Catalog Load Scenarios

Please refer for CM event details: CatalogMaster events Use cases

	PCM	Catalog Master
Full-Load ingestion		When Full Load is initiated, then • add an entry to DynamoDB (fullload=true) to identify full load running (all incremental jobs should check this flag (fullload=false) before executing) • Empty the SQS queue having the CM events
Category CSV contains all category attributes and relationships	N/A	Category handling • Generate CSV file for category data (L1-L3) for each country (US, CA) using the CatalogMaster APIs ◦ contains all category attributes ◦ contains all category relationships • Ingest the file into Commercetools and create categories and its hierarchies for both US and CA individually
SKU CSV		Sku handling • Generate CSV file with all skus, its linked product and its attributes, linked categories in flattened structure, individually for each country **Note**: Bring only the required attributes consumed from CM into the CSV. • Ingest sku CSVs into commercetools using “product variant patch” ◦ For each sku, create/update a product and sku 1:1 relationship with all sku attirbutes ◦ for linked product, update the CT product with product attributes like productId, nonreturnable, unifiedId etc ◦ link the CT product with Categories as defined in CSV
Catalog hierarchy changes		
category changes (Clean up task for De-link process)		Identify the diff of all categories during the aggregation with old and new snapshot. Find the ones that have been dropped from the new snapshot Update its parent to “Unlinked categories” in CT

PCM Events (SKU)	• Get the PCM event, and identify the CT product using the <skuId> ◦ productId=us_<skuId>, ca_<skuId> ◦ get the products using the proudctId • For each country: ◦ if product not found ▪ create “product draft” request with all relevant sku and style attributes in that country ▪ the localized attributes should be added to the country based sku • eg: canadian french display name should be added to `ca_` sku ▪ Price handling • Add us pricelist to us sku and ca pricelist to ca sku • fetch pricing using CM api call ◦ else, if found ▪ Diff the attributes between relevant PCM attributes and CT attributes, if changes found ▪ create a “product variant patch” request with all the attributes required from PCM event • the localized attributes should be added to the country based sku ◦ eg: canadian french display name should be added to `ca_` sku • ~~SKU handling~~ ◦ ~~create product and sku 1:1 for each sku event~~ ◦ ~~Create 2 skus for US and CA, eg: us_<sku_id> and ca_<sku_id>~~ ◦ ~~the localized attributes should be added to the country based sku~~ ▪ ~~eg: canadian french display name should be added to ca_ sku~~ • ~~Price handling~~ ◦ ~~Add us pricelist to us sku and ca pricelist to ca sku~~ ◦ ~~fetch pricing using CM api call~~	N/A

Incremental Load		When Incremental Load is initiated, then • check the flag (fullload=false, full load not running) before executing. • Catalog10001 represents US catalog Id , catalog10002 represents CA catalog Id in any event coming from CM. • Use SQS batch from lambda to read msgs in batch. SQS batch with multi-clients or threads will provide good throughput as events will comes in burst during drop days. If a message fails within batch , retry queue should be used before putting event in DLQ. Note: • @Rohit Singh to confirm whether to use java sync or import APIs . • **From initial discussion it seems Import API is more suitable as different events can be out of sync** • Create one container based on resource and country, eg: ◦ us_category ◦ ca_category ◦ us_product etc • If java sync is to be used, in case there is any exception because of resource still not created in CT , we need to have a retry pattern after some delay. eg. category event being processed before a product event and category has reference to new product . • @Rohit Singh To verify if the import api action is patch or put? • Create following catalog structure using Terraform ◦ US Root category ▪ Non Navigable Products (usNonNavigableProducts) ▪ Men, *> these are already present* ▪ Women etc ◦ US Inactive category (us_inactive) ◦ CA Root category ▪ Non Navigable Products (caNonNavigableProducts) ◦ CA Inactive category (ca_inactive) • When change to any L1 category, then update in the full load aggregation and Terraform • Update current Full Load aggregation to include the above categories as well (usNonNavigableProducts, caNonNavigableProducts etc)
PCM Events	• SKU handling ◦ Add/update CT product and sku 1:1 for each sku event ◦ Add/update 2 skus and products for US and CA, eg: Sku key: `us_<sku_id>` and `ca_<sku_id>` Product key: `us_<sku_id>` and `ca_<sku_id>` ◦ the localized attributes should be added to the country based sku ▪ eg: canadian french display name should be added to `ca_` sku only ◦ • Price and all additional sku attribute handling from CM. (Need to revisit for future, keep as is) ◦ Add USD price to US variant and CAD price to CA variant ◦ fetch price by Catalogmaster api call ◦ Fetch additional missing PCM sku attributes to be set on CT product (displayable, hazmat, xborder, productId etc) from CM ▪ if multiple parentproducts, please use the first parentProduct ▪ if parentproducts=null, update the CT.productId=Null/empty	

