.

dimName

> Dimension whose size determines the size of the array variable. You must surround `dimName` with brackets [] when using the **ARRAY** command.

constList

> Optional list of data values used to initialize the array variables. If no initialization is performed, then the array variables are set to `#MISSING`. The order of the values corresponds to the order of the members in the dimension used to define the array.

CALC ALL [EXCEPT DIM (*dimList*) | MBR (*mbrList*)] ;

Calculates and aggregates the entire database based on the database outline.

EXCEPT

> Defines an exception list of dimensions or members that are not to be included in the calculation.

DIM

> Single-dimension specification.

dimList

> Optional comma-delimited list of dimensions.

MBR

> Single-member specification.

mbrList

Optional comma-delimited list of members, macro @functions, or range functions.

CALC AVERAGE ;

Calculates all members tagged in the database outline as type “Average” or “Average non-missing.” All other member calculations are ignored during this process.

Note: This command calculates the members based on the Accounts dimension only; it does not do a Time Series calculation on the Time dimension.

CALC DIM (*dimList*) ;

Calculates the formulas and aggregations associated with each member of all the specified dimensions.

dimList

Dimension or comma-delimited list of dimensions to be calculated.

CALC FIRST ;

Calculates all members tagged in the Database Outline as type “First”.

Note: This command calculates the members based on the Accounts dimension only; it does not do a Time Series calculation on the Time dimension.

CALC LAST ;

Used to calculate all members tagged in the Database Outline as type “LAST”.

Note: This command calculates the members based on the Accounts dimension only; it does not do a Time Series calculation on the Time dimension.

CALC TWOPASS ;

Calculates all members tagged in the database outline as type “TwoPass”. These members must be on a dimension tagged as “Accounts”.

CCONV *currExchMbr* | TOLOCALRATE *curType* ;

Calculates currency conversions.

This command is only available if your company has purchased the optional Currency Conversion feature in Essbase.

currExchMbr

> Currency name containing the required exchange rate. This is a member from the Currency database.

TOLOCALRATE

> Converts a converted currency back to the original, local rate.

curType

> Currency type. This is a member from the CurType dimension in the Currency database.

CLEARBLOCK ALL | UPPER | NONINPUT ;

Clears blocks of data and sets them to `#MISSING`.

ALL

Clears all data blocks.

UPPER

Clears all consolidated level blocks.

NONINPUT

Clears blocks containing values derived from calculations.

CLEARDATA *mbrName* ;

Clears data values from the database and sets them to `#MISSING`.

mbrName

Member name or cross-dimensional member combination.

DATACOPY *mbrName1* TO *mbrName2* ;

Copies a range of data cells to another range within the database.

mbrName1 and *mbrName2*

Individual members or a cross-dimensional member combination.

FIX (*fixMbrs*) *COMMANDS* ; **ENDFIX**

This block command restricts database calculations to a subset of the database. All commands nested between the **FIX** and **ENDFIX** statements are restricted to the specified database subset.

fixMbrs

A member name or list of members from any number of database dimensions. *fixMbrs* can also contain:

- AND / OR operators. Use the AND operator when all conditions must be met. Use the OR operator when one condition of several must be met.
- Macro functions, which are used to build member lists based on other members.

COMMANDS

The commands you want executed for the duration of the **FIX**.

LOOP (*integer*, [*break*])
COMMANDS ;
ENDLOOP

This block command lets you specify the number of times Essbase iterates calculations. All commands between the **LOOP** and **ENDLOOP** statements are performed the number of times that you specify.

integer

The integer constant that indicates the number of times to execute the commands contained in the loop block.

break

Optional parameter used to break the iterative process of a loop. *break* must be the name of a temporary variable (VAR). Setting the value of the variable to 1 during the execution of the loop will cause the loop to break at the beginning of its next iteration.

COMMANDS

Those commands that you want executed for the duration of the **LOOP**.

SET Commands

SET commands in a calc script are procedural. The first occurrence of a **SET** command in a calc script stays in effect until the next occurrence of the same **SET** command.

SET AGGMISSG ON | OFF ;

Specifies whether Essbase aggregates `#MISSING` values in the database. The default behavior of AGGMISSG is determined by the global setting for the database, as described in the *Database Administrator's Guide.*

SET CACHE HIGH | DEFAULT | LOW | OFF | ALL ;

Specifies the size of the calculator cache.

HIGH, DEFAULT, and LOW

Levels defining the size of the calculator cache.

OFF

Essbase does not use a calculator cache.

ALL

Essbase uses a calculator cache, even when you do not calculate at least one full sparse dimension.

SET CALCHASHTBL ON | OFF ;

Allows you to optimize the calculation of large, flat database outlines; for example, where one member has more than 5000 children.

ON

Turns on optimization the calculation of large, flat database outlines.

OFF

Turns off the optimization of large, flat dimensions.

Default setting is OFF. Change this setting using the `CALCOPTCALCHASHTBL=TRUE` in the `ESSBASE.CFG` file.

SET CLEARUPDATESTATUS AFTER | ONLY | OFF ;

Specifies when Essbase marks data blocks as *clean*. This *clean* status is used during Intelligent Calculation.

The data blocks in your database have a calculation status of either *clean* or *dirty*. When Essbase does a full calculation of your database, it marks the calculated data blocks as *clean*. When a data block is *clean*, Essbase will not recalculate the data block on subsequent calculations, provided that Intelligent Calculation is turned on.

To ensure the accuracy of your calculation results, consider carefully the effect of the **SET CLEARUPDATESTATUS AFTER** command on your calculation.

If you do *not* use **SET CLEARUPDATESTATUS**, Essbase does *not* mark calculated data blocks as *clean* when you calculate a subset of your database. Essbase marks data blocks as *clean* only on a full calculation (`CALC ALL;`) or when Essbase calculates all members in a single calculation pass through your database

AFTER

Essbase marks calculated data blocks as *clean*, even if you are calculating a subset of your database.

ONLY

Essbase marks the specified data blocks as *clean*, but does not actually calculate the data blocks. This does the same as AFTER, but disables calculation.

OFF

Essbase does not mark the calculated data blocks as *clean*. Data blocks are not marked as clean, even on a default calculation (`CALC ALL;`) of your database. The existing clean or *dirty* status of the calculated data blocks remains unchanged.

SET FRMLBOTTOMUP ON | OFF ;

Optimizes the calculation of formulas on sparse dimensions in large database outlines by turning on the bottom-up sparse formula calculation method.

ON

Turns on the bottom-up sparse formula calculation method.

OFF

Turns off the bottom-up sparse formula calculation method. The default setting is OFF. You can change this setting using `CALCOPTFRMLBOTTOMUP=TRUE` in the `ESSBASE.CFG` file.

SET LOCKBLOCK HIGH | DEFAULT | LOW ;

Specifies the maximum number of blocks that Essbase can fix (get addressability) concurrently when calculating a sparse member formula.

HIGH, DEFAULT, and LOW

> Levels defining the number of blocks that Essbase can fix concurrently.

SET MSG SUMMARY | DETAIL | ERROR | WARNS | INFO | NONE ;

Traces your calculations.

SUMMARY

> Displays calculation settings, and provides statistics on the number of:
>
> - Data blocks created, read, and written
> - Data cells calculated.

DETAIL

> Provides the same information as Summary. In addition, it displays a detailed information message every time Essbase calculates a data block.

ERROR

> Displays only error messages.

WARNS

> Displays warning and error messages.

INFO

> Displays only information, warning, and error messages.

NONE

> Displays no messages during the life of the calc script except error messages which may contain vital information.

SET NOTICE HIGH | DEFAULT | LOW ;

Monitors the progress of your calculation by providing completion notices at intervals during the calculation.

HIGH, DEFAULT, and LOW

> Levels defining the frequency and number of completion notices.
>
> You can set the values of HIGH, DEFAULT, and LOW using the **CALCNOTICE** parameter in the `ESSBASE.CFG` file. If you do not set the value of DEFAULT in the `ESSBASE.CFG` file, Essbase uses a default value of 10 which provides 10 completion messages at 10% intervals during the calculation.

SET UPDATECALC ON | OFF ;

Turns Intelligent Calculation on and off.

ON

> Essbase calculates only blocks marked as *dirty*. Dirty blocks include updated blocks and their dependent parents.

OFF

> Essbase calculates all data blocks, regardless of whether they have been updated.

The default setting is ON. You can change this default using the `UPDATECALC TRUE | FALSE` setting in the `ESSBASE.CFG` file.

SET UPTOLOCAL ON | OFF ;

Restricts aggregations to those parents with the same defined currency. The default is OFF.

VAR *varName* [= *value*] ;

Declares a temporary variable containing a single value.

varName

Name of the temporary variable.

value

Optional parameter that declares the data value.

Report Writer Commands

The Essbase Report Writer is a text-based editor that you can use to report on data in an Essbase database. You can combine Report Writer's selection, layout, and formatting commands to build a variety of reports.

This chapter includes the following:

- Table of Report Delimiters

  The < or {} delimiters are required for Report Writer commands. If you do not use a delimiter, Report Writer assumes that the command name is a member name.

- An alphabetic listing of commands.

Table 5, Report Delimiters

Delimiter	Use in Report Writer	Example
{}	Encloses report formatting commands	{**SUPFORMATS**}
<	Precedes layout and member sorting, selection, calculation, and some formatting commands	<**PAGE**

&*variableName*

& prefaces a substitution variable in the report script.

variableName

The name of the substitution variable set on the database. The variable name can be any combination of letters, numbers, and underscores up to 80 characters in length. Spaces, punctuation, and other special characters are not supported in the variable name.

!

Tells Essbase to output the instructions in the report script to the current line.

{**AFTER** *char* [*columnList*]}

Displays the first character of a designated character string following the data columns in the report.

char

Single character enclosed in quotes.

columnList

Optional list of one or more column numbers, separated by spaces. AFTER affects only these columns. If you do not specify *columnList*, all data columns are affected.

<**ALLINSAMEDIM** *mbrName*

Selects all the members from the same dimension as the specified dimension member for the report.

mbrName

Single member representing a dimension. All members from this dimension are selected.

Abbreviation: `<ALINSAME`

<**ALLSIBLINGS** *mbrName*

Adds all the siblings of the specified member to the report.

mbrName

Name of member whose siblings you want to add.

Abbreviation: `<ALLSIB`

<**ANCESTORS** *mbrName*

Adds all the ancestors of the specified member to the report.

mbrName

Name of member whose ancestors you want to add.

Abbreviation: `<ANCEST`

<**ASYM**

Causes a report to be printed in an asymmetric format.

{**BEFORE** "*char*" [*columnList*]}

Displays a character string before the data columns in the report.

char

Single character enclosed in quotes.

columnList

Optional. List of the column numbers, separated by spaces, that you want *char* to precede. Without *columnList*, *char* is displayed before all columns in the report.

{**BLOCKHEADERS**}

Displays all members that apply to a column as the column heading, in the style used by asymmetric reports.

Abbreviation: `{BLOCK}`

<**BOTTOM** ([<*rowgroupDimension*>,] <*rows*>, <*column*>)

Returns rows with the lowest values of a specified data column.

<*rowgroupDimension*>

Row grouping dimension that determines the rows to sort as a set. Optional.

<*rows*>

Positive integer number of rows to be returned; must be greater than 0.

<*column*>

@DATACOL (<*colnumber*>) |
@DATACOLUMN (<*colnumber*>)

where <*colnumber*> is the target column number; must be between 1 and the maximum number of columns in the report.

{BRACKETS}

Displays parenthesis around negative numbers instead of negative signs. The BRACKETS command need only be used to cancel the effect of a previously issued SUPBRACKETS command, as brackets are the default for negative numbers.

Abbreviation: {BRACK}

{**CALCULATE COLUMN** "*newColumn*" = *expression* }

Lets you create a new report column, perform on-the-fly calculations, and display the calculation results in the newly-created column.

"newColumn "

New column name, enclosed by quotation marks.

expression

Any valid column calculation expression.

The following mathematical operators are supported in column calculations:

+	Addition operator.
-	Subtraction operator.
*	Multiplication operator.
%X%Y	Evaluates X as a percentage of Y.
/	Division operator.
:X:Y	Performs a summation of data values from X to Y (inclusive). Must be the first operator if used with multiple operators.

Abbreviation: {CALCCOL}

{**CALCULATE ROW** "*newRow*" [*columnNo*] = *expression*} or {**CALCULATE ROW** "*newRow*" [*operator*]}

Lets you create a named report row and perform on-the-fly calculations.

newRow

Name of a new row, enclosed by quotation marks, that was declared with CALCROW, SAVEROW, or SAVEANDOUTPUT

columnNo

Optional. Designation of column numbers to which Essbase selects and applies the expression.

expression

Any valid row calculation expression.

operator

Any valid mathematical operator. The following mathematical operators are supported in row calculations:

+	Addition operator.
-	Subtraction operator.
*	Multiplication operator.
%X%Y	Evaluates X as a percentage of Y.
/	Division operator.
OFF	Turn off the row operator.

<**CHILDREN** *mbrName*

Selects all members in the level immediately below the specified member as defined in the database outline.

mbrName

Dimension or member name of the parent.

Abbreviation: `<CHILD`

{**CLEARALLROWCALC**}

Resets the value of all calculated rows to `#MISSING`.

Abbreviation: `{CLEARALLROW}`

{**CLEARROWCALC** *name*}

Resets the value of the row calculation *name* to `#MISSING`.

name

Name of a calculated row from a **CALCULATE ROW** command.

Abbreviation: `{CLEARROW}`

{**COLHEADING**}

Turns on automatic display of the column header, and sets it to be output prior to display of the next non-suppressed output data row.

Abbreviation: `{COLHEAD}`

<COLUMN (*dimList*)

Specifies the column or columns to display across the page.

dimList

> Dimension name or a comma-delimited list of dimensions.

Abbreviation: `<COL`

{COMMAS}

Displays commas for numbers greater than 999 after commas have been suppressed with either a **SUPCOMMAS** or **SUPALL** command.

{CURHEADING}

Enables the display of the currency conversion heading.

Abbreviation: `{CURHEAD}`

<CURRENCY *targetCurrency*

Performs on-the-fly currency conversions in a report.

targetCurrency

Currency and currency type to display in the report. Currency type is optional. Up to four members (at most, one from each currency database dimension) in a cross-dimensional member (->).

For example: `US$`, or `FFrancs->Actual->Jun93`

Abbreviation: `<CURR`

{**DECIMAL** *decPlaces* | VARIABLE [*columnN* [*columnN*]]}

Determines the number of decimal places to display in the report.

decPlaces

> Number of decimal places to display. Positive integer from 0 (the default) to 40. Specify either VARIABLE or *decPlaces*.

VARIABLE

> Allows the decimal to float; may switch to scientific notation (E+00 format) if necessary to display the significant digits of a number in the given column width. Specify either VARIABLE or *decPlaces*.

columnN

> Optional. List of columns, separated by spaces, to be affected by this command. If omitted, the command affects all data columns.

Abbreviation: `{DEC}`

<**DESCENDANTS** *mbrName*

Adds the descendants of the specified member to the report, excluding *mbrName* itself.

mbrName

> Name of parent of descendants to include.

Abbreviation: `<DESC`

<**DIMBOTTOM** *mbrName*

Includes the members at the bottom of the dimension, those members with no children (level 0), in the report.

mbrName

> Single member from the dimension for which you want to use the bottom-level members.

Abbreviation: `<DIMBOT`

<**DIMEND** *dimList*

Allows you to specify a dimension format that is processed after cycling through all the members in the dimension.

dimList

> List of members, separated by commas, that represents the dimensions for which the format is intended.

Abbreviation: `<DIMEND`

<**DIMTOP** *mbrName*

Adds the top of the dimension for the member to the report.

mbrName

> Single member from the dimension to designate.

Abbreviation: `<DIMTOP`

<DUPLICATE *mbrRange*

Allows a member name to occur more than once in a dimension group selection.

mbrRange

Specifies single member name or selection command.

- Single member: A member already selected for the dimension can be selected again.
- Selection command: <**DUPLICATE** applies to all members selected by *mbrRange*. For example, `<CHILDREN Accounts`.

Abbreviation: `<DUP`

{ENDHEADING}

Ends the definition of the custom page heading that is displayed at the top of each page in the report.

Abbreviation: `{END}`

{EUROPEAN}

Enables non-US formatting of numbers by switching commas and decimal points in the final report data values.

Abbreviation: `{EUR}`

{FEEDON}

Enables page breaks in a report when the number of output lines on a page is greater than the current **PAGELENGTH** setting.

The defaults are **FEEDON** and **PAGELENGTH** of 66 lines.

Abbreviation: `{FEED}`

{**FIXCOLUMNS** *number* }

Fixes the number of total columns in the final report regardless of how many actual columns are originally selected.

number

Number of columns that you want displayed in your final report.

Abbreviation: `{FIXCOL}`

{**FORMATCOLUMNS** *number* }

Expands the number of data columns when processed.

number

Expected number of columns that are encountered for formatting purposes.

Abbreviation: `{FORMATCOL}`

{**HEADING**}

Displays the page heading: either the default heading or the heading as defined with the **STARTHEADING** and **ENDHEADING** commands.

Abbreviation: `{HEAD}`

<**IANCESTORS** *mbrName*

Adds a member and its ancestors to the report.

mbrName

Single member whose ancestors you want to include.

Abbreviation: `<IANCEST`

<ICHILDREN *mbrName*

Selects the specified member and all members in the level immediately below it.

mbrName

Dimension or member name of the parent.

Abbreviation: `<ICHILD`

<IDESCENDANTS *mbrName*

Adds the specified member and its descendants to the report.

mbrName

Name of single member and descendants to add to the report.

Abbreviation: `<IDESC`

{IMMHEADING}

Forces the immediate display of the heading without waiting for the next non-suppressed data row.

Abbreviation: `{IMMHEAD}`

{INCEMPTYROWS}

Displays empty rows of data, or rows that contain only zeros or `#MISSING` data values, in the final report.

Abbreviation: `{INCEMPTY}`

{INCFORMATS}

Controls the formats affected by the following commands:

- SUPMASK
- SUPMISSING
- SUPZERO

Abbreviation: `{INCFORM}`

{INCMASK}

Re-includes (turns back on) the mask that has been suppressed by the command SUPMASK.

{INCMISSINGROWS}

Displays missing rows of data, or rows that contain all `#MISSING` data values, in the final report.

Abbreviation: `{INCMISS}`

{INCZEROROWS}

Includes rows that contain only data values of zero, in the final report.

Abbreviation: `{INCZERO}`

{**INDENT** [*offset*]}

Shifts the first row names column in column-output order by the specified number of characters.

offset

Optional. Number of spaces to indent column 0 from the left boundary of the name column. Values:

- Positive number (up to 100): Shifts column 0 to the right.
- Negative number: Shifts column left, but cannot indent to the left of the start of the name column.
- 0: Returns column to original position.
- Default (no value): Indents columns by 2.

{**INDENTGEN** [*offset*]}

Indents subsequent row members in the row names column based on the generation in the database outline.

offset

Optional number that determines the amount to indent each succeeding generation from the previous generation. The default is INDENTGEN -2.

<IPARENT *mbrName*

Adds the specified member and its parent to the report.

mbrName

A single member, which must not be the dimension's top member.

<LATEST *mbrName* or *reservedName* (*mbrName*)

Specifies a Dynamic Time Series member in a report script. You must create a Dynamic Time Series member in the database outline before you use it in a report script.

reservedName

One of the following pre-defined generation names:

History-To-Date (H-T-D)
Year-To-Date (Y-T-D)
Season-To-Date (S-T-D)
Period-To-Date (P-T-D)
Quarter-To-Date (Q-T-D)
Month-To-Date (M-T-D)
Week-To-Date (W-T-D)
Day-To-Date (D-T-D)

mbrName

The name of the Level 0 member in the Time dimension.

<**LINK** (*extractionCommand* [*operator extractionCommand*])

Uses the AND, OR, and NOT Boolean operators, combined with extraction commands, to refine member selections.

extractionCommand

Any of the following extraction commands or another AND/OR expression.

<ALLINSAMEDIM (*member*)
<ALLSIBLINGS (*member*)
<ANCESTORS (*member*)
<CHILDREN (*member*)
<DESCENDANTS (*member*)
<DIMBOTTOM (*member*)
<DIMTOP (*member*)
<GEN (*Dimension, generation_name*)
<IANCESTORS (*member*)
<ICHILDREN (*member*)
<IDESCENDANT (*member*)
<IPARENT (*member*)
<LEV (*Dimension, level_name*)
<MATCH (*Dimension, match_string*)
<MEMBER (*member*)
<OFSAMEGEN (*member*)
<ONSAMELEVELAS (*member*)
<PARENT (*member*)
<UDA (*Dimension, UDA_name*)

Operator

Any of the following Boolean operators:

- Use the AND operator when all conditions must be met.
- Use the OR operator when either one condition or another must be met.
- Use the NOT operator to choose the inverse of the selected condition.

{**LMARGIN** [*marginSize*]}

Sets the left margin for the report to *marginSize* characters.

marginSize

Optional numeric value: number of character spaces for left margin.

{**MASK** *charPosition "replacement"* [*charPosition "replacement"*]}

Overwrites the text in each output row with the specified characters at the specified position.

charPosition

Character position at which to start replacing text.

"replacement"

New text, in quotes, with which to overwrite the original output.

<**MATCH** ("*Member*" **|** "*Gen*" **|** "*Level*", "*Pattern*")

Performs wildcard member selections.

Member

Member name at the top of the member hierarchy you want to search. Essbase searches the member name and its descendents.

Gen

Default or user-defined name of the generation you want to search.

Level

Default or user-defined name of the level you want to search.

Pattern

The character pattern you want to search for, including a wildcard character (* or ?).

- ? Substitutes one occurrence of any character; can be placed anywhere in the string.
- * Substitutes any number of characters; must be used at the end of the string.
- You can include spaces in the character pattern. Ensure that you enclose the pattern in quotation marks ("").

{**MISSINGTEXT** ["*text*"]}

Substitutes a text label for missing values.

text

Optional text to use for missing values.

Abbreviation: `{MISSING}`

{**NAMESCOL** [*columnList* | CENTERED]}

Determines the location of the row names columns in the report.

columnList

Optional list, separated by spaces, of the locations for each row name. List position corresponds to the number of the affected column.

CENTERED (or C)

Key word that centers the column of row member names in the report. Before using this parameter:

- Define all columns in the report.
- Use the **FORMATCOLUMNS** command to set the number of columns.

