|  |
| --- |
| **eCommerce Storefront On Demandware Platform** |
| Technical Specifications Document (TSD)  Table of Contents  [1 Change History 8](#_Toc372735233)  [2 Overview 8](#_Toc372735234)  [2.1 External Documents 8](#_Toc372735235)  [2.1 Data Samples 8](#_Toc372735236)  [3 SiteGenesis 11](#_Toc372735237)  [3.1 Net New Requirements 11](#_Toc372735238)  [4 Disabled SiteGenesis Functionalities 12](#_Toc372735239)  [5 Architecture 12](#_Toc372735240)  [5.1 Data Migration 13](#_Toc372735241)  [5.1.1 Legacy Customer 13](#_Toc372735242)  [5.1.2 Order 13](#_Toc372735243)  [5.1.3 Payment History 13](#_Toc372735244)  [5.2 Regular Imports 14](#_Toc372735245)  [6 Data Mapping & Structure 14](#_Toc372735246)  [6.1 Shared Cart Architecture [Multi Band Site] 14](#_Toc372735247)  [6.1.1 Data Model 15](#_Toc372735248)  [6.1.2 Branding UI 16](#_Toc372735249)  [6.1.3 Implementation Notes 16](#_Toc372735250)  [7 Integrations Summary 16](#_Toc372735251)  [7.1 Integration Specifications – Cybersource 17](#_Toc372735252)  [7.1.1 Sites Integrated 17](#_Toc372735253)  [7.1.2 Use Cases 17](#_Toc372735254)  [7.1.3 Integration Flow Diagram 18](#_Toc372735255)  [7.1.4 Source Details 18](#_Toc372735256)  [7.1.5 Configuration 18](#_Toc372735257)  [7.1.6 Credit Cards Accepted 18](#_Toc372735258)  [7.1.7 End Point Credentials 18](#_Toc372735259)  [7.1.8 Known Issues 19](#_Toc372735260)  [7.1.9 Volume and Frequency 19](#_Toc372735261)  [7.1.10 Source and Target Data Formats 19](#_Toc372735262)  [7.1.11 Transport 19](#_Toc372735263)  [7.1.12 Timeout settings 19](#_Toc372735264)  [7.1.13 Storage and Retention Plan 19](#_Toc372735265)  [7.1.14 Availability and Fall Back Plans 19](#_Toc372735266)  [7.1.15 System Error Messages 19](#_Toc372735267)  [7.1.16 Document Reference 19](#_Toc372735268)  [7.1.17 Best Practices 19](#_Toc372735269)  [7.1.18 Developer Notes 19](#_Toc372735270)  [7.2 Taxation 19](#_Toc372735271)  [7.2.1 Tax rates Sample 19](#_Toc372735272)  [7.2.2 Sample DW Tax Table 20](#_Toc372735273)  [7.3 Integration Specs - OMS 21](#_Toc372735274)  [7.3.1 DW Order History 21](#_Toc372735275)  [7.3.2 Real Time Inventory Check 21](#_Toc372735276)  [7.3.3 Integration Specs - Order Export 21](#_Toc372735277)  [7.4 Integration Specs - Email Sign-Up 22](#_Toc372735278)  [7.4.1 Functional Overview 22](#_Toc372735279)  [7.4.2 Limitation and Constraints 22](#_Toc372735280)  [7.4.3 Sites Integrated 22](#_Toc372735281)  [7.4.4 Use Cases 22](#_Toc372735282)  [7.4.5 Integration Flow Diagram 23](#_Toc372735283)  [7.4.6 Source Details 23](#_Toc372735284)  [7.4.7 Configuration 23](#_Toc372735285)  [7.4.8 End Point Credentials 23](#_Toc372735286)  [7.4.9 Known Issues 23](#_Toc372735287)  [7.4.10 Volume and Frequency 23](#_Toc372735288)  [7.4.11 Source and Target Data Formats 23](#_Toc372735289)  [7.4.12 Transport 23](#_Toc372735290)  [7.4.13 Timeout settings 23](#_Toc372735291)  [7.4.14 Storage and Retention Plan 23](#_Toc372735292)  [7.4.15 Availability and Fall Back Plans 23](#_Toc372735293)  [7.4.16 Document Reference 23](#_Toc372735294)  [7.4.17 System Error Messages 23](#_Toc372735295)  [7.5 Integration Specs - Scene 7 23](#_Toc372735296)  [7.5.1 Image Assignment (Catalog) 24](#_Toc372735297)  [7.5.2 Creating Image URL 26](#_Toc372735298)  [7.5.3 Zoom 26](#_Toc372735299)  [7.5.4 Flow Diagram 26](#_Toc372735300)  [7.5.5 Configuration data 27](#_Toc372735301)  [7.5.6 Image Flow Diagram 27](#_Toc372735302)  [7.6 Integration Specs - Media Pixels 28](#_Toc372735303)  [7.7 Integration Specs - Omniture 28](#_Toc372735304)  [7.7.1 Omniture Script Configuration 29](#_Toc372735305)  [7.8 Integration Specs - Google Analytics 29](#_Toc372735306)  [7.8.1 Implementation details 29](#_Toc372735307)  [7.8.2 Page Specific code 30](#_Toc372735308)  [8 JavaScript On/Off Behavior 32](#_Toc372735309)  [9 Browser Support Matrix 32](#_Toc372735310)  [10 Mobile 32](#_Toc372735311)  [11 Accessibility 33](#_Toc372735312)  [12 Scheduled Jobs 33](#_Toc372735313)  [13 Manual Jobs 34](#_Toc372735314)  [14 Data Model Extensions 35](#_Toc372735315)  [14.1 System Objects 35](#_Toc372735316)  [14.1.1 Product 35](#_Toc372735317)  [14.1.2 Category 35](#_Toc372735318)  [14.1.3 Order 35](#_Toc372735319)  [14.1.4 Global and Site Preferences 35](#_Toc372735320)  [14.2 Custom Objects 35](#_Toc372735321)  [14.2.1 Feed Job Configuration 35](#_Toc372735322)  [14.2.2 Cartridge Structure 37](#_Toc372735323)  [15 Environment and Build Process 38](#_Toc372735324)  [15.1 Build Cartridge (Optional) 39](#_Toc372735325)  [15.2 Build Frequency 39](#_Toc372735326)  [15.3 Central Build Server (Optional) 39](#_Toc372735327)  [15.4 Build Responsibilities 39](#_Toc372735328)  [16 Aliases 39](#_Toc372735329)  [17 Multiple Storefronts 40](#_Toc372735330)  [17.1 Store BM front URLs 40](#_Toc372735331)  [17.2 Cartridge Structure 40](#_Toc372735332)  [17.3 Cartridge Definitions 40](#_Toc372735333)  [17.4 Style Sheet 40](#_Toc372735334)  [17.5 Style Guide 40](#_Toc372735335)  [18 Development UX Studio / Sandbox - Special Settings 41](#_Toc372735336)  [18.1 Cartridge Structure 41](#_Toc372735337)  [18.2 Coding Standards 41](#_Toc372735338)  [18.3 Naming Conventions 41](#_Toc372735339)  [19 Business Manager 41](#_Toc372735340)  [19.1 Redirects 41](#_Toc372735341)  [19.2 Site redirect URL 43](#_Toc372735342)  [19.3 Static mapping 43](#_Toc372735343)  [19.4 Search Engine Optimization 43](#_Toc372735344)  [19.5 Aliases 43](#_Toc372735345)  [20 Source Control 43](#_Toc372735346)  [20.1 SVN Link and Accounts / Team Process 43](#_Toc372735347)  [20.2 Backup Schedule 43](#_Toc372735348)  [20.3 Source/ Metadata Update/ Recovery Procedure 43](#_Toc372735349)  [21 Monitoring 43](#_Toc372735350)  [22 Security 43](#_Toc372735351)  [22.1 Provisioning of 2-level Authentication 43](#_Toc372735352)  [22.2 SSL Certificates for Launch and Development needs 43](#_Toc372735353)  [23 POD Control Center 44](#_Toc372735354)  [23.1 Accounts / Access List 44](#_Toc372735355)  [23.2 POD Details 44](#_Toc372735356)  [23.3 Outgoing address of POD 44](#_Toc372735357)  [24 Single Sign-On 44](#_Toc372735358)  [25 Global Error Codes and Messages 44](#_Toc372735359)  [26 Internationalization / Localization 44](#_Toc372735360)  [27 Site Performance 44](#_Toc372735361)  [27.1 KPIs 44](#_Toc372735362)  [27.2 Conversions 44](#_Toc372735363)  [27.3 Data Volumes 44](#_Toc372735364)  [27.4 API Quotas 44](#_Toc372735365)  [27.5 Platform Quotas 44](#_Toc372735366)  [27.6 Canonicalization 44](#_Toc372735367)  [27.7 Data Replication 45](#_Toc372735368)  [27.8 Resource Usage Estimates 45](#_Toc372735369)  [27.8.1 Cyber Monday Estimates 45](#_Toc372735370)  [27.8.2 Expected Averages 45](#_Toc372735371)  [27.8.3 Migration Estimates 46](#_Toc372735372)  [27.9 Email Accounts 46](#_Toc372735373)  [27.10 Transaction Emails 46](#_Toc372735374)  [27.11 Key Performance Indicators 46](#_Toc372735375)  [27.12 Static File Optimization Plan 47](#_Toc372735376)  [27.13 Data Retention 47](#_Toc372735377)  [27.14 Caching Strategy 47](#_Toc372735378)  [27.15 Cartridge Structure 47](#_Toc372735379)  [27.16 Site Cartridge Paths 48](#_Toc372735380)  [27.17 Coding Standards 49](#_Toc372735381)  Revision History   | Date | Description | Author | | --- | --- | --- | | 09/03/2018 | Initial Draft | Kalyanachakravarthy Etthamukala | |  |  |  |  |  | | --- | |  |  Change History  |  |  |  |  | | --- | --- | --- | --- | | **Version** | **Date** | **Prepared By** | **Comments** | | 1.0 | 09/03/2018 | Kalyanachakravarthy Etthamukala | Initial Draft  Added new requirements as per Training FSD under section 3.1 Net New Requirements.  Added new column implementation Details for section 3.1. | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |  Overview This document describes the eCommerce storefronts implementation details of the features reused from SiteGenesis, features that need to be built that are not available on SiteGenesis, integrations with 3rd party solution providers and backend system(s). External Documents  * Data mapping * FSD * SiteGenesis-wireframe.pdf * Data Mapping Spreadsheet  Data Samples <Client Name> will be provided with an initial sample file will be provided in the following areas. Demandware will expect that a complete file for each of these areas will be returned prior to the start of the project build phase.   |  | | --- | | **Function** | | Data Mapping Spreadsheet | | Catalog File | | Inventory File | | Pricebook File | | Order History Data File Sample | | Legacy Customer Data File Provided in DW Format | | Store Locator Showroom/Retailer File |   **Disclaimers**   |  |  | | --- | --- | | Section | Redefined requirement | |  |  | |  |  | |  |  | |  |  |   **Questions / Issues List**   | Type | Description | Owner | Date | | --- | --- | --- | --- | |  |  |  |  | |  |  |  |  | |  |  |  |  |   **Glossary of terms**   |  |  |  | | --- | --- | --- | |  | Term | Description | |  | SiteGenesis | Out of the box Storefront implementation on Demandware platform | |  | SFI | Simple Feed Integration |  SiteGenesis SiteGenesis is an out of the box standard Storefront implementation that comes with the Demandware eCommerce platform. SiteGenesis features include, but are not limited to, the following:   * Global Header and Footer * Global Navigation (driven by Site Catalog) * Homepage * Category Landing page * Sub-Category landing page with product grid * Search results page with product grid * Refinements on Category landing pages or search results page * Product Details page * Mini-cart in global navigation * Shopping cart page * Shipping page (shipping address and shipping method) * Billing page (billing address and payment details) * Order Review page * Order Confirmation page * Order Status page * Registration * Login (login, checkout login, password retrieval) * My Account (Profile, saved address, saved payment details)   SiteGenesis also comes with out of the box import/ export simple feed integration (SFI) that can be generated/ consumed by 3rd party solutions providers following are the feeds that will be used for the current project   * Order Export * Catalog (categories and products) * Pricing * Inventory * Order Status  Net New Requirements Requirements described in this section are net new requirements beyond the HLD process.   |  |  |  |  | | --- | --- | --- | --- | | FSD Ref | Redefined requirement | Implementation Details | Estimation in Hrs | | 1.3. Global Page - Category drop down 🡪 Footnote-2 | Display Product image for second level subcategory in category menu Bar. | 1. There are 3 level categories in SG. 2. Find the isml file where Category menu Bar is displayed. 3. Get the first product of first 3rd level subcategory and display beside to 2nd level subcategory. | 4 | | 1.0. Global Page – Header 🡪 Footnote - 12 | Add **Capgemini > Home** as a common breadcrumb in navigation | 1. Goto Administration > Site Development > System Object Definitions > SitePreferences 2. Goto Attribute Definitions tab then create an attribute named ‘sg\_breadcrumb’ and data type as ‘Set of String’. 3. Goto Attribute Groups tab then create an New Attribute Group and assign the attribute to this group. 4. Get this attribute and display it on store front breadcrumb. | 2 | | 1.10. Product Detail Page –  Footnote - 14 | Add the product to wishlist along with the actual quantity entered by the user | 1. Goto Wishlist pipeline and find the pipelet node where the product is adding to wishlist 2. Entered quantity will be available in CurrentHttpParameterMap. 3. Then pass the quantity to corresponding pipelet node. | 4 | | 1.10. Product Detail Page –  Footnote – 13,24,25 | Add to bag button ALT text | Create a div with text “Select size and color” next to the button. Find in app.js where the attribute disabled is getting removed when the size and color is selected there make your div visible |  | | 1.10. Product Detail Page –  Footnote – 26 | List link | 1. Identify the ProductDetails isml where next & Prev links are displaying 2. Check for search query string or category id in CurrentHttpParameterMap. 3. Call ‘Search-Show’ pipelet by passing the search query string to ‘q’ parameter/ category id to ‘cgid’ parameter. | 6 | | 1.2. Global Page – Footer -> Footnote - 14 | Email Registration - Submit the email address to receive the updates and offer on the site | 1. Create a new pipeline by adding the script node to achive this functionality. 2. On Submit the email address, create a CustomObject by using CustomObjectMgr for the attribute “EmailSubscription” in script node/file. 3. You can verify the submitted email address in BusinessManager.   Site -> Custom Objects -> Custom Object Editor | 6 | | 1.29. Payment Details – Add / Edit  Footnote – 6,8 | Edit or Add Billing address to MyAccount | **Add Billing address:**   1. create a custom attribute in CustomerPaymentInstrument as addressID 2. Create a custom attribute named ‘billingAddress’ as Boolean in ‘**CustomerAddress** ‘ of BM with default value false. 3. Add billingFields to paymentinstruments form 4. Alter pipeline to create Address after creating paymentinstrument object 5. Update billing address fields to Address object. 