Development- X User Defined Constants

User Defined Constants published by Development-X allow you to control the default behaviour of Ostendo. Development-X makes use of these to increase the functionality of Ostendo. These can act like switches in Ostendo to control how Ostendo can process or display information. New constants are added over time as Development-X continues to make use of this approach to provide further flexibility and functionality to Ostendo without the need to publish official Updates. By using this approach, we can control behaviour in one database different to that in another. **Please note that the Constant Names are case sensitive.**

Module	Description	Constant Name	Constant Type	What This Constant Controls
Jobs	Use Template Code for Order Description	**Use Template Code for Order Description**	**Yes/No**	This constant allows you to control the default Job Description that is automatically pre-filled when a job is created from the 'Service Orders Required' Screen. By default the Job Description copies the *Job Template Description* from the Asset as the Job Description. ***<u>Yes (True)</u>*** Ostendo will use the Job Template **Code** as the Job Description when the job is automatically created ***<u>No (False)</u>*** Ostendo will use the Job Template **Description** as the Job Description when the job is automatically created (*NB: This is default behaviour even if this constant is not defined*)
Jobs	Controlling Item / Descriptor Planned Costs when copying a Job	**REFRESH ITEM COST ON JOB COPY** and/or **REFRESH DESCRIPTOR COST ON JOB COPY**	**Yes/No**	This User Defined Constant modifies the behaviour when copying from one job to another. By default Ostendo will copy the originating job item and descriptor planned costs to the new job. (Labour Code planned costs are always refreshed with the current cost). There are some environments where they wish to always refresh all these costs when a job is copied. These constants will allow for this.
Jobs	Use Current Date for Job Line Actual Deletions	**USE CURRENT DATE FOR JOB ACTUAL DELETIONS**	**Yes/No**	This User Defined Constant controls the date of the reversing transaction when Job Line Actuals (Issues) are deleted. If set to True, this will date the reversing transaction with that of the current system date. By default, Ostendo will date the reversing transaction one date beyond the current Financial Cutoff Date if the originating transaction was dated prior to the current cutoff date. If it was dated after the current cutoff date then the originating issue date will be used.
Assembly	Use Planned Cost for Issue	**Use Assembly Planned Cost For Issue**	**Yes/No**	This constant allows you to control the Actual Issue Cost of a Descriptor on an Assembly Order. By default Ostendo issues descriptors using the Buy Price value. This constant allows you to issue the Descriptor at the Assembly Line Planned Cost instead. ***NB: Any Purchase Invoices (with a Style of "Invoice Only") posted directly to the Assembly Order will be posted at the Actual Purchase Invoice Cost. Further to this, this rule will only apply to Assembly Order Lines where the Planned Qty is Zero.*** ***<u>Yes (True)</u>*** Ostendo will use the Assembly Order Line Planned Cost as the Actual Issue Cost for Descriptors ***<u>No (False)</u>*** Ostendo will use the Buy Price as the Actual Issue Cost for Descriptors (*NB: This is default behavior even if this constant is not defined*)

Development- X User Defined Constants

User Defined Constants published by Development-X allow you to control the default behaviour of Ostendo. Development-X makes use of these to increase the functionality of Ostendo. These can act like switches in Ostendo to control how Ostendo can process or display information. New constants are added over time as Development-X continues to make use of this approach to provide further flexibility and functionality to Ostendo without the need to publish official Updates. By using this approach, we can control behaviour in one database different to that in another. **Please note that the Constant Names are case sensitive.**

Module	Description	Constant Name	Constant Type	What This Constant Controls
Assembly	Use Current Date for Assembly Line Actual Deletions	**USE CURRENT DATE FOR ASSEMBLY ISSUE DELETIONS**	**Yes/No**	This User Defined Constant controls the date of the reversing transaction when Assembly Line Issues are deleted. If set to True, this will date the reversing transaction with that of the current system date. By default, Ostendo will only allow Assembly Issues to be deleted if the originating Issue was dated on or after the current cutoff date.

Module	Description	Constant Name	Constant Type	What This Constant Controls
Assembly	Use Feature Groups Lookup in Inventory Lists	**List Feature Groups**	**Yes/No**	This constant allows you to control the lookup screen on the Inventory Lists Lines screen. By default you would setup a Group against the List and then select that group from the Lines screen. This option allows you to alternatively lookup Feature Groups instead, so they are able to be selected against List Lines. (Feature Groups are setup from Inventory->Settings->Feature Group) ***Yes (True)*** Ostendo will use allow you to lookup 'Feature Groups' from the Group field in the Inventory List Lines screen. ***No (False)*** Ostendo will use the Inventory List 'Group' as the lookup for the Group field on the Inventory List Lines screen (*NB: This is default behavior even if this constant is not defined*)

Module	Description	Constant Name	Constant Type	What This Constant Controls
POS	Control the Displayed Cost on the POS Edit Line Screen	**POS Line Display Cost (AVERAGE,LAST,STANDARD,BUY)**	**Text**	By default, Ostendo will display the cost on the POS Lines screen (If that Employee is set to see costs) using the Inventory Costing Cost Method specified in the System Settings screen. This User Defined Constant allows you to control which costing method will be used to display Costs on the POS Line Edit screen. ***Value: (NB: Type one of the following. This is Case Sensitive) AVERAGE, LAST, STANDARD, BUY***

Module	Description	Constant Name	Constant Type	What This Constant Controls
Sales	Default Reversal Date for Sales Delivery Picks	**USE CURRENT DATE FOR SALES PICKED DELETIONS**	**Yes/No**	This User Defined Constant modifies the behaviour when Picked Sales Delivery Lines are deleted. By default, Ostendo will date the reversing transaction with the same date as the original pick record. If this constant is used, the reversing transaction will be dated the current system date. Standard Ostendo behaviour prevents the deletion of a pick record where the date of that pick record is before the current Cut Off dates

Development- X User Defined Constants

User Defined Constants published by Development-X allow you to control the default behaviour of Ostendo. Development-X makes use of these to increase the functionality of Ostendo. These can act like switches in Ostendo to control how Ostendo can process or display information. New constants are added over time as Development-X continues to make use of this approach to provide further flexibility and functionality to Ostendo without the need to publish official Updates. By using this approach, we can control behaviour in one database different to that in another. **Please note that the Constant Names are case sensitive.**

Module	Description	Constant Name	Constant Type	What This Constant Controls
Financial	ACCTSWVERSION - Used by Software Integration Script	**ACCTSWVERSION**	**Text**	Do not amend this Constant. This constant is managed automatically by the Ostendo Accounting Integration script. Each time the accounting posting script is run, it checks the actual Accounting Software Product and Version and updates this User Defined Constant if it has changed since the last posting. Apart from capturing general information about the accounting software product, where MYOB AR Live 2020.2 or beyond is used, Ostendo controls the number of Invoice Number characters able to be used in Ostendo and therefore passed to MYOB. Previous versions of MYOB (prior to 2020.2) only allowed 8 character sales invoice numbers, whereas 2020.2 allows 13 character invoice numbers. Therefore in this case Ostendo allows this increase in characters when the AACTSWVERSION has a value of 2020.2

Module	Description	Constant Name	Constant Type	What This Constant Controls
Financial	Summarisation Of Sales and Purchases Invoice Lines	**Summarise Purchase Invoice Lines** and/or **Summarise Sales Invoice Lines**	**Yes/No** **Yes/No**	Summarisation Of Sales and Purchases Invoice Lines The purpose of this is to reduce the volume of line records being posted to the Accounting Software. For example, if we have a 50 line invoice, each line is presently posted to the Accounting Software. If this enhancement is turned on, Ostendo would summarise the Posting Journal down to one line per Cost Centre / Tax Code. At some sites, this would dramatically reduce data volumes being posted across to the Accounting Software. Summarisation of Sales Invoice Lines can be configured independently to Purchase Invoice Lines thereby providing a level of flexibility. eg: Sales Lines could be summarised whereas Purchase Invoice Lines might be left as detailed. **Points To Be Aware Of:** Summarisation will lose the itemised detail of the invoice lines eg: Item Codes, Descriptions, Qty's and Ext pricing Sales and Purchase Invoices will always posted immediately to the Accounting Software based on your posting frequency set in the System Master. These transactions are NOT posted based on the System Settings Daily Summary rules. ie: the posting will always occur "Today" not the following day. You should be mindful the Accounting Software you are integrating to is not making use of other software or features that may require all invoice line details to show. eg: third party pay now software used for end customers to download their invoice from the Accounting Software to pay. This enhancement is only available to sites integrating with MYOB AR Live / QBO, Xero and Sage Evolution **Implementation Of This Enhancement:** Both Update 238 and Update 242 include this logic if this new feature is to be turned on. Ensure both Constants are set to either **True (**to turn this feature on) or **False** (to turn it off). Ostendo will process these transactions in their detailed form if no Constant exists.

Development- X User Defined Constants

User Defined Constants published by Development-X allow you to control the default behaviour of Ostendo. Development-X makes use of these to increase the functionality of Ostendo. These can act like switches in Ostendo to control how Ostendo can process or display information. New constants are added over time as Development-X continues to make use of this approach to provide further flexibility and functionality to Ostendo without the need to publish official Updates. By using this approach, we can control behaviour in one database different to that in another. **Please note that the Constant Names are case sensitive.**

Module	Description	Constant Name	Constant Type	What This Constant Controls
Financial	MYOBDELETIONHORIZON	**MYOBDELETIONHORIZON (MYOB AR Live Only)**	**Integer**	The default deletion horizon is 45 days. This means the script will check payments up to 45 days old to see if any are deleted in MYOB and update Ostendo accordingly. Adding this UDC and specifying a value other than 45 will modify the deletion horizon used.
Financial	MYOBDELETIONLIMIT	**MYOBDELETIONLIMIT (MYOB AR Live Only)**	**Integer**	To prevent accidental mass deletion of payments from Ostendo, the script has a default limit of 10 – i.e. a maximum of 10 payments can be deleted in any batch. This could be a possible reason why at times, payments deleted in MYOB are not reflected in Ostendo. This UDC allows you to change the limit if so required.
Financial	MYOBCOMPANYCATEGORY	**MYOBCOMPANYCATEGORY (MYOB AR Live Only)**	**Text**	One MYOB Category ID can be assigned to each Ostendo database. This is useful in environments where a group of companies have multiple sites and each site has its own Ostendo database. Assigning a different Category ID to each site in MYOB gives the ability to analyse or group financial data by site. To set this up, you need to first define the Category IDs in MYOB Categories List. Then for each Ostendo database you add a UserDefinedConstant called "MYOBCOMPANYCATEGORY" and assign the appropriate Category ID as its value. All journals and invoices posted to AR Live will carry the assigned Category ID for that database
Financial	UPDATE_MYOB_SUPPLIER_PAYMENT DETAILS	**UPDATE_MYOB_SUPPLIER_PAYMENTDETAILS (MYOB AR Live Only)**	**Integer**	Use this UDC (Ticked = Yes) to indicate whether you wish to update Supplier's payment details (Bank AccountName and Number) when Supplier records are updated. The default is No
Financial	MYOB_PRELOAD_DATA	**MYOB_PRELOAD_DATA (MYOB AR Live Only)**	**Yes/No**	Preloading data (Customer, supplier, journals, invoices, payments) into memory, in theory, should improve performance especially for large MYOB files. However, in recent times, it has been observed that MYOB API does not handle large files very well – resulting in lots of Response Code 500 errors. So preloading data has become counter-productive. So the default now is NOT to preload data when posting a batch of transactions. Use this UDC (Ticked = Yes) to initiate preloading of data only if it does not result in lots of Response Code 500 errors.
Financial	MYOBTIMEOUT_REPOST	**MYOBTIMEOUT_REPOST (MYOB AR Live Only)**	**Yes/No**	There have been various reports of duplicate transactions in MYOB, and the source of such occurrences seems to be gateway timeouts (Response Codes 500, 504). With the recent reports of frequent occurrence of gateway timeouts, the script will now NOT attempt to repost transactions. This default option is to prevent high instances of duplicate transactions posted to MYOB . This may result in more failed transactions which can be reposted again later. This may be a better option than having duplicate transactions in MYOB. However, if duplicate transactions have never been an issue, but timeout errors have resulted in too many failed batches, then setting this flag "MYOBTIMEOUT_REPOST" to YES may help reduce the number of failed batches.
Financial	MYOB EXTERNAL PAYMENT METHOD	**MYOB EXTERNAL PAYMENT METHOD (MYOB AR Live Only)**	**Text**	When External Payments are created in Ostendo, the default Payment Method used is as specified in the CustomerMaster record (or in the Sales Rules). The Payment Method specified with this constant must be already defined in the Payment Methods screen. Specify the Payment Method to use NB: Case Sensitive

Development- X User Defined Constants

User Defined Constants published by Development-X allow you to control the default behaviour of Ostendo. Development-X makes use of these to increase the functionality of Ostendo. These can act like switches in Ostendo to control how Ostendo can process or display information. New constants are added over time as Development-X continues to make use of this approach to provide further flexibility and functionality to Ostendo without the need to publish official Updates. By using this approach, we can control behaviour in one database different to that in another. **Please note that the Constant Names are case sensitive.**

Module	Description	Constant Name	Constant Type	What This Constant Controls
Financial	QBO-TAXEXCLUDED	**QBO-TAXEXCLUDED (QBO Only)**	**Yes/No**	This Constant controls whether QBO will calculate the Tax Amount itself, rather than Ostendo forcing the Tax Amount when the QBO transaction is posted. The Default method is Tax Inclusive if this Constant is note defined or is Defined as 'No'
Financial	MWTRANSDATEHORIZON	**MWTRANSDATEHORIZON (MYOB AR Live Only)**	**Integer**	Transactions which were backdated several years back have been found to cause batches to run for excessive amounts of time as lots more records needs to be retrieved from Mworks. To avoid such issues and to improve overall performance, the following changes have been made: Transactions dated before CutOver Date will NOT be posted across to MWorks. Such transactions will automatically be set to Valid in JOURNALHEADER/JOURNALINVHEADER tables. The TransactionMessage field will contain the message "Not Posted – Transaction before CutOverDate). Transactions dated before CutOver Date will NOT be posted across to MWorks. Such transactions will automatically be set to Valid in JOURNALHEADER/JOURNALINVHEADER tables. The TransactionMessage field will contain the message "Not Posted – Transaction before CutOverDate). This Constant can be defined in Ostendo (if required). This is an integer value and indicates the time horizon in days. (e.g. 365 = 1 year horizon) The resulting "horizon date" cannot be older than the CutOver Date. If it is, then it will be reset to the CutOver Date. Any transaction that is dated after the CutOver Date, but before the "horizon date" – will be set as Invalid with the TransactionMessage: "Not Posted – Transaction before TransHorizonDate".
Financial	ONLY POST FINANCIAL INVOICES AND PAYMENTS	**ONLY POST FINANCIAL INVOICES AND PAYMENTS**	**Yes/No**	If this Constant exists with a Constant Value of 'True', Simplified Posting will be used, otherwise in the absence of this Constant or if the Constant Value is False, then full Standard posting will occur. Refer to Simplified Financial Posting Option http://ostendofreeway.info/HelpDocs/Ostendo/Accounting/SimplifiedFinancialPostingOption.html
PDF Export	Print PDF Optimized	**PDF Print Optimized**	**Yes/No**	This constant allows you to control the 'Print Optimized' setting when exported a report to PDF. If set to Print Optimized, the PDF graphics will output a higher quality graphic. NB: This setting will cause the resulting PDF to be increased in size, therefore it is recommended that you test before implementing this option. ***Yes (True)*** Ostendo will automatically select the Print Optimized option on the 'Export to PDF' prompt screen ***No (False)*** Ostendo will leave this parameter blank on the 'Export to PDF' prompt screen. (NB: This is default behavior even if this constant is not defined)

Development- X User Defined Constants

User Defined Constants published by Development-X allow you to control the default behaviour of Ostendo. Development-X makes use of these to increase the functionality of Ostendo. These can act like switches in Ostendo to control how Ostendo can process or display information. New constants are added over time as Development-X continues to make use of this approach to provide further flexibility and functionality to Ostendo without the need to publish official Updates. By using this approach, we can control behaviour in one database different to that in another. **Please note that the Constant Names are case sensitive.**

Module	Description	Constant Name	Constant Type	What This Constant Controls
Pivot Output	PivotExcelHeadings	PivotExcelHeadings	**Yes/No**	This User Defined Constant PivotExcelHeadings (Yes/No) to be used to control whether Row Field names are output to Excel from a Pivot Table. Some sites wish to show these headings in Excel however by allowing this, the output software component used by Ostendo automatically inserts some rows and column which may cause formatting issues. ***Yes (True)*** Ostendo will output all row headings to Excel ***No (False)*** Ostendo will not output row heading to Excel. (NB: This is default behavior even if this constant is not defined)