Abbreviation: `{NAMESCOL}`

{**NAMESON**}

Turns on the display of column(s) of row member names.

{**NAMEWIDTH** [*width*]}

Determines the width of all row name columns in the report.

width

Optional. Specifies the total number of characters displayed for each column.

Abbreviation: `{NAMEW}`

{**NEWPAGE**}

Inserts a new page in the report regardless of how many lines have been generated for the current page.

{**NOINDENTGEN**}

Displays all row member names left-justified in the row names column without indenting members based on generation in the database outline.

Abbreviation: `{NOINDENT}`

{**NOPAGEONDIMENSION** *mbrName*}

Turns off insertion of a new page when the member in the report from the same dimension as *member* changes in a row of the report.

mbrName

Single member whose dimension is part of the **PAGEONDIMENSION** declaration.

Abbreviation: `{NOPAGE}`

{NOROWREPEAT}

Prevents row member names from being repeated on each line of the report if the row member name does not change on the next line. This is the default.

Abbreviation: `{NOROW}`

{NOSKIPONDIMENSION *mbrName*}

Prevents insertion of a new line when a member from the same dimension as *mbrName* changes in a row of the report.

mbrName

Single member that defines a dimension for which to halt line-skipping.

Abbreviation: `{NOSKIP}`

{NOUNAMEONDIM *mbrName*}

Turns off underlining for the new member name when the member in the report from the same dimension as the specified member changes in a row of the report.

mbrName

Member whose dimension is part of the **UNAMEONDIM** declaration.

{OFFCOLCALCS}

Disables all column calculations within the report.

Abbreviation: `{OFFCOL}`

{OFFROWCALCS}

Temporarily disables all row calculations.

Abbreviation: `{OFFROW}`

<OFSAMEGEN *mbrName*

Adds to the report all members from the same dimension and generation as the specified member.

mbrName

> Single member that designates the dimension and generation to retrieve.

{ONCOLCALCS}

Re-enables column calculations in the report after they have been disabled by **OFFCOLCALCS**.

Abbreviation: `{ONCOL}`

{ONROWCALCS}

Re-enables all row calculations after they have been disabled by **OFFROWCALCS**. Each subsequent row of data after using the command is calculated.

Abbreviation: `{ONROW}`

<**ONSAMELEVELAS** *mbrName*

Adds to the report all members on the same level as the specified member.

mbrName

Single member that designates the dimension and generation to retrieve.

Abbreviation: `<ONSAMEL`

{**ORDER** *columnList* }

Specifies the order of columns in a report.

columnList

Numeric designations of the columns to rearrange, separated by a space between each column number.

Each column number represents the *initial* positions of each column (from 0 to *n* where *n* is the last column—counting names, data, and calculated columns, respectively).

The position of each number in the *columnList* represents the new order in which you want the columns displayed.

<**ORDERBY** ([<*rowgroupDimension*>,] <*column*> [<*direction*>] {,<*column*> [<*direction*>]})

Orders the rows in a report according to data values in the specified columns.

<*rowgroupDimension*>

Row-grouping dimension that determines the rows to sort as a set. Optional.

<*column*>

@DATACOL (<*colnumber*>) |
@DATACOLUMN (<*colnumber*>)

where <*colnumber*> is the target column number; must be between 1 and the maximum number of columns in the report.

<*direction*>

You can specify multiple columns with different sorting directions, where:

ASC	is the ascending sort
DESC	is the descending sort

<OUTALT

Sets the output alias to the database outline alias name, as defined in the current alias table.

<OUTALTMBR

Sets the output alias to the database outline alias name (as defined in the current alias table) followed by the database outline member name.

Abbreviation: `<OUTALTM`

{OUTALTNAMES}

Displays alias names for members in a report.

Abbreviation: `{OUTALTN}`

<OUTALTSELECT *AliasTableName*

Selects an alias table in a report script.

AliasTableName

The name of the selected alias table associated with the database outline.

<OUTMBRALT

Sets the output name to the database outline member name followed by the outline alias, as defined in the current alias table.

Abbreviation: `<OUTMBRA`

{OUTMBRNAMES}

Reverts to the default member name display after the **OUTALTNAMES** command has been used to display alternate names.

Abbreviation: `{OUTMBRN}`

<OUTPUT

Causes Essbase to resume output, reversing the action of **SUPOUTPUT**.

<**PAGE** (*dimList*)

Defines which dimensions are displayed as page members in the final report.

dimList

Dimension name or a comma-delimited list of dimensions.

{**PAGEHEADING**}

Displays the page heading before the next data-output row.

Abbreviation: `{PAGEHEAD}`

{**PAGELENGTH** [*lines*]}

Sets the maximum number of lines for one page in the report.

lines

Optional total number of output lines for the size of paper you are using. Because the Report Writer does not recognize any font characteristics of the output report, it operates based on lines rather than inches.

Abbreviation: `{PAGELEN}`

{**PAGEONDIMENSION** *mbrName*}

Performs a page break whenever a member from the same dimension as the specified member changes from one line in the report to the next.

mbrName

Single member. If any member of the same dimension increments, a new page is started.

Abbreviation: `{PAGEON}`

<**PARENT** *mbrName*

Adds the parent of the member to the report.

mbrName

> Single member, which must not be the dimension (top) member.

Abbreviation: `<PAR`

{**PRINTROW** *"rowName"*}

Immediately displays the calculated *rowName* with its current values.

rowName

> Character string enclosed by quotation marks, which designates a previously declared calculated row. When the command is issued, the designated row is printed immediately in the report.

Abbreviation: `{PRINT}`

{**PYRAMIDHEADERS**}

Displays column members in centered, pyramid-shaped levels above columns (the default style used by symmetric reports).

Abbreviation: `{PYRAMID}`

<**QUOTEMBRNAMES**

Displays all the member names within double quotes in the report script output.

Abbreviation: `<QUOTE`

{**REMOVECOLCALCS**}

Removes all column calculation definitions from the report.

Abbreviation: `{REMOVECOL}`

{**RENAME** "*newMbrName*"} *mbrName*

Renames a member within the report.

newMbrName

Quote-enclosed valid member name to be used as the replacement name.

mbrName

Name of the member that you want to rename temporarily.

<**RESTRICT** (*<column | value> <operator> <column | value>* {*<logicalOperator> <column | value> <operator> <column | value>*})

Specifies the conditions that the row must satisfy before it becomes part of a result set.

<column>

@DATACOL (*<colnumber>*) |
@DATACOLUMN (*<colnumber>*)

where <colnumber> is the target column number; must be between 1 and the maximum number of columns in the report.

<value>

Cell data type (real number) | `#MISSING`

<operator>

>, >=	greater than, greater or equal
<, <=	less than, less than or equal
=	equal
!=, < >	not equal

<logicalOperator>

Report Writer processes logical operations from left to right without exception. Parentheses are not supported. The logical operators supported are AND and OR.

<**ROW** (*dimList*)

Determines the row dimensions for a report whose member names appear in the data rows of the report.

dimList

Dimension name or a comma-delimited list of dimensions.

{ROWREPEAT}

Displays all applicable row members on each row of the report even if a member describing a row is the same as in the previous row.

Abbreviation: `{ROWREP}`

{**SAVEANDOUTPUT** ["*rowCalcName*"]} *rowMbr*}

Adds *rowMbr* to the report, and creates a new calculated row whose default name is *rowMbr,* but which may be renamed with an optional name in quotes.

rowCalcName

Optional. Name, enclosed in quotation marks, for the calculated data row created by the **SAVEROW** command.

*C*an be multi-part, separated by a tilde (~), as in the **CALCULATE ROW** and **CALCULATE COLUMN** syntax.

rowMbr

Row member that determines the row name for the calculated data row.

Abbreviation: {SAVEANDOUT}

{**SAVEROW** *"newRowCalcName"*} *rowMbr*}

Creates a new calculated row whose default name is *rowMbr*, but which may be renamed with an optional name in quotation marks.

newRowCalcName

Optional. Name, enclosed by quotation marks, for the data row created by the **SAVEROW** command. The name can be multi-part, separated by a tilde (~), as in the **CALCULATE ROW** and **CALCULATE COLUMN** syntax.

rowMbr

Default row member used to determine the row name for the calculated data row.

rowMbr is the next member encountered after the {**SAVEROW**} command, so other intervening format commands ({}) or non-member-selecting commands (<) are allowed and do not affect which member is saved.

{**SCALE** *factor* [*columnList*]}

Scales the data in the report by multiplying it by a numeric value.

factor

Numeric value by which all output values are multiplied. Result: scaled value.

columnList

Optional. List of column numbers that this command affects.

{**SETCENTER** *charPosition*}

Sets a new centerline position on the page.

charPosition

Integer representing a character position on your page. Character position is counted from the left edge of the page, and is not affected by the left margin setting.

Abbreviation: {SETCEN}

{**SETROWOP** "*rowCalcName*" [*operation*]}

Lets you define on-the-fly calculations for a named row created with **CALCULATE ROW**.

rowCalcName

Named row, in double quotes, to which **SETROWOP** applies. You can use any valid row calculation expression.

operation

SETROWOP accepts the same mathematical operators as **CALCULATE ROW**. In addition, **SETROWOP** accepts the OFF operator, which turns off row operations for rows that follow.

<SINGLECOLUMN

Displays a column heading when there is only one column member extracted in the report.

Abbreviation: `<SINGLECOL`

{**SKIP** *n*}

Skips one or more lines in the output report.

n

Positive integer representing the number of lines to skip.

{**SKIPONDIMENSION** *mbrName*}

Inserts a blank line when a member from the same dimension as the specified member changes on the next line in the report.

mbrName

> Name of single member. When a member from this dimension changes during report processing, a blank line is inserted before the member change.

Abbreviation: `{SKIPONDIM}`

<SORTALTNAMES

Alphabetically sorts members by their alternate names within a member selection command (for example, `<CHILDREN`).

Abbreviation: `<SORTALTN`

<SORTASC

Specifies an ascending sort order.

<SORTDESC

Specifies a descending, hierarchical sort order.

<SORTGEN

Sorts members by outline generation number.

<SORTLEVEL

Sorts members by outline level number.

<SORTMBRNAMES

Sorts members alphabetically by name.

Abbreviation: `<SORTMBRN`

<SORTNONE

Disables all previous sorting commands.

<SPARSE

Tells Essbase to use the SPARSE data extraction method, which optimizes performance when a high proportion of the reported data rows are `#MISSING`. This data extraction method is different from the regular method. Essbase cannot use the sparse retrieval optimization method on Dynamic Calc or Dynamic Calc And Store members.

{STARTHEADING}

Starts the definition of the page heading in place of the default heading. The page heading is displayed at the top of each page in the report or immediately following a **HEADING** command.

Abbreviation: `{START}`

{SUPALL}

Suppresses the display of the page and column headings, member names, page breaks, commas, and brackets in the final report.

{SUPBRACKETS}

Suppresses the display of parenthesis around negative numbers.

Abbreviation: {SUPBRACK}

{SUPCOLHEADING}

Suppresses the display of the default column headings in the report.

Abbreviation: {SUPCOLHEAD}

{SUPCOMMAS}

Suppresses the display of commas in numbers greater than 999.

{SUPCURHEADING}

Suppresses the display of currency information when you use the **CURRENCY** command to convert the data values in your report to a specified currency.

Abbreviation: {SUPCURHEAD}

{SUPEMPTYROWS}

Suppresses the display of rows that have only 0 or #MISSING values in the row.

Abbreviation: {SUPEMPTY}

{SUPEUROPEAN}

Disables the European method for displaying numbers.

Abbreviation: {SUPEUR}

{SUPFEED}

Suppresses the automatic insertion of a page break whenever the number of lines on a page exceeds the current **PAGELENGTH** setting.

{SUPFORMATS}

Used to suppress formats that produce output such as underlines and skips.

Abbreviation: `{SUPFORM}`

{SUPHEADING}

Suppresses the display of the default heading (page header and column headers) or custom header, if defined, at the top of each page.

Abbreviation: `{SUPHEAD}`

{SUPMASK}

Suppresses the display of a text mask.

{SUPMISSINGROWS}

Suppresses the display of all rows that contain only `#MISSING` values.

Abbreviation: `{SUPMISS}`

{SUPNAMES}

Suppresses the display of row member names in the final report.

{SUPOUTPUT}

Suppresses all output while continuing to process all operations, such as calculations, format settings, and so forth. Use the **OUTPUT** command to resume output.

Abbreviation: {SUPOUT}

{SUPPAGEHEADING}

Suppresses the display of the page member heading whenever a heading is generated.

Abbreviation: {SUPPAGE}

<SUPSHARE

Suppresses the display of duplicate shared members when you use generation or level names to extract data for your report.

<SUPSHAREOFF

Reinstates the display of duplicate shared members after they have been suppressed using the **SUPSHARE** command.

{SUPZEROROWS}

Suppresses the display of rows that have only 0 values in the row in the report.

Abbreviation: {SUPZERO}

<SYM

Forces a symmetric report, regardless of the data selection. Use **SYM** to change the symmetry of a report that Essbase would, by default, create as an asymmetric report.

{**TABDELIMIT**}

Places tabs rather than spaces between columns.

{**TEXT** *charPosition* "*text*" [*charPosition* "*text*"...]}

Adds descriptive text to a report.

charPosition

Character position on the line to start *text*. Successive *charPositions* need not be in ascending order.

The only effect order has is that if two text strings' positions would cause an overlap, the last would overwrite the first. "Last" is determined by left-right order in the **TEXT** statement, not by *charPosition*.

text

Text to add to the report.

<**TOP** ([<*rowgroupDimension*>,] <*rows*>, <*column*>)

Returns rows with the highest values of a specified data column.

<*rowgroupDimension*>

Row grouping dimension that determines the rows to sort as a set. Optional. The default is the inner row.

<*rows*>

A positive integer that specifies the number of rows to be returned; must be greater than 0.

<*column*>

@DATACOL (<*colnumber*>) |
@DATACOLUMN (<*colnumber*>)

where <*colnumber*> is the target column number; must be between 1 and the maximum number of columns in the report.

{**UCHARACTERS** ["*char*"]}

Underlines all non-blank characters in the preceding row in the report.

char

Optional single character, enclosed by quotation marks, used as the underline character.

Abbreviation: `{UCHAR}`

{**UCOLUMNS** ["*char*"]}

Underlines all the columns, including both names and data, in the preceding row in the report.

char

Optional. Single character, enclosed by quotation marks, that creates an underline character.

Abbreviation: `{UCOL}`

<**UDA** (*dimName*, *udaString*)

Selects members based on a common attribute defined as a user-defined attribute (UDA) on the server.

dimName

Name of the dimension associated with *udaString*.

udaString

Name of the user-defined attribute as it appears in the database outline.

{**UDATA** ["*char*"]}

Underlines all the data columns for a row in a report, while not underlining the row name columns.

char

Optional single character, enclosed by quotation marks, used as the underline character.

{**UNAME** ["*char*"]}

Underlines all the row name columns in the preceding row while not underlining the data columns.

char

Optional single character, enclosed by quotation marks, used as the underline character.

{**UNAMEONDIMENSION** *mbrName*}

Underlines the row member names in a row whenever a member from the same dimension as the specified member changes.

mbrName

Single member representing a dimension. When a new member from this dimension is output, an underline appears under all row names in the previous line.

Abbreviation: `{UNAMEONDIM}`

{**UNDERLINECHAR** ["*character*"]}

Sets the default underline character displayed when underlining any information in a report.

character

> Single character, enclosed by quotation marks, for the new underline character.

Abbreviation: `{UNDERLINE}`

{**UNDERSCORECHAR** *"char"*}

Replaces the _ (underscore) character in a member name with another character.

char

> Single character, enclosed by quotation marks, to display in place of underscore.

Abbreviation: `{UNDERSCORE}`

{**WIDTH** *number* [*column1* [*column2* [*columnN*]]]}

Specifies the width of columns in a report.

number

> New column width in characters.

column1, column2, columnN

> Optional. Numbers that designate the columns to resize, separated by spaces. Values: between 2 and 161.

{**ZEROTEXT** ["*text* "]}

Replaces zero data values with a text string if a zero data value is output.

text

> Optional. Text string, in quotes, to use in place of the 0 value.

Abbreviation: `{ZERO}`

ESSCMD Commands

Your Arbor Essbase OLAP Server includes ESSCMD and its Windows and Macintosh versions, ESSCMDW and ESSCMDM, respectively. ESSCMD is a command-line interface that performs operations interactively or through a batch or script file.

This chapter includes an alphabetic listing of commands.

ADDUSER *groupName userName*

Adds a user to a group.

groupName

Name of an existing group.

userName

Name of an existing user.

APPLYOTLCHANGEFILE *numFiles fileName*

Applies the outline changes specified in the `.CHG` log file to the target database outline. Use this command after calling the **GETPARTITIONOTLCHANGES** command, which creates the log file for the **APPLYOTLCHANGEFILE** command.

numFiles

A numeric value indicating the number of `.CHG` log files to read.

fileName

The name of the `.CHG` log file to read. If more than one file exists, enter each file name at the command prompt.

ARCHIVE

This command is retained for backward compatibility. When you run the **ARCHIVE** command, Essbase returns an informational message. Use the **BEGINARCHIVE** and **ENDARCHIVE** commands.

See **BEGINARCHIVE** and **ENDARCHIVE**.

BEGINARCHIVE *App Db file*

Places a database in Read-only mode for archiving.

App

Name of the application to archive.

Db

Name of the database to archive.

file

Name of the file to contain the archive.

BEGININCBUILDDIM

Prepares Essbase for upcoming incremental, dimension-building commands. Used in conjunction with the **ENDINCBUILDDIM** command together with one or more **INCBUILDDIM** statements.

BUILDDIM *location rulobjName dataLoc sourceName fileType errorLog*

Use this command to dynamically build one or more dimensions from a data file or SQL source.

location

Location of the rules object. Values:

1 Local/client-based rules object

2 Remote/server rules object

3 File

Use option 3 if you want to specify the full path name. Otherwise, Essbase looks in the `appname/dbname` directory.

rulobjName

Name of the rules object.

dataLoc

Location of the data file. Values:

1 Local/client data file

2 Remote/server data file

3 File

4 SQL source

Use 3 if you want to specify the full path name. Otherwise, Essbase looks in the `appname/dbname` directory.

sourceName

Source of the data file. Value:

- If dataLoc is 1, 2, 3, specify the data file name.
- If dataLoc is 4, specify the SQL user name and password.

fileType

Data file type. Values:

1	Excel file
2	Lotus `.WK1` file
3	Lotus `.WK3` file
4	Text file
5	Lotus `.WK4` file

This parameter is not required if you are using an SQL source.

errorLog

Name of the text file to receive error messages and rejected records.

CALC "*calcString*; [*calcString*;]"

Executes one or more calculation strings.

calcString

A calculation string (any valid string that is accepted by a calc script).

CALCDEFAULT

Calculates the default database calculation.

CALCLINE *calcString*

Executes a single calculation string.

calcString

A calculation string (any valid string accepted by a calc script).

COMMITDB

Retained for backward compatibility; no longer used. When the Transaction Isolation Level is Committed access, Essbase handles commits automatically. When the Transaction Isolation Level is Uncommitted access, you can specify a commit threshold. See **SETDBSTATEITEM** for more information.

COPYAPP *sourceApp destApp*

Copies an application.

sourceApp

Name of the application to copy.

destApp

Name of the application copy.

COPYDB *sourceApp sourceDb destApp destDb*

Copies a database.

sourceApp

Name of the application that includes the database to copy.

sourceDb

Name of the database to copy.

destApp

Name of the application for the new database.

destDb

Name of the database copy.

COPYFILTER *sourceApp sourceDb sourceFilter destApp destDb destFilter*

Copies a filter.

sourceApp

Name of the application that includes the filter to copy.

sourceDb

Name of the database that includes the filter to copy.

sourceFilter

Name of the filter to copy.

destApp

Name of the application for the new filter.

destDb

Name of the database for the new filter.

destFilter

Name of the filter copy.

COPYOBJECT *objType sourceApp sourceDb sourceObj destApp destDb destObj*

Copies a database object.

objType

Type of object to list. Values:

0	Abort
1	Outline object (not available)
2	Calc script
3	Report script
4	Rules object
5	Alias table
6	Structure file
7	Backup file (not available)
8	Worksheet of any type (not available)
9	Text object
10	Partition
11	Linked Reporting Object
12	Selection
13	Wizard

sourceApp

Name of the application that includes the object to copy.

sourceDb

Name of the database that includes the object to copy.

sourceObj

Name of the object to copy.

destApp

Name of the application for the new object.

destDb

Name of the database for the new object.

destObj

Name of the object copy.

CREATEAPP *appName*

Creates a new application.

appName

Name of the application to create.

CREATEDB *appName dbName*

Creates a database.

appName

Name of the application in which to create a database.

dbName

Name of the database to create.

CREATEGROUP *groupName*

Creates a group.

groupName

Name of the group to create.

CREATEUSER *userName password*

Creates a new user ID.

userName

Name of the user ID to create.

password

Password for the new user.

CREATEVARIABLE *variableName serverName* [*appName* [*dbName*]] *value*

Defines a new substitution variable and its corresponding string value on the server. A variable can be defined at the database application, or server level.

variableName

The name to give to the new substitution variable. The name must be alphanumeric and can contain a maximum of 80 characters. You can use underscores, but not spaces.

serverName

Name of the server for which the variable is defined.

appName

Optional. Name of the application for which the variable is defined.

dbName

Optional. Name of the database for which the variable is defined.

value

The string value that corresponds to the substitution variable set on the server. The name must be alphanumeric and can contain a maximum of 255 characters. It can include a null value. Do not use the & character as the leading character in the value string.

DEFRAG

This command is retained for backward compatibility with previous Essbase versions. It is no longer used; issuing this command has no effect on Essbase files.

DELETEAPP *appName*

Deletes an application.

appName

Name of the application to delete.

DELETEDB *appName dbName*

Deletes a database.

appName

Name of the application containing the database to delete.

dbName

Name of the database to delete.

DELETEGROUP *groupName*

Deletes a group.

groupName

Name of the group to delete.

DELETELOG *appName*

Deletes an application log file or the Agent log file.

appName

Name of application.

If you do not specify *appName*, Essbase deletes the Agent log file(s).

DELETEUSER *userName*

Deletes a user ID.

userName

Name of the user to delete.

DELETEVARIABLE *variableName serverName* [*appName* [*dbName*]]

Removes an existing substitution variable.

variableName

Name of the substitution variable to delete.

serverName

Name of the server from which the variable is deleted.

appName

Optional. Name of the application from which the variable is deleted.

dbName

Optional. Name of the database from which the variable is deleted.

DISABLELOGIN [*appName*]

This command is used to prevent users from logging in to an application.

appName

Optional. Application name required only if no application is selected.

DISPLAYALIAS *aliasTableName*

Lists the alias names defined in an alias table.

aliasTableName

Name of the alias table to display.

ENABLELOGIN [*appName*]

Enables connections to an application.

appName

Optional. Application name required only if no application selected.

ENDARCHIVE *appName dbName*

Restores the database to Read-write mode after archiving is complete.

appName

Name of the application containing the archived database.

dbName

Name of the archived database.

ENDINCBUILDDIM *preserve*

Ends the programming block started by **BEGININCBUILDDIM** and restructures the database after one or more incremental, dimension-building (**INCBUILDDIM**) commands.

preserve

Specifies whether to preserve existing data in the database. This parameter is required. Values:

1	Preserves all existing data blocks.
2	Preserves existing level 0 data.
3	Preserves existing input-level data.
4	Discards all existing data.

EXIT

Exits the ESSCMD utility.

EXPORT *exportName amount*

Dumps the data values of a database to an ASCII text file.

exportName

Name of the file to create to receive the exported data.
Number representing the data to export. Values:

1 Export all data.

2 Export only level 0 blocks.

3 Export only data from blocks with input data.

amount

Number representing the data to export. Values:

1 Export all data

2 Export only level 0 blocks

3 Export only data from blocks with input data

GETALLREPLCELLS *sourceServerName sourceAppName sourceDbName*
or
GETALLREPLCELLS ALL

Replicates all data cells in the replicated partition from a data source database to the selected data target database. Use this command when you are in the data target database.

sourceServer

Name of the data source server from which cells are replicated.

sourceApp

Name of the data source application from which cells are replicated.

sourceDb

Name of the data source database from which cells are replicated.

ALL

Updates cells for all partitions where the selected database is a data replication target.

GETAPPACTIVE

Returns the name of the currently selected application.

GETAPPINFO [*appName*]

Returns server, user, and database information for the current application.

appName

Optional. Required only if no application is selected.

GETAPPSTATE [*appName*]

Returns information on the state of the currently selected application.

appName

Optional. Required only if no application is selected.

GETCRDB

Returns the name of the currency database that is linked to the currently selected database.

GETCRDBINFO

Returns information about the currency database that is linked to the currently selected database.

GETCRRATE

Returns the currency rate for every currency partition.

GETCRTYPE

Returns information about the default currency type and conversion method.

GETDBACTIVE

Returns the name of the currently selected database.

GETDBINFO [*appName dbName*]

Returns information on the state of the currently selected database.

appName dbName

Optional. Both parameters required if no application and database are selected.

GETDBSTATE [*appName dbName*]

Returns information on the state of the currently selected database.

appName dbName

Optional. Both parameters required if no application and database are selected.

GETDBSTATS

Returns detailed statistics for the currently selected database.

GETDEFAULTCALC

Returns the default calc script of the currently selected database.

GETMBRCALC *mbrName*

Returns the calc string associated with the selected member.

mbrName

Member name

GETMBRINFO *mbrName*

Returns information on a specific member.

mbrName

Member name

GETMEMBERS *mbrString*

Returns a list of members from the currently selected database.

mbrString

Optional. Dimension or member name. If specified, returns the children of the named dimension or member. The default is NULL. Returns a list of top-level members from the database.

GETPARTITIONOTLCHANGES *sourceServerName sourceAppName sourceDbName sourcePartitionType getAllOtlChanges* [*getAllDimChanges* [*getNewDim getDeletedDim getUpdatedDim getMovedDim getRenamedDim*] *getAllMbrChanges* [*getNewMbrs getDeletedMbrs getRenamedMbrs getMovedMbrs*] *getAllMbrAttribChanges* [*getChngedMbrStatus getChngedMbrAlias getChngedMbrCalcSym getChngedMbrAcctType getChngedMbrCurrCnvInfo getChngedMbrUda getChngedMbrCalcFormulas*] *getChangedLevNbr getChangedGenNbr*]

Note: All arguments must be provided on one line.

Retrieves a list of outline changes made to the partitioned area in the source database, and writes these changes to the `.CHG` file on the target database you select with the SELECT command.

sourceServerName

Name of the data source server where the outline changes were made.

sourceAppName

Name of the data source application where the outline changes were made.

sourceDbName

Name of the data source database where the outline changes were made.

sourcePartitionType

Name of the partition type where the outline changes were made. Can be any of the following:

1. Replicated
2. Transparent
3. Linked

getAllOtlChanges

Lists all changes to the database outline. Values Y / N.

getAllDimChanges

Lists all changes to the dimensions, including member names. Values Y / N.

getNewDim

Lists newly created dimensions. Values Y / N.

getDeletedDim

Lists deleted dimensions. Values Y / N.

getUpdatedDim

Lists updated dimensions. Values Y / N.

getMovedDim

Lists moved dimensions.Values Y / N.

getRenamedDim

Lists renamed dimensions. Values Y / N.

getAllMbrChanges

Lists all member changes. Values Y / N.

getNewMbrs

Lists newly created members. Values Y / N.

getDeletedMbrs

Lists deleted members. Values Y / N.

getRenamedMbrs

Lists renamed members. Values Y / N.

getMovedMbrs

Lists moved members. Values Y / N.

getAllMbrAttribChanges

Lists all changes to member attributes. Values Y / N.

getChngedMbrStatus

Lists members that have a changed status such as data storage or Dynamic Time Series information. Values Y/N.

getChngedMbrAlias

Lists changed member aliases. Values Y/N.

getChngedMbrCalcSym

Lists changed member unary operators. Values Y/N.

getChngedMbrAcctType

Lists changed account type information for members in an Accounts dimension. Values Y/N.

getChngedMbrCurrCnvInfo

Lists changed member currency conversion information. Values Y/N.

getChngedMbrUda

Lists changed member user-defined attributes. Values Y/N.

getChngedMbrCalcFormulas

Lists changed member calc formulas. Values Y/N.

getChangedLevNbr

Lists changed level numbers. Values Y/N.

getChangedGenNbr

Lists changed generation numbers. Values Y/N.

GETUPDATEDREPLCELLS *sourceServerName sourceAppName sourceDbName*
or
GETUPDATEDREPLCELLS ALL

Replicates all changed data cells in the replicated partition from a data source database to the selected data target database. Use this command when you are in the data target database.

sourceServerName

Name of the data source server from which cells are replicated.

sourceAppName

Name of the data source application from which cells are replicated.

sourceDbName

Name of the data source database from which cells are replicated.

ALL

Updates cells for all partitions where the selected database is a data replication target.

GETUSERINFO *userName*

Returns information about a specified user or group.

userName

Name of user or group.

GETVERSION

Returns the version number information on the current Arbor Essbase server software installation.

GOTO "*Label* ";

<*skipped commands* >
:*Label* ;
<*commands* or *EOF* >

Skips all commands until it encounters the associated label.

"*Label*"

A string of ASCII characters; not case-sensitive.

:*Label*

Target location, preceded by a colon (:) and associated with "Label." Processing skips to this label.

HELP ? | *commandName*

Provides online help on a specific command or lists all available commands.

?

All available commands.

commandName

Name of a specific command.

IFERROR "*Label* ";

<*skipped commands* >
:*Label* ;
<*commands* or *EOF* >

Checks the status returned by a command, and either continues processing or branches to the associated label in response to the status.

"*Label*"

String of ASCII characters, terminated by a whitespace; not case-sensitive.

:*Label*

Target location, preceded by a colon (:) and associated with “Label.” Processing skips to this label.

IMPORT *numeric dataFile fileType* Y/N *ruleLoc rulobjName* Y/N [*ErrorFile*]

For an SQL data source, the syntax is as follows:

IMPORT 4 *SQLUserName SQLUserPassword ruleLoc rulobjName* Y/N [*ErrorFile*]

Loads data values from an external source into the currently selected database.

numeric

Location of the *dataFile* file. Values:

1	Local/client data file
2	Remote/server data file
3	File
4	SQL source

dataFile

Name of data source file.

fileType

File type of *dataFile*. Values:

1	Excel file
2	Lotus 2 file
3	Lotus 3 file
4	Text file
5	Lotus 4 file

Y/N

Whether to use rules when importing *dataFile*.

ruleLoc

Location of the *rulobjName* file. Values:

1 Local/client rule object file

2 Remote/server rule object file

3 File

Use option 3 if you want to specify the full path name. Otherwise Essbase looks in the `appname/dbname` directory.

rulobjName

Name of the rules file.

Y/N

Whether to abort on error.

SQLUserName

User name that connects to the SQL database.

SQLUserPassword

User password for the SQL database.

ErrorFile

The name of the error file. This is required only if you choose not to abort on error.

INCBUILDDIM *location rulobjName dataLoc sourceName fileType errorLog appendLog*

Incrementally build one or more dimensions from a data file, without restructuring the database. This command provides better performance than **BUILDDIM** whenever you need to do multiple dimension builds from multiple rules files and your database contains data.

location

Location of the rules file. Values:

1 Local/client-based rules object

2 Remote/server rules object

3 File

Use option 3 if you want to specify the full path name. Otherwise, Essbase looks in the *`appname/dbname`* directory.

rulobjName

Name of the rules file.

dataLoc

Location of the data file. Values:

1 Local/client data file

2 Remote/server data file

3 File

4 SQL source

Use option 3 if you want to specify the full path name. Otherwise, Essbase looks in the *`appname/dbname`* directory.

sourceName

Source of the data file.

- If *dataLoc* is 1 or 2, specify the data file name.
- If *dataLoc* is 3, specify the data file name and path.
- If *dataLoc* is 4, specify the SQL user name and password.

fileType

Data file type. Values:

1	Excel file
2	Lotus `.WK1` file
3	Lotus `.WK3` file
4	Text file
5	Lotus `.WK4` file

This parameter is not required if you are using an SQL source.

errorLog

Name of text file to receive error messages and rejected records. Each **INCBUILDDIM** command in a **BEGININCBUILDDIM**...**ENDINCBUILDDIM** block can specify a different error log.

appendLog

Specifies whether to append to the error log file or overwrite it. Values:

1	Append
2	Overwrite

LISTALIASES

Returns a list of alias tables that are defined for the currently selected database.

LISTAPP

Returns a list of applications that are defined on the server.

LISTDB

Returns a list of databases that are contained within the currently selected application.

LISTFILTERS *appName dbName*

Lists the filters in a database.

appName

Name of the application containing the filters.

dbName

Name of the database containing the filters.

LISTGROUPS

Returns a list of groups that are defined on the server.

LISTGROUPUSERS *groupName*

Returns a list of users that belong to a specified group.

groupName

Group for which to list users.

LISTLINKEDOBJECTS *userName modDate*

Lists information about the objects linked to the active database for a given user name or modification date.

userName

The name of a user. If specified, Essbase returns a list of all objects last modified by the given user.

modDate

A modification date. If specified, Essbase returns a list of all objects modified on or before the given date.

LISTLOGINS

Returns the list of login instances in your session.

LISTLOCKS [*appName dbName*]

Returns a list of all users who have locks on blocks for the currently selected database.

appName dbName

Optional. Both parameters required if no application and database are selected.

LISTMEMBERS *filename*

Reads a `.STR` file (an exported Essbase outline file). Use this command only when working with Version 2.x or earlier outlines.

filename

Name of `.STR` file to read.

LISTOBJECTS *number appName dbName*

Returns a list of objects.

number

Type of object to list. Values:

0	Abort
1	Outline
2	Calc script
3	Report script
4	Rules object
5	Alias table
6	Structure file
7	Backup file
8	Worksheet of any type
9	Text object
10	Partition
11	Linked Reporting Object (stored)
12	Selection
13	Wizard

appName

Name of the application containing the objects.

dbName

Name of the database containing the objects.

LISTUSERS

Returns a list of the users that are defined on the server.

LISTVARIABLES *serverName* [*appName* [*dbName*]]

Lists all substitution variables defined for a specified server.

serverName

Name of the server for which the variable is defined.

appName

Optional. Name of the application for which the variable is defined.

dbName

Optional. Name of the database for which the variable is defined.

LOADALIAS *aliasName fileName*

Loads an alias table to the currently selected database.

aliasName

Name of the alias table to load.

fileName

Name of the feeder file that loads into the table. Must be in the format required to import an alias table.

LOADAPP *appName*

Used to load an application and its respective databases into memory.

appName

Name of the application to load.

LOADDATA *numeric fileName*

Loads data without a rules file.

numeric

Location of the data file. Values:

1 Local/client-based rules object (file)

2 Remote/server data file

3 File

Use option 3 if you want to specify the full path name. Otherwise, Essbase looks in the *`appname/dbname`* directory.

fileName

Name of the file to load.

LOADDB *appName dbName*

Loads a database into memory.

appName

Name of the application in which the database resides.

dbName

Name of the database to load.

LOGIN *hostNode userName password* [*appName*] [*dbName*]

Used to log in to the server.

hostNode

Server name.

userName

User ID defined on the Essbase server.

password

User’s password.

appName

Optional. Name of the application to load.

dbName

Optional. Name of the database to load.

LOGOUT

Used to log out of the server.

LOGOUTALLUSERS Y/N

Logs out all users from the Essbase server.

Y / N

Sets whether users are logged out.

LOGOUTUSER *userNumber*

Logs a specific user off the server.

userNumber

Login ID number associated with a user. Issue **LOGOUTUSER** with no parameter to display a list of users and user number.

OUTPUT *outputType* [*outputName* / *errorName*]

Directs process information output from the ESSCMD session to a text file.

outputType

Number representing output operation. Values:

1	Outputs all process information
2	Outputs only errors
3	Stops output of process information
4	Stops output of errors

outputName

Required for *outputType* 1 only. Name of file to receive output.

Not used with other values for *outputType*.

errorName

Required for *outputType 2* only. Name of file to receive errors.

Not used with other values for *outputType*.

PRINTPARTITIONDEFFILE *location* [*ddbFileName*] *textFileName*

Produces an ASCII text file of the distributed database's partition mapping tables.

location

Possible values:

1 Local / client file with a `.DDB` file extension that is stored in the directory pointed to by `ARBORPATH`. The *ddbFileName* is automatically retrieved.

2 Remote / server `.DDB` file. The *ddbFileName* is automatically retrieved.

3 Local / client file not stored in the `ARBORPATH`, or without a `.DDB` file extension. The *ddbFileName* is required when using this option.

ddbFileName

The partition mapping definition `.DDB` file name to read information from. This is usually the name of the database, such as `BASIC.DDB`.

If location is 1 or 2, `.DDB` file name is not required.

If location is 3, the full path, file name, and file extension of the file is required.

textFileName

The full path, file name, and file extension of the ASCII text output file to create.

PURGELINKEDOBJECTS *userName modDate*

Deletes objects linked to the active database for a given user name or modification date.

userName

The name of a user. If *userName* is specified, Essbase deletes all objects last modified by the given user.

modDate

A modification date. If *modDate* is specified, Essbase deletes all objects modified on or before the given date.

PURGEOTLCHANGEFILE *serverName appName dbName partitionType direction*

Deletes outline changes that have already been applied from the `.CHG` log file.

serverName

Name of the server from which to delete `.CHG` information

appName

Name of the application from which to delete `.CHG` information

dbName

Name of the database from which to delete `.CHG` information

partitionType

The partition type that the deletions are applied to:

1. Replicated
2. Transparent
3. Linked

direction

Values:

Source	The selected database is used as a data source for the replicated, transparent, or linked partition.
Target	The selected database is used as a data target for the replicated, transparent, or linked partition.

PUTALLREPLCELLS *targetServerName targetAppName targetDbName* or **PUTALLREPLCELLS** ALL

Replicates all data cells in a replicated partition from the data source database you selected with the SELECT command, to a specified data target database. Use this command when you are in the data source database.

targetServerName

Name of the data target server to which cells are replicated.

targetAppName

Name of the data target application to which cells are replicated.

targetDbName

Name of the data target database to which cells are replicated.

ALL

Updates all cells in partitions where the selected database is a data replication source.

PUTUPDATEDREPLCELLS *targetServerName targetAppName targetDbName*
or
PUTUPDATEDREPLCELLS ALL

Replicates all changed data cells in the replicated partition from the data source database you selected with the **SELECT** command, to the specified data target database. Use this command when you are in the data source database.

targetServerName

Name of the data target server to which changed cells are replicated.

targetAppName

Name of the data target application to which changed cells are replicated.

targetDbName

Name of the data target database to which changed cells are replicated.

ALL

Updates all changed cells in all partitions where the selected database is a data replication source.

REMOVEUSER *groupName userName*

Removes a user from a group.

groupName

Name of group from which to remove user.

userName

Name of the user to remove.

REMOVELOCKS *userNumber*

Removes any locks that a specified user has acquired through a spreadsheet operation.

userNumber

Logon ID of user for whom you are removing locks.

RENAMEAPP *sourceApp newAppName*

Renames an existing application.

sourceApp

Name of the existing application.

newAppName

New name for the application.

RENAMEDB *sourceApp sourceDb newDbName*

Renames an existing database.

sourceApp

Name of the application that contains the database to be renamed.

sourceDb

Name of the database to be renamed.

newDbName

New name for the database.

RENAMEFILTER *sourceApp sourceDb sourceFltr newFltrName*

Renames an existing filter.

sourceApp

Name of the application that includes the filter.

sourceDb

Name of the database that includes the filter.

sourceFltr

Name of the existing filter.

newFltrName

New name for the filter.

RENAMEOBJECT *objType sourceApp sourceDb sourceObj newObjName*

Renames an existing object.

objType

Type of object to rename. Values:

0	Abort
1	Outline object (not available)
2	Calc script
3	Report script
4	Rules object
5	Alias table
6	Structure file
7	Backup file, not available
8	Worksheet of any type (not available)

9	Text object
10	Partition
11	Linked Reporting Object (stored)
12	Selection
13	Wizard

sourceApp

Name of the application that includes the object.

sourceDb

Name of the database that includes the object.

sourceObj

Name of the existing object.

newObjName

New name for the object.

RENAMEUSER *userName newUserName*

Renames an existing user.

userName

Name of the existing user.

newUserName

New name for the user.

REPORT *reportString*

Used to execute one or more report strings.

reportString

One or more report strings.

REPORTLINE *reportString*

Used to execute a single report string.

reportString

Report string.

RESETDB

Clears all the data from the currently selected database.

RESETOTLCHANGETIME *fromPartition toPartition*

Changes the time on the server you selected with the **SELECT** command, to match the time on another server.

fromPartition

Includes the following parameters:

serverName Name of the server from which the time change is applied

appName Name of the application from which the time change is applied

dbName Name of the database from which the time change is applied

partitionType The name of the type of partition from which the time change is applied:

1. Replicated
2. Transparent
3. Linked

direction Values:

Source—The selected database is used as a data source for the replicated, transparent, or linked partition.

Target—The selected database is used as a data target for the replicated, transparent, or linked partition.

toPartition

Includes the following parameters:

serverName Name of the server to get the time change

appName Name of the application to get the time change

dbName Name of the database to get the time change

partitionType The name of the type of partition the time change is applied to:

1. Replicated
2. Transparent
3. Linked

direction Values:

Source—The selected database is used as a data source for the replicated, transparent, or linked partition.

Target—The selected database is used as a data target for the replicated, transparent, or linked partition.

RESETSTATUS

Resets all saved status values to 0 (zero).

RESTORE

No longer used. Retained for backward compatibility. When you run the **RESTORE** command, Essbase returns an informational message. See **BEGINARCHIVE** and **ENDARCHIVE**.

RUNCALC *numeric calcScript*

Runs a calc script.

numeric

Location of the calc script data file. Values:

1	Local/client-based calc script
2	Remote/server calc script
3	File

Use option 3 if you want to specify the full path name. Otherwise, Essbase looks in the `appname/dbname` directory.

calcScript

Name of the calc script to run.

RUNREPT *numeric reptScript outputFile*

Runs a report script.

numeric

Location of the report script file. Values:

1	Local/client-based report script
2	Remote/server report script
3	File

Use option 3 if you want to specify the full path name. Otherwise, Essbase looks in the *`appname/dbname`* directory.

reptScript

Name of the report script to run.

outputFile

Target file name for report output.

SELECT *appName dbName*

Used to select an application and database.

appName

Name of the application containing the desired database.

dbName

Name of database within the selected application.

SETALIAS *aliasName*

Sets an alias table as the primary table.

aliasName

Name to set for the alias table.

SETAPPSTATE ["*appName*"] "*desc*" Y/N Y/N *accessLevel* Y/N Y/N Y/N Y/N *lockTimeout MaxLROFileSize* ;

Defines application settings.

appName

Name of the application. Do not include *appName* if the active application is selected.

desc

Text string describing the application.

Y/N

Sets whether the application is loadable.

Y/N

Sets whether autoload occurs.

accessLevel

Default access level. Values:

0	None
1	Read
2	Write
3	Calculate
4	Application or Database Designer

Y/N

Sets whether connections can be made.

Y/N

Sets whether commands can be issued.

Y/N

Sets whether updates can occur.

Y/N

Sets whether security is enabled.

lockTimeout

Maximum number of seconds that locks can be placed on blocks by Spreadsheet Add-in users.

MaxLROFileSize

Maximum size, in kilobytes, for a Linked Reporting Objects (LRO) file.

SETDBSTATE ["*appName*"] ["*dbName*"] "*desc*" Y/N Y/N *accessLevel* *dataCacheSize* Y/N Y/N Y/N *currDb ccType* 0/1 *indexCacheSize* *IndexPageSize* Y/N ;

Defines database settings. For more options, see **SETDBSTATEITEM**.

Note: It is most efficient to load the required database before you run the **SETDBSTATE** command, then stop and restart the database for the command to take effect.

appName

Name of the application. Not required if the active application is selected.

dbName

Name of the database; required if *appName* is used.

desc

Text string describing the database.

Y/N

Sets whether the database is loadable.

Y/N

Sets autoload on or off.

accessLevel

Default access level. Values:

0	None
1	Read
2	Write
3	Calculate
4	Database Designer

dataCacheSize

Maximum amount of server memory allocated for data cache. Default: 3145728 bytes.

Y/N

Sets whether to aggregate missing values.

Y/N

Sets whether to perform a Two-Pass calc.

Y/N

Sets whether to create blocks on equations.

currDb

Links a currency database.

ccType

Specifies the default currency type member.

0/1

Sets the conversion method. Values:

0	Division
1	Multiplication

indexCacheSize

Maximum amount of server memory allocated for index cache. Default: 1048576 bytes.

indexPageSize

Maximum amount allocated for index page. Value: integer between 1024 and 8192 bytes.

Y/N

Enable (Y) or disable (N) data compression on disk.

SETDBSTATEITEM [*optionNumber*] [*appName*] [*dbName*] [*value(s)*]

Defines database settings by number, and provides more options than **SETDBSTATE**.

Note: It is most efficient to load the required database before you run the **SETDBSTATE** command, then stop and restart the database for the command to take effect.

optionNumber

An integer between 0 and 26, inclusive. This number corresponds to the options listed below.

appName

Name of the application. Not required if the active application is selected.

dbName

Name of the database; required if *appName* is used.

value(s)

Acceptable value or values; these vary from option to option.

If you specify the *optionNumber* or follow the prompts, the following value(s) are:

0

All options. Prompts for each option in turn. Use only in interactive mode.

1

Description. Text string describing the database.

2

Allow database to start? (Sets whether the database is loadable.) Values: Y / N.

3

Start database with application? (Sets autoload on or off.) Values: Y / N.

4

Default access level. Values:

0	None
1	Read
2	Write
3	Calculate
4	Database Designer

5

Data cache size: The maximum size of a buffer in memory that holds data blocks for the current operation. Default and minimum: 3145728B (3 megabytes).

6

Sets whether to aggregate missing values. Values: Y/N.

7

Sets whether or not to perform a Two-Pass calc. Values: Y/N.

8

Sets whether to create blocks on equations. Values: Y/N.

9

Currency database name. Links a currency database that you specify.

10

Specifies the default currency conversion type member.

11

Sets the conversion method. Values:

0	Division
1	Multiplication

12

Index cache size: Maximum size of a memory buffer that holds index pages for the current operation. Default: 1048576 bytes (1 megabyte).

13

Index page size: The size of an index page (all pages in a database are the same size). Acceptable values: 1024B, 2048B, 4096B, or 8192B.

14

Enables (Y) or disables (N) data compression on disk.

15

Data compression type. Values:

1 Run-Length Encoding

2 Bitmap (the default)

16

Retrieval buffer size. Specifies the size of the internal server sorting buffer that holds extracted row data cells before they are evaluated by the **RESTRICT** or **TOP/BOTTOM** Report Writer command. This buffer is used by the Report Writer and the Retrieval Wizard. The default is 10K.

17

Retrieval sort buffer size. Specifies the size of the internal server sorting buffer that holds data until it is sorted. This buffer is used by the Report Writer and the Retrieval Wizard. The default is 10K.

18

Isolation level. Choose Committed or Uncommitted access to your database. Committed access provides better data integrity. Uncommitted access provides consistency with Essbase Version 4. Values:

1 Committed access

2 Uncommitted access (the default)

Depending on which type of access you specify, ESSCMD prompts you for other parameters (or you can supply the values on the command line).

If you choose 1 (Committed access), ESSCMD prompts for:

- Pre-image access. See item 19.
- Time Out. See item 20.

If you choose 2 (Uncommitted access), ESSCMD prompts for:

- Number of blocks modified before internal commit. The default is 3000.
- A value of 0 means no implicit commit; Essbase commits blocks at the end of the transaction.
- Number of rows to data load before internal commit. The default is 0, no implicit commit; Essbase commits blocks at the end of the transaction.

19

Pre-image access. Valid for Committed access only. Provides users Read-only access to data blocks that are locked for the duration of another transaction. Users see the last committed data values for those data blocks. If you choose N (No), your transaction waits for the blocks to become available, or Essbase issues a time-out error. Values: Y/N. The default is N (No).

20

Time Out: the length of time, in seconds, to wait to acquire a lock on data blocks that are locked by another transaction. Acceptable values are:

-1	Indefinite wait
0	Immediate access, or no wait
n	A number of seconds that you specify.

21

Number of blocks modified before internal commit. The default is 3000. See item 18.

22

Number of rows to data load before internal commit. The default is 0. See item 18.

23

Add disk volume definition(s). Use this parameter if you want to allocate storage across multiple volumes or restrict space used on a volume. For information on disk volumes, see the *Database Administrator's Guide*.

ESSCMD prompts you for the following values, unless you supply them on the command line:

- Number of new disk volume definitions.
  The default is 1.
- Volume name or drive letter (required).
- Volume size (maximum space to use on that volume). The default is unlimited (0). The minimum is 8 megabytes.

You can specify this value in bytes (B), kilobytes (K), megabytes (M, the default), or gigabytes (G).

- File type(s) to be stored on this volume:

  1 Index files only

  2 Data files only

  3 Index and data files (the default)

- File size: the maximum size that each index or data file can attain before Essbase creates a new file. The default is 2G. The minimum is 8 megabytes.

You can specify this value in bytes (B), kilobytes (K), megabytes (M, the default), or gigabytes (G).

24

Change the Disk Volume settings on an allocated volume. This command prompts you for several values. See item 23.

25

Stop Essbase from storing additional files on an allocated volume. This command prompts you for:

Volume Definition - *n*

where *n* is the number corresponding to the disk volume definition you want to remove. For example, suppose you defined three volumes: first, C; then, E; then, D. Essbase considers D the third volume - definition number 3.

26

Abort. Returns you to the ESSCMD command line. Use only in interactive mode.

SETDEFAULTCALC *calcString*

Sets a calculation string as the default database calculation.

calcString

Calculation string to set.

SETDEFAULTCALCFILE *calcobjName*

Sets a calculation object as the default database calculation.

calcobjName

Calculation object to set. Give full path name if this object is not in the `CLIENT` directory.

SETHELP *numberLines*

Changes the number of lines of ESSCMD help that the screen displays.

numberLines

Number of lines to display in ESSCMD help.

SETLOGIN *sesNo*

Sets the active login to a particular instance.

sesNo

Login instance session number. Values:

prev	Previous number
next	Next session number
sessionNo	Integer representing session

SETMSGLEVEL *level*

Defines the level of messages that are generated by the server.

level

Level setting for messages. Values:

1	Make no changes.
2	Display all information messages.
3	Display only warning messages.
4	Display only error messages.
5	Display no messages.

SETPASSWORD *userName newPassword*

Assigns a new password to an existing user.

userName

Name of the existing user.

newPassword

New password for the user.

SHUTDOWNSERVER *servername username password*

Lets you shut down the Agent from any terminal connected to it. You must have Supervisor privilege to use this command.

servername

Name of the server you want to shut down.

username

Your user name.

password

Your password.

SLEEP "*seconds*"

Pauses an ESSCMD script.

seconds

Number of seconds for the batch file execution to sleep.

UNLOADALIAS *aliasName*

Unloads the existing alias table.

aliasName

Name of the alias table to unload.

UNLOADAPP *appName*

Unloads an application from memory on the server.

appName

Name of the application to unload.

UNLOADDB *appName dbName*

Used to unload a database from memory on the server.

appName

Name of the application in which the database resides.

dbName

Name of the database to unload.

UNLOCKOBJECT *objType sourceApp sourceDb sourceObj*

Unlocks an existing object.

objType

Type of object to list. Values:

1	Outline object
2	Calc script
3	Report script
4	Rules object
5	Alias table (not available)
6	Structure file (not available)
7	Backup file (not available)
8	Worksheet of any type (not available)
9	Text object
10	Partition
11	Linked Reporting Object (stored)
12	Selection
13	Wizard

sourceApp

Name of the application that includes the object.

sourceDb

Name of the database that includes the object.

sourceObj

Name of the existing object to unlock.

UPDATE *dataString*

Loads a single data record.

dataString

A single data record.

UPDATEFILE *location fileName update*

Loads data, unlocks blocks, and verifies a data file.

location

Location of the data file. Values:

1 Local/client-based rules object

2 Remote/server data object

3 File

Use option 3 if you want to specify the full path name. Otherwise, Essbase looks in the *appname/dbname* directory.

fileName

Name of the file to load

update

Update action. Values:

1 Load data

2 Unlock data blocks

3 Verify data

UPDATEVARIABLE *variableName* [*serverName* [*appName* [*dbName*]]] *value*

Updates the variable value that corresponds to the specified substitution variable.

variableName

The name of the existing substitution variable.

serverName

Optional. Name of the connection server to which the variable is applied.

appName

Optional. Name of the application to which the variable is applied.

dbName

Optional. Name of the database to which the variable is applied.

value

The new string value that corresponds to the substitution variable. The name must be alphanumeric, and can be a maximum of 255 characters.You can have a null value, but do not use a leading & character in the value.

VALIDATE [*errorlogFile*]

Checks the database for data integrity.

errorlogFile

Optional. Name of destination file for error messages. Default: List file in *`application\database`* directory. For example: `ESSBASE\APP\app\db\VALIDATE.LST`

VALIDATEPARTITIONDEFFILE

Validates shared partition definitions.

ESSBASE.CFG File Settings

The `ESSBASE.CFG` (`essbase.cfg` on UNIX platforms) file lets you customize your Essbase configuration. You can create separate `ESSBASE.CFG` files at the server and client levels, each accepting different settings. This chapter includes an alphabetic listing of settings available in the server file and client file.

Server File Settings

AGENTDELAY *n*

Specifies the number of seconds an Agent thread will wait to perform a specific action.

n

The number of seconds the Server Agent waits before performing a specific action. *n* must be an integer and must be 5 or higher. The default value is 20.

AGENTTHREADS *n*

Specifies how many threads the Server Agent may spawn.

n

The number of threads the Server Agent may produce. *n* is an integer between 2 and 500, inclusive.

The default value is 5. It is strongly recommended that you use this default value.

CALCCACHE TRUE | FALSE

Specifies whether Essbase uses a calculator cache when calculating your database.

TRUE

Tells Essbase to use a calculator cache when calculating the database. The default is TRUE.

FALSE

Tells Essbase not to use a calculator cache when calculating the database.

By default, **CALCCACHE** is set to TRUE.

CALCCACHEHIGH | CALCCACHEDEFAULT | CALCCACHELOW *n*

Sets the HIGH, DEFAULT and LOW values for the calc script **SET CACHE** command.

HIGH

The maximum calculator cache size a user can choose to use during calculation.

DEFAULT

The default calculator cache size.

LOW

The minimum calculator cache size a user can choose to use during calculation.

n

An integer value for each level, representing the size of the calculator cache in bytes.

If you do not set the value of DEFAULT, Essbase uses a default value of 200,000 bytes.

The maximum calculator cache size that you can specify is 200,000,000 bytes.

CALCLOCKBLOCKHIGH | CALCLOCKBLOCKDEFAULT | CALCLOCKBLOCKLOW *n*

Sets the HIGH, DEFAULT and LOW values for the calc script **SET LOCKBLOCK** command.

This setting specifies the maximum number of blocks that Essbase can fix (get addressability to) when calculating one block. HIGH, DEFAULT, and LOW are levels.

HIGH

> The maximum number of blocks that a user can choose to fix concurrently when one data block is calculated.

DEFAULT

> The default number of blocks that can be fixed concurrently.

LOW

> The minimum number of blocks that a user can choose to fix concurrently.

n

> An integer value for each level, representing the total number of blocks that can be fixed concurrently.

CALCHASHTBLMEMORY *MemSize*

Sets the Calculator hash table memory limit.

MemSize

> The maximum amount of memory (in bytes) that you want the Calculator hash table to use. The default is 20 megabytes.

CALCNOTICEHIGH | CALCNOTICEDEFAULT | CALCNOTICELOW *n*

Sets the HIGH, DEFAULT, and LOW values for the calc script **SET NOTICE** command.

The **SET NOTICE** command monitors the progress of your calculation, displaying completion notices during the calculation. HIGH, DEFAULT, and LOW are levels.

HIGH

> The maximum number of completion notices a user can choose to display.

DEFAULT

> The default number of completion notices.

LOW

> The minimum number of completion notices a user can choose to display.

n

> An integer value for each level, representing the number of notices to be displayed at set intervals during the calculation.

CALCOPTCALCHASHTBL TRUE | FALSE

Specifies whether Essbase optimizes the calculation of large, flat database outlines.

TRUE

> Optimizes the calculation of large, flat database outlines; for example, where one member has more than 5000 children.

FALSE

> Does not optimize the calculation of large, flat dimensions. The default is FALSE.

CALCOPTFRMLBOTTOMUP TRUE | FALSE

Specifies whether Essbase optimizes the calculation of formulas on sparse dimensions in large database outlines. To do this, Essbase uses the bottom-up sparse formula calculation method.

TRUE

> Optimizes the calculation of formulas on sparse dimensions in large database outlines, using the bottom-up sparse formula calculation method.

FALSE

> Does not use the bottom-up sparse formula calculation method. The default is FALSE.

CASESENSITIVE TRUE | FALSE

Determines whether member names are case-sensitive.

TRUE

> Member names are case-sensitive.

FALSE

> You need not pay attention to case when entering a member name. The default is FALSE.

CCTRACK TRUE | FALSE

Modifies the behavior of currency conversion calculations.

TRUE

> Essbase tracks exchange rates as conversions are calculated. The default value is TRUE.

FALSE

> The tracking system is turned off.

CLEARLOGFILE TRUE | FALSE

Determines whether Essbase overwrites log files.

TRUE

Essbase overwrites the application and server log files.

FALSE

Essbase appends to the existing log files. The default setting is FALSE.

COMMITBLOCKS

No longer used in `ESSBASE.CFG`. To specify this setting, choose Database | Settings in the Application Manager and click the Transaction tab, or enter SetDbStateItem in ESSCMD.

DATACOMPRESS

No longer used in `ESSBASE.CFG`. To enable data compression, choose Database | Settings in the Application Manager and click the Storage tab, or enter SetDbStateItem in ESSCMD. See the online help for those items for more information.

DATACOMPRESSIONTYPE

No longer used in `ESSBASE.CFG`. To enable data compression, choose Database | Settings in the Application Manager and click the Storage tab, or enter SetDbStateItem in ESSCMD. See the online help for those items for more information.

DATAERRORLIMIT *n*

Determines the number of records that can be written to an error log file during a data load operation.

n

The number of records, per data load, that can be written to the error log file. The default setting is 1000. The maximum setting is 1000.

DISKVOLUMES

No longer used in ESSBASE.CFG. To set or change Disk Volumes, choose Database | Settings in the Application Manager and click the Storage tab, or enter SetDbStateItem in ESSCMD. See the online help for those items for more information.

EXCLUSIVECALC TRUE | FALSE

Determines whether Essbase allows concurrent calculations.

TRUE

If a calculation operation (command or script) is running, Essbase fails any other calculation operations.

FALSE

Essbase allows concurrent calculation operations. The default value is FALSE.

EXCEPTIONLOGOVERWRITE TRUE | FALSE

Determines where Essbase writes application and server exception error log data.

TRUE

Essbase overwrites the existing error log. The default value is TRUE.

FALSE

Essbase keeps the existing error log and creates new logs for every exception.

INCRESTRUC *appname dbname* TRUE | FALSE

INCRESTRUC specifies whether Incremental Restructuring is enabled for a database. You can enable Incremental Restructuring for individual databases, for all databases in an application, or for all databases on a server.

appname

Application name. Optional parameter for enabling Incremental Restructuring for one or all databases in an application. This parameter may be used in combination with the *dbname* parameter. If you omit *appname,* you cannot specify *dbname,* and INCRESTRUC will be enabled for all applications and databases.

dbname

Database name. Optional parameter for enabling Incremental Restructuring for an individual database. This parameter must be used in combination with the *appname* parameter. If you specify *dbname,* you must also specify *appname*.

TRUE

When you make certain outline or dimension changes that normally result in immediate database restructuring, Essbase defers restructuring until the next time it accesses the affected block(s).

FALSE

> Essbase immediately restructures the database whenever an outline or dimension change calls for it. The default is FALSE (for all databases).

You can use a wildcard, *xxxxx*, to indicate “all” for any application or database argument. For example:

```
INCRESTRUC xxxxx Basic TRUE
```

enables incremental restructuring for any application with a Basic database.

Settings for nonexistent applications or databases are ingored.

You can issue up to ten (total) INCRESTRUC statements per application. The settings are applied according to their order of appearance in the `ESSBASE.CFG` file.

INDEXCACHESIZE

No longer used in `ESSBASE.CFG`. To set or change the index cache size, choose Database | Settings in the Application Manager and click the Storage tab, or enter SetDbStateItem in ESSCMD. See the online help for more information.

INDEXPAGESIZE

No longer used in `ESSBASE.CFG`. To set or change the index page size, choose Database | Settings in the Application Manager and click the Storage tab, or enter SetDbStateItem in ESSCMD. See the online help for more information.

LOCKTIMEOUT *n*

Limits the amount of time a user can hold an exclusive lock.

n

A number of seconds.

The default value is 3600 seconds (60 minutes).

NETDELAY *n*

Specifies the network request delay time.

n

An integer value of 100 or above, expressed in milliseconds.

The default value is 200 milliseconds.

NETNPCONNECTTIMEOUT *n*

Limits the amount of time a Named Pipes network continues to reattempt connection after an unsuccessful connect attempt.

n

An integer value, expressed in seconds.

The default value is 30 seconds.

NETRETRYCOUNT *n*

Specifies the number of times the network will retry a connection.

n

An integer value for the number of retries.

The default value is 600 retries.

The minimum value is 300.

NODENAME *server_name*

Identifies the server where your Essbase application is running.

server_name

The name of the server where your Essbase application is running.

NODENAME uses the current server by default.

NUMERICPRECISION *n*

Defines the number of precision digits that will be used by the Essbase Report Writer and Retrieval Wizard for numerical comparison.

n

The number of precision digits to be considered in the numerical comparison. Acceptable values for *n* are -1 through 15. A value of -1 causes Essbase to perform a full comparison.

The default value is 4.

OUTLINECHANGELOG TRUE | FALSE

Logs all outline changes into the file *`database_name`*`.OLG`. **OUTLINECHANGELOG** allows database administrators to review the outline revision history and gather enough information to roll back changes if needed.

TRUE

Essbase logs outline changes into the file *`database_name`*`.OLG`.

FALSE

Essbase does not log outline changes. The default is FALSE.

OUTLINECHANGELOGFILESIZE *n*

A maximum file size setting in bytes for the Outline Change Log file. When the Outline Change Log reaches the maximum file size, Essbase copies the contents of the file to a separate backup file with the same name as the Outline Change Log file (*`database_name`*`.OLG`), but with an .OLB extension.

n

Number of bytes to allocate for the change log. The default is 64000 bytes. The minimum file size is 8092 bytes. The maximum file size is 2 megabytes. Each time the Outline Change Log reaches its maximum file size, Essbase replaces the backup file with a backup of the current Outline Change Log.

PIPEBUFFERSIZE *n*

Defines the size of the buffer used for communication between the Arbor Essbase Spreadsheet Extractor and the Report Writer.

n

> An integer value from 2,048 to 65,534, expressed in bytes. The default value is 4K (4,096 bytes).

REPTKBYTEBUF

No longer used in `ESSBASE.CFG`. This setting is now called Retrieval Buffer Size and must be set on a per-database basis. To set or change this setting, choose Database | Settings in the Application Manager, or enter SetDbStateItem in ESSCMD. See the online help for those items for more information.

REPTKBYTESORTBUF

No longer used in `ESSBASE.CFG`. This setting is now called Retrieval Sort Buffer Size and must be set on a per-database basis. To set or change this setting, choose Database | Settings in the Application Manager, or enter SetDbStateItem in ESSCMD. See the online help for those items for more information.

SERVERTHREADS *n*

Specifies the number of threads the Server Application may spawn.

n

The number of threads the Server Application may produce. *n* is an integer between 20 and 50, inclusive.

The default value depends on the number of available ports. It is strongly recommended that you use the default value.

The default value of **SERVERTHREADS** depends on the number of available ports on the server.

Number of Ports	Default number of Threads
5 or fewer	5
6 – 20	10
21 and over	20

SSAUDIT *audit_appname audit_dbname log_path*
or
SSAUDITR *appname audit_dbname log_path*

SSAUDIT and **SSAUDITR** enable spreadsheet update logging. The two settings are identical in function except that **SSAUDITR** automatically clears update log files after archiving, and **SSAUDIT** appends log files after archiving. The **ENDARCHIVE** command signals **SSAUDITR** to clear the log files.

audit_appname

Application name

audit_dbname

Database name

log_path

Directory where you want this information stored

Default behavior:

- If **SSAUDIT** is not specified at all, spreadsheet update logging is not enabled.
- If **SSAUDIT** is issued with no arguments, Essbase activates spreadsheet logging for all databases in all applications on the server, and puts the log in the default directory:

`ARBORPATH\app\audit_appname audit_dbname`

You can use a wildcard, *xxxxx*, to indicate “all” for any argument.

You can issue up to ten (total) **SSAUDIT** and **SSAUDITR** statements per application.

SSLOGUNKNOWN TRUE | FALSE

Controls whether Essbase logs error messages when it encounters an unknown member name during a spreadsheet operation.

TRUE

Essbase displays and logs an error message for each unknown member name that it encounters during a spreadsheet operation. The default is TRUE.

FALSE

Essbase does not display these error messages.

SSPROCROWLIMIT *n*

Specifies the maximum number of spreadsheet rows the server will process on a user spreadsheet request.

n

An integer value between 16384 and 500000, inclusive.

The default value is 250000.

TIMINGMESSAGES TRUE | FALSE

Controls whether Essbase logs the duration of each spreadsheet and report query in the application log file.

TRUE

Essbase logs the duration of each spreadsheet and report query in the application log file. The default is TRUE.

FALSE

Essbase does not log the duration of spreadsheet and report queries in the application log file.

UPDATECALC TRUE | FALSE

Specifies whether Intelligent Calculation is turned on or off by default.

TRUE

> Intelligent Calculation is turned on. Essbase calculates only updated blocks and their dependent parents.

FALSE

> Intelligent Calculation is turned off. Essbase calculates all data blocks, regardless of whether they have been updated.

Client File Settings

NETDELAY *n*

Specifies the network request delay time.

n

An integer value of 100 or above, expressed in milliseconds.

The default value is 200 milliseconds.

NETNPCONNECTTIMEOUT *n*

Limits the amount of time a Named Pipes network continues to reattempt connection after an unsuccessful connect attempt.

n

An integer value, expressed in seconds.

The default value is 30 seconds.

NETRETRYCOUNT *n*

Specifies the number of times the network will retry a connection.

n

An integer value.

The default value is 600 retries.

The minimum value is 300; the maximum is 232.