6. Update address id to the custom attribute addressID of PaymentInstrument and assign value as true to the Boolean attribute ‘billingAddress’ of CustomerAddress..   **Edit PaymentInstrument:**   1. Update the form with PaymentInstrument and associated Address. 2. Update changed fields to the Actual System Objects. | **16** | | 1.25. Address Book | Remove Link | Display the Remove link only if the value of ‘billingAddress’ is false | 2 | | 1.28 Payment Details  Footnote – 7 | Display Saved Billing Address | Displaying PaymentInfo section.(display billingaddress):   1. get addressID from CustomerPaymentInstrument 2. get AddressBook from CurrentCustomer 3. get Address by addressID from addressBook 4. Use Address object to display Billing Address. | 8 | | 1.14.Personalization Overlay  Footnote – 7 | Personalization overlay with Message field. | 1. Create a custom attribute 'personalize'  at product level as a boolean to decide wheather the product is personalize or not. 2. Create a custom attribute 'personalizeMessage' as a string at productlineitem level to save the message 3. create a personalize template(isml) to display on Dialogue.  * Use the SiteGenesis Dialogue to display personalize template content.  1. write hidden input in addtocart form to pass personalize message. 2. If click on add to bag button on Dialogue, update personalize message to hidden input in pdp, trigger add to cart button and close dialogue. | 12 | | 1.32 Mini Cart  Footnote – 9 | Should display Personalized message | Get the personalizeMessage attribute from productlineitem and Display 'personalizeMessage' in mini and card pages | 3 | | 1.7 Sub Category Landing Page  Footnote – 18 | Tag the product as ‘New’ | Create a custom attribute with string type at product level and display on product tile on PLP pages | 6 | | Sub Category Landing Page  Footnote – 11 | Product Image 2 | 1. configure 2 images for the product 2. show first image on tile on the page load and hide second image. 3. on hover hide first image, display second image and mouse out do vice versa using JQuery. | 8 | | 1.4 Home Page -  Testimonial  Footnote - 6 | Form Field | 1. Create a Testimonial template/isml to display the form. 2. Create Testimonial custom object to store the message. 3. Get 5 Recent messages form the testimonial custom object and display. | 8 | | 1.7 Sub Category Landing Page  Footnote - 21 | Grid View | Create a hyperlink with text “grid view” on click find selector #search-result-items .grid-tile and find selector .product-swatches and hide it. |  | | 1.31. Order History – Guest User  Footnote - 7 | Phone Number | 1. Identify the isml page where the new field is needs to be added. 2. Add the ‘Phone Number’ field as shown in FSD. 3. Identify the pipeline and add the phone number attribute ID to ‘SearchSystemObject’ pipelet node. | 6 | | 1.33. Shopping Bag  Footnote – 17,18 | Quantity Update | 1. Add ‘updateCart’ action in each lineitem level of cart page/isml. 2. Need to Edit the ‘updateCart’ Connector of ‘Cart-Show’ pipeline. 3. Add a Script Pipelet node ‘RemoveZeroQuantityLineItems.ds’ to updateCart’ Connector to remove the zero quantity product. 4. Then add ‘UpdateObjectWithForm’ pipelet node (From: TriggeredAction.parent, Object: TriggeredAction.object) to update the product line item quantity. 5. Then Add ‘Cart-Calculate’ call node to recalculate the cart. | 8 | | 1.7. Sub Category Landing Page | Add a new refinement | 1. Create a custom attribute ‘label’ at product level and add some values like SALE, NEW,etc.. 2. Go to BM and create a AttributeRefinement for attribute ‘label’ at Catalog level. 3. Then reindex your products. | 4 |  Disabled SiteGenesis Functionalities Following list of features and functionalities provided by SiteGenesis will be disabled / hidden for <Client Name> sites.   * Gift Certificates (Sample) * Gift Registry (Sample)  Architecture The Demandware system has built-in protections to optimize the performance and security of inbound data. The following diagram shows the logical distinctions between the production and staging environments. Performance and security concerns are met by loading most data through the staging environment and then using a daily replication process to move it to production. The one caveat is inventory which we typically reload on a very tight interval to adjust on an almost real-time basis.  <Client Name> storefronts will be integrating with various 3rd party solution providers. Following diagram gives an illustration of the <Client Name> technical architecture on Demandware Ecommerce platform –    *[Suggestions: Add any additional details of location of the integration end points, color code LINK cartridge integrations. Visio diagram file will be made available on exchange. ]* Data Migration Please provide a cut-over plan for data migration prior to go-live. This should include tasks with responsibilities assigned. Legacy CustomerOrderPayment History *[Note: All details with sample must be provided to Demandware prior to the start of our build process.]* Regular Imports SiteGenesis will leverage the import/ export simple feed integration (SFI). Following are the feeds that will be used for the <Site Name>:   * Order Export * Catalog (categories and products) * Pricing * Inventory  Data Mapping & Structure Following diagram illustrates <Client>’s catalog structure.  C:\Ravi\SRAs\Templates\Catalog Structure V.01.jpg Shared Cart Architecture [Multi Band Site] <Client Name> storefronts need to have shared cart architecture in place for future use. In this release each site will have only 1 brand.  Shared cart architecture requires the following components. Data ModelStorefront Catalog To implement shared cart, each site must have only one storefront catalog i.e. each brand is a top level category in the storefront catalog. And then brand level navigation lives underneath these categories. The top (brand) level categories define the site context.  Example  root  Brand1  top navigation category 1  top navigation category 2  Brand 2  top navigation category 1  top navigation category 2 Content Assets Taxonomy Just like catalog structure, all content assets on the site need to follow similar folder structure as the storefront catalog. i.e. each content asset lives under its brand folder.  Example  Root-content-folder  Brand 1  Content-assets  Brand 2  Content-assets Branding UI Each brand/country would define its own style sheet, which lives in the brand/country specific cartridge. e.g. clientbrand1.css would be defined in brand1’s cartridge. And similarly other brands would define their own style sheets. The important thing to realize is that the name of the CSS matches with the site context (brand category as defined in the storefront catalog).  What happens when user goes to common pages e.g. My Account, Checkout etc.? last visited Site Context? Implementation Notes [Add any implementation notes here] Integrations Summary We will integrate the <Client Name> sites with the following 3rd party systems. *[Note: Please summarize all LINK cartridges specific to each site. If implementation is over multiple phases, please specify them here.]*   |  |  |  |  | | --- | --- | --- | --- | | **Integration** | **US** | **UK** | **CA** | | **OMS**   * Real Time Inventory Check * DW Order History * Order Export | Y | Y | Y | | **Cybersource**   * Payment Authorization * Taxation * AVS | Y | Y | Y | | **Google Analytics** | Y | Y |  | | **Scene7**   * Product Images * Hero Image Zoom | Y | Y | Y | | **Media Pixels** | Y |  | Y | | **Add-This**   * Site Genesis built in feature would be utilized. | Y | Y | Y | | **Google MAP**   * Store details page would show Google map of the store address. | Y | Y | Y |  Integration Specifications – Cybersource <Client Name> storefronts integration will leverage the standard LINK Cartridge with the following features:   * Payment Authorization   + Full order amount authorization * AVS   + What is the behavior? What kind of messaging? Etc. * Tax   + After user successfully enters a valid shipping address, calculate taxes. If service is down or unavailable use DW tax table. What is the service timeout?  Sites IntegratedUse Cases  |  |  |  |  | | --- | --- | --- | --- | | **#** | **Conditions/Actions** | **Resulting Action** | **Notes** | | 1 | Customer places order; payment authorized and configured AVS tests passed. | Order is placed, as it would be with successful basic credit card authorization in SiteGenesis |  | | 2 | Customer places order; payment declined, or configured AVS tests failed, or other processing error such as timeout occurs. | Order is rejected, as it would be with failed basic credit card authorization in SiteGenesis |  |   Please refer to LINK cartridge documentation for implementation details. Integration Flow Diagram  Source DetailsConfiguration The following table shows the configuration data needed for Cybersource implementation.  [Note : Configure Cybersource merchant ID, security key, endpoints, and AVS rules (ignore/decline) as using the following site preferences:]   |  |  |  |  | | --- | --- | --- | --- | |  | Name of the attribute | System Object | Type | | 1 | CyberSourceMerchantName | Site Preference | String | | 2 | CyberSourceMerchantID | Site Preference | String | | 3 | CyberSourceMerchantPassword | Site Preference | String | | 3 | CyberSourceMerchantKey | Site Preference | String |  Credit Cards Accepted *[List of all the credit card types and BM configurations pertaining to respective Credit Cards per site]* End Point Credentials To log in to cybersource control center, use:  Url : https://xxxx.cybersource.com/xxxest/login/Logout.do  Merchant ID: xxxxx  Username: xxxxxx  PW: xxxxxx Known IssuesVolume and FrequencySource and Target Data FormatsTransportTimeout settingsStorage and Retention PlanAvailability and Fall Back Plans Retries should be enabled. The default values for retry\_start (30 seconds) and timeout (110 seconds) should be adequate. System Error MessagesDocument Reference Cybersource LINK integration cartridge – downloads and documentation:  <https://xchange.demandware.com/docs/DOC-10535> Best Practices *[Note: Some References to be included here for*   * *Dynamic Payment* * *Asynchronous Payment* * *PCI Best Practices* * *Integration Best Practices*   *]* Developer NotesTaxation <Client Name> will use Cybersource integration for real time tax calculations. DW tax tables will be setup as a fallback mechanism.  DW will define the following tax classes   * Standard – Class for standard tax calculations. * Exempt – Class for tax exemption i.e. 0 tax rate.  Tax rates Sample     'CA' => 12.15, // California      'CO' => 8.5,  // Colorado      'IA' => 7.0,   // Iowa     'NC' => 8.25,  // North Carolina     'PA' => 7.0,   // Pennsylvania Sample DW Tax Table <?xml version="1.0" encoding="UTF-8"?>  <tax xmlns="http://www.demandware.com/xml/impex/tax/2007-02-14">  <tax-classes>  <tax-class class-id="Standrd">  <display-name>Standard</display-name>  <description>Standard Tax Class </description>  </tax-class>  <tax-class class-id="Exempt">  <display-name>Exempt</display-name>  <description>Exempt Tax Class (0 rate) </description>  </tax-class>  </tax-classes>  <tax-jurisdictions>  <tax-jurisdiction jurisdiction-id="MA">  <display-name>Massachusetts</display-name>  <address-mappings>  <address-mapping key1="MA" key2="-"/>  </address-mappings>  </tax-jurisdiction>  <tax-jurisdiction jurisdiction-id="NH">  <display-name>New Hampshire</display-name>  <address-mappings>  <address-mapping key1="NH" key2="-"/>  </address-mappings>  </tax-jurisdiction>  <tax-jurisdiction jurisdiction-id="US" default="true">  <display-name>United States</display-name>  </tax-jurisdiction>  </tax-jurisdictions>  <tax-rates>  <tax-rate jurisdiction-id="MA" class-id="Standard">0.5</tax-rate>  <tax-rate jurisdiction-id="NH" class-id="Standard">0.0</tax-rate>  <tax-rate jurisdiction-id="US" class-id="Standard">0.0</tax-rate>  <tax-rate jurisdiction-id="MA" class-id="Exempt">0.0</tax-rate>  <tax-rate jurisdiction-id="NH" class-id="Exempt">0.0</tax-rate>  <tax-rate jurisdiction-id="US" class-id="Exempt">0.0</tax-rate>  </tax-rates>  </tax> Integration Specs - OMSDW Order History DW would only show order history for the orders placed in DW. A job will be created to fetch order status from client backend OMS. Standard DW order schema would be used for this purpose. Please refer to DW order schema documentation for more details. Real Time Inventory Check DW will integrate with <Client Name> OMS web service layer (interface TBD) to do real time inventory check. The check will be done at the following instances   * When user adds an item to the cart * When user performs checkout (final step?) DW to recommend.   DW will also update its inventory upon receiving real time inventory status.  If an item becomes out of stock, user will have to remove that item from the cart before placing the order. Integration Specs - Web Service TBD. Integration Specs - Order Export <Client Name> storefront implementation will leverage SFI cartridge for order export feature with the following customizations. Please check SFI documentation to configure the cartridge and the job.   * Order sequence number format, ‘CN’ or ‘CN’+18 chars, universally unique (organization scope) * No payment method info in the order export file except the authorization token. * Reason Code, site preference   + List of values, 2-4 AlphaNumeric  Integration Specs - Email Sign-Up DW will leverage <Cartridge Name> LINK cartridge to implement Email Sign-Up functionality. <Cartridge Name> will be the system of record for this feature. <LINK > integration will be used to capture the email preference. Refer to functional specification document for the use case scenarios for the email opt-ins. Following diagram illustrates the <LINK> integration.  ISD: <https://xchange.demandware.com/docs/DOC-8052>   Functional OverviewLimitation and ConstraintsSites IntegratedUse Cases  |  |  |  |  | | --- | --- | --- | --- | | **#** | **Conditions/Actions** | **Resulting Action** | **Notes** | | 1 |  |  |  | | 2 |  |  |  |   Please refer to LINK cartridge documentation for implementation details. Integration Flow DiagramSource DetailsConfigurationEnd Point CredentialsKnown IssuesVolume and FrequencySource and Target Data FormatsTransportTimeout settingsStorage and Retention PlanAvailability and Fall Back PlansDocument Reference Cybersource LINK integration cartridge – downloads and documentation: System Error MessagesIntegration Specs - Scene 7 Scene7 integration for <Client Name> will be done with data driven approach as outlined below.  Image path is defined in the master catalog as a proper scene7 resource along with catalog setup as outlined below.  Catalog Setup (Master catalog -> Image Settings)   * Image Location: External * http URL: scene7 provided base http URL. * https URL: scene7 provided base https URL. * View Types for DW (image presets for scene7)   + “large” – Large hero image   + “productalt” – Product thumbnails (small could also be used if same size)   + “small” – Cart page   + “medium” – Grid page   + “productnav” – Product detail page product navigation (previous/next, only required if UI dictates)   + “productrec” – Product recommendation   + “swatch” – Product detail page swatch   + “gridswatch” – Grid page swatch (if different than pdp)  Image Assignment (Catalog) Product images would be assigned to the products via the master catalog feed. Here is the sample xml. All images live under master product under images tag. View-type attribute ties what is displayed on storefront based on the FSD i.e. large is PDP hero image, small is alternate/thumbnail etc. variation-value attributes ties the images with the product variation.  Catalog needs to define every required image view-type for every product.  Path attribute has two parts, first part is the image name defined in scene7 and the second part within $$ is the image preset defined in scene7 as well.  <product product-id="10474">  <images>  <image-group view-type="large">  <image path="sweater-10474-model-nk11?$large$"/>  </image-group>  <image-group view-type="large" variation-value="CANDY CANE BEAR">  <image path="sweater-10474-bear-jk11?$large$"/>  </image-group>  <image-group view-type="large" variation-value="FAIRY">  <image path="sweater-10474-fairy-jk11?$large$"/>  </image-group>  <image-group view-type="large" variation-value="GINGERBREAD GIRLS">  <image path="sweater-10474-ginger-jk11?$large$"/>  </image-group>  <image-group view-type="large" variation-value="JOY SNOWMEN">  <image path="sweater-10474-snow-jk11?$large$"/>  </image-group>  <image-group view-type="large" variation-value="REINDEER AND TREE">  <image path="sweater-10474-deer-jk11?$large$"/>  </image-group>  <image-group view-type="large" variation-value="TRAIN">  <image path="sweater-10474-train-jk11?$large$"/>  </image-group>  <image-group view-type="medium">  <image path="sweater-10474-model-nk11?$medium$"/>  </image-group>  <image-group view-type="medium" variation-value="CANDY CANE BEAR">  <image path="sweater-10474-bear-jk11?$medium$"/>  </image-group>  <image-group view-type="medium" variation-value="FAIRY">  <image path="sweater-10474-fairy-jk11?$medium$"/>  </image-group>  <image-group view-type="medium" variation-value="GINGERBREAD GIRLS">  <image path="sweater-10474-ginger-jk11?$medium$"/>  </image-group>  <image-group view-type="medium" variation-value="JOY SNOWMEN">  <image path="sweater-10474-snow-jk11?$medium$"/>  </image-group>  <image-group view-type="medium" variation-value="REINDEER AND TREE">  <image path="sweater-10474-deer-jk11?