CatalogMaster - Category Events CatalogMaster events Use cases	N/A	• identify any change to category attributes required for NGC ◦ Events contain old snapshot and new snapshot of the category ◦ Identify event country based on `site_id` • Update the category attribute changes in CT (unifiedId, displayName, description etc): Category • If event does not contain old snapshot , it means category is new . Create a new category in CT & set its parent categories based on the snapshot. • Add/update Category relationships (L1, L2, L3) ◦ in case ***category.parentCategory*** is empty/Null (category unlinked from all categories), then assign the Category to a placeholder category “InactiveCategories“ in CT ◦ In other cases , identify which categories need to be added or removed in parent categories relationship in CT based on snapshots. (category.childcategory changes should be ignored, as that will also appear in the child category event) • Add/update ***childProduct*** relationships (L1, L2, L3): ◦ Product removed in new snapshot ▪ get all the products in CT (using product projection search) whose product.productId=<removed productId> in CT ▪ If there are no products returned from CT, it means its a all together new product assigned to this category. Get all SKUs for this product from Catalog master API & follow the below. ▪ ~~*Execute product “remove-from-category” update action to remove the specific category from parent*~~ *Or* ▪ *Use the import api to create the product draft/patch payload to update the parent product (preferred)* ◦ Products added in new snapshot ▪ Get all SKUs for this product from Catalog master API & follow the below. ▪ ~~*Execute product “Add to Category” update action to add the specific category from parent*~~ *Or* ▪ *Use the import api to create the product draft/patch payload to update the parent product (preferred)* Note : • **Make a config in code , never delete root categories (US, CA) in CT.** • Do not read Category events in Batches to prevent timeouts. • Add scaling (lambda concurrency) for reading events from SQS

CatalogMaster - Product Events	N/A	• Identify any change to Product attributes required (product display name, FR display name , onsale, returnable, free shipping, childSku etc) for NGC ◦ events contain old snapshot and new snapshot of the product ◦ Identify event country based on `site_id` • New Product (no old snapshot) ◦ Create “product variant patch” request for all childsku in the product (same as below **Update logic**) • Modified product with new skus or removed skus ◦ For all the new skus, follow the below **Update logic** (create product variant patch request) ◦ For all the removed skus, follow the below **Update logic** (create product variant patch request and update all product attributes: ▪ productId=null/empty ▪ product unifiiedId, ▪ product ParentCategory etc • **Update logic “product variant patch” request** ◦ Update product.productId to CM productId ◦ Update product.categories in CT to CM product.parentCategories ◦ update all relevant product attributes like product display name , onsale, returnable, free shipping, pdpUrl, parentCategory etc. ◦ **Note:** ▪ parentCategory update would be a “**product patch**” request • As product patch request requires name (sku displayname), hence need to make a CM sku api call. ▪ This should automatically take care of new products as well ▪ If category does not exist in CT at this time, we need to check import API finally sets it correctly when category event gets ingested as part of category event process. ▪ upstream publishes a product without linking it to a category, but sku would be available, so product.parentCategory should be empty/NULL

CatalogMaster - Sku Events	N/A	• Compare all NGC specific attribute in old and new snapshot , if there is any change detected update (“product variant patch”) the Sku/product entity in CT. **Note**- **make attributes configurable. Diff is very important as lot of SKU updates come in night which have no change w.r.t NGC.** • Make a call in CT to get the sku details. If sku is missing, then create a product draft with all required details from CM event data. **Note** : this will take care of edge cases as NGC need to be in sync with merch system. Logging should happen to find out these instances. Also follow the below steps (even for above skus): • Identify event country based on `site_id` or SKU prefix (us_ or ca_). Site Id might be null/blank as well. • If SKU **does not have price** in payload , means don't update price attributes in CT. Only update other attributes (using patch call). • Always create a payload of CT (Create “product variant patch” request) with SKU and Product entity along with price & CM specific attributes. .Get latest data from Catalog master SKU event without looking up in CT. Attributes also include productId (from parentProducts property in SKU event). • **Ignore productId** update based on decision on 03/24/2022. This is not required. **NOTE** • **SKU will always be updated via catalog master event for CM specific attributes.** • Even if SKU does not belong to a catalog in source event (i.e site id is blank/null), SKU need to be processed as pricing data comes much before SKUs are attached to product catalog upstream.

Catalog Load pipeline