

CONFIGURATION_enumeration

Basic STOCK configuration for data import and computation. Enumertion of possible keys and values of sisi-CONFIGURATION table.

Name	DefaultValue	DataType	PossibleValues	Description
treeview_product_structure_category_type_count	NULL	int		Number of product structure category levels (category_type.is_product_structure = 1).
treeview_warehouse_structure_category_type_count		int		Number of product structure category levels (category_type.is_warehouse_structure = 1). The warehouse level (is_warehouse_level =1) is omitted.
recompute_set_history_length_in_days	60	int		If we want to recompute the minimal quantity/packaging, we do it based on purchases. This parameter determines how many days back in the history will the recomputation take into consideration.
recompute_set_percentage_of_lower_orders	70	int		If the minimal quantity/packaging is not submitted by the customer, the system can compute it automatically based on past transactions. If we want to recompute the minimal quantity/packaging, the recomputed value is set only if X% of purchases lower than the original value.
redistribution_minimal_period_source	DM	varchar	PW/DM	PW = STOCK GUI settings, DM = data from the SIDI input
ordering_minimal_period_source	NULL	varchar	PW/DM	PW = STOCK GUI settings, DM = data from the SIDI input
ordering_order_days_source	DM	varchar	PW/DM	PW = Planning Wizard GUI settings, DM = data from the SIDI input

redistribution_order_days_source	DM	varchar	PW/DM	PW = STOCK GUI settings, DM = data from the SIDI input
production_order_days_source	DM	varchar	PW/DM	PW = STOCK GUI settings, DM = data from the SIDI input
ordering_number_of_days_to_solve	720	int		How many days in advanced are orders to be suggested.
ordering_amount_max_days_to_round	0	int		How long the quantity of delivered stock should last at maximum, rounded to days. Should the level be exceeded, unrounded quantity is ordered. The parameter is used for rentability evaluation. If set to 0, rentability is not evaluated.
cancellation_cleanup_enabled	1	int	0/1	1 = cleaning cancellations, i.e., if there is a transaction of an opposite polarity than expected, it is a cancellation. Cancellation is subtracted from the history of transactions of a similar type. For example, for sales, the quantity should have 'minus' sign (goods leave the warehouse). If the quantity is positive, this transaction is a cancellation.
cancellation_cleanup_days_to_history	365	int		How many days after a transaction that should be cancelled must a cancellation transaction appear. In other words, transaction is treated as cancelled only if it is at most X days older than the cancellation transaction date.
cancellation_cleanup_transaction_type_id	1,7	varchar		Types of transactions for which automated cancellation will be performed, divided by comma; see cancellation_cleanup_enabled.

cancellation_cleanup_order_string	'date_of_transaction'	varchar		A auxiliary parameter of ORDER function, to be used for identification what kind of cancellations should be performed primarily. It can be anything suitable after ORDER BY above the table TRANSACTION.
cleanup_pw	1	int	0/1	1= all STOCK tables are cleared after UDM import (used during the implementation phase of a project), 0=computation to increments (used later in production phase)
cleanup_pwdata	0	int	0/1	If 1, then in the instance test *_pwdata, from UDM and STOCK all plids that are not set as warehouse_product.import_pwdata and aggregated tables are erased
transaction_mode	REPLACE	varchar	REPLACE/INCREMENT	1) transaction_mode = REPLACE and transaction_history_reimport_days = 0: all transactions in STOCK are deleted and inserted from the table TRANSACTION; 2) transaction_mode = REPLACE and transaction_history_reimport_days = X, X>0: the transactions for the last X days are deleted from STOCK and inserted from the table TRANSACTION; 3) transaction_mode = INCREMENT and transaction_history_reimport_days = 0: nothing is deleted from STOCK, all transactions are inserted from TRANSACTION to STOCK (and replace the original data in STOCK, if present) 4) transaction_mode = INCREMENT and transaction_history_reimport_days = X, X>0: nothing is deleted from STOCK, transactions from the last X days are inserted into STOCK (and replace the data original in STOCK, if present).
transaction_history_reimport_days	NULL	int	0-365	see description of transaction_mode

warehouse_category_type_id	2	int		ID of category_type with is_warehouse_level = 1; cf. a category that represents the warehouse level.
pwdate		date		If config file has a STOCK date set to 'universal' or 'import', then this date is used.
max_warehouse_product_count	500000	int		The highest possible number of items in the table warehouse_category. By default, it is 500,000, if you need more than that, set the default number as required.
leadtime_calculation_method	HEADER	varchar	HEADER/ROW	HEADER - if more than one product of the same type is on the order, only the first one counts. ROW - if there is more deliveries, each gets counted (hence, it increases the LT STD)
treeview_category_hiding_mode	HIDE	varchar	HIDE/SKIP	HIDE - it respects the category.hidden, products are displayed from the first real (not hidden) category. SKIP - it doesn't display a category if there is nothing under it.
warehouse_product_disabled_mode	DISABLE_CHILDS	varchar	DISABLE_CHILDS/THROW_EXCEPTION	DISABLE_CHILDS - if disabled is set on SKU in parent warehouse automatically set disabled on all SKU's on child warehouses, THROW_EXCEPTION - kill pw batch if this is detected
special_preorder_minimal_quantity_mode	ALWAYS	varchar	ALWAYS/ZERO	When to apply the rule of minimal quantity in pre-order: ALWAYS - pre-order for the promotion action won't be lower than the value in special_preorder_minimal_quantity. ZERO - it gets applied only if we wouldn't order anything otherwise (i.e., either the stock is sufficient, or this is a promo with forecast 0).
special_preorder_use_expiration	1	int	0/1	If set to 0, special pre-order for a promo won't respect expiration date.

import_holiday	NONE	varchar	NONE/CZ/SK	Set holidays as days without sale. The parameter refers to either NONE or country of operation (CZ, SK, RO, RS, IT, PL).
repair_price_move_type	MISSING	varchar	MISSING/NONE	The missing prices from the transactions of the same item are added. Options: MISSING: adds prices of transactions; NONE: missing prices from other transactions are not added.
repair_price_product_locator_type	MISSING	varchar	MISSING/ALL/NONE	Recomputation of prices from transactions. Options: MISSING: only null prices are recomputed; ALL: all prices are recomputed; NONE: no price gets recomputed.
repair_price_product_locator_aggregation	PLID	varchar	PLID/PRODUCT	Settings for price of items: which transactions are going to be used to supply the missing prices. Options: PLID: only transaction of a given item in a particular warehouse, PRODUCT: if there are no transactions for a given item in a particular warehouse, then transactions in other warehouses are used.
repair_price_product_locator_interval_months	3,12	varchar		Numbers separated by comas - time periods in month, which serve as a base to search for suitable transactions to supply the item price; if none transaction is to be found in the first time period (by default 3 months), the transactions are searched for in the period indicated by the second number. The number of time periods is not restricted.
cancellation_cleanup_incremental_enabled		int	0/1	1 = enables incremental cancellation, i.e. the table in the MOVE table will be restored and the cancellations will be reused for x days back (including all moves and cancellations related to these cancellations), where x is the value in cancellation_cleanup_days_to_history

foreign_key_error_type	WARNING	varchar	CRITICAL/WARNING	If FOREIGN KEY type error occurs and this setting is marked as CRITICAL, the computation is terminated and an error appears. WARNING won't stop the computation, only a warning message appears.
future_delivery_history_mode	REPLACE	varchar	REPLACE/INCREMENT	Similar behaviour as transaction_mode configuration, only for future_delivery_history => real_order_history table import.
future_delivery_history_reimport_days	NULL	int	0-365	Similar behaviour as transaction_history_reimport_days configuration, only for future_delivery_history => real_order_history table import.
id_in_pw_consistency	WARNING	varchar	WARNING\CRITICAL	One of WARNING, CRITICAL. Finding inconsistent IDs either causes a warning to be trigger or the computation to be killed.
product_locator_predecessor_max_depth	10	int	<1, ?)	Maximum allowed depth of product locator predecessors.
recompute_set_group_by	10	varchar	PRODUCT\PLID	(PRODUCT / PLID - 1) PRODUCT for a supplier, a set is computed on "product-supplier" combination and for distribution on "product" 2) PLID for a supplier, a set is computed for "product-locator-supplier" and for distribution on "product-locator"