$medium$"/>  </image-group>  <image-group view-type="medium" variation-value="TRAIN">  <image path="sweater-10474-train-jk11?$medium$"/>  </image-group>  <image-group view-type="small">  <image path="sweater-10474-model-nk11?$small$"/>  </image-group>  <image-group view-type="small" variation-value="CANDY CANE BEAR">  <image path="sweater-10474-bear-jk11?$small$"/>  <image path="sweater-10474-model-jk11?$small$"/>  </image-group>  <image-group view-type="small" variation-value="FAIRY">  <image path="sweater-10474-fairy-jk11?$small$"/>  <image path="sweater-10474-model-jk11?$small$"/>  </image-group>  <image-group view-type="small" variation-value="GINGERBREAD GIRLS">  <image path="sweater-10474-ginger-jk11?$small$"/>  <image path="sweater-10474-model-jk11?$small$"/>  </image-group>  <image-group view-type="small" variation-value="JOY SNOWMEN">  <image path="sweater-10474-snow-jk11?$small$"/>  <image path="sweater-10474-model-jk11?$small$"/>  </image-group>  <image-group view-type="small" variation-value="REINDEER AND TREE">  <image path="sweater-10474-deer-jk11?$small$"/>  <image path="sweater-10474-model-jk11?$small$"/>  </image-group>  <image-group view-type="small" variation-value="TRAIN">  <image path="sweater-10474-train-jk11?$small$"/>  <image path="sweater-10474-model-jk11?$small$"/>  </image-group>  </images>  </product> Creating Image URL DW getImage APIs would be used to create image URLs in the storefront.  For example, to show a hero image  pdict.Product.getImage('large',0).url Zoom Scene7 alternate viewer JavaScript code will be modified and adapted to display the alternate viewer functionality as per the comps shown in the functional specification document. Following is the URL for - [http://s7testweb.adobe.com/SampleCode/html/Misc/Check\_Imageset.html](file:///C:/Ravi/Projects/PureFishing/TSD/http%3A%2F%2Fs7testweb.adobe.com%2FSampleCode%2Fhtml%2FMisc%2FCheck_Imageset.html) Flow Diagram Scene7 integration for <Client> will leverage the LINK partner integration.   Configuration data  |  |  |  | | --- | --- | --- | | Scene7 | | | | http URL | Dev/Test: http://s7xx.scene7.com/is/image/<Client Name>  Production: http://s7xx.scene7.com/is/image/<Client Name> | Dev/Test: http://s7xx.scene7.com/is/image/<Client Name>  Production: http://s7xx.scene7.com/is/image/<Client Name> | | https URL | Dev/Test: https://s7xx.scene7.com/is/image/<Client Name>  Production: https://s7xx.scene7.com/is/image/<Client Name> | Dev/Test: https://s7xx.scene7.com/is/image/<Client Name>  Production: https://s7xx.scene7.com/is/image/<Client Name> |  Image Flow Diagram DW will import all image types from <Client>, once imported the images will be saved in the master catalog image folder. Since we are not going to use DIS, <Client>, will provide all image types. We will implement the following image classes in BM:   Integration Specs - Media Pixels DW will create a place holder content asset to be included in the footer. <Client Name> will be able to leverage it later on to add any media pixels needed. Content asset name TBD. Integration Specs - Omniture Omniture Site Catalyst integration for <Client Name> is illustrated in the diagram below.    Following three Omniture related assets will be added for each page:   |  |  |  |  | | --- | --- | --- | --- | |  | Name | Type | Description | | 1 | Omniture | ISML | Included on footer with a flag to include omniture tags based on the setting in site preference | | 2 | Omniture\_asset\_include | ISML | The ISML populates all the required omniture variables based on the type of the page | | 3 | libOmniture | script | Used to process the logic to generate values for the omniture variables. |  Omniture Script Configuration  |  |  |  | | --- | --- | --- | | Custom Attribute | Type | Description | | Id | String | ID | | scriptBody | String | Omniture script variables |  Integration Specs - Google Analytics Google analytics JavaScript will be added to all the pages that are listed in the functional specification document. ISML templates will be customized to pass the dynamic variable to the Google JavaScript code snippet. Implementation details Reach out to Reji James or Ghulam Abbas for setting the test credentials in the attributes listed below. Following custom attributes will be created.   |  |  |  |  | | --- | --- | --- | --- | |  | Name of the attribute | System Object | Type | | 1 | googleAnalyticsId | Site Preference | String | | 2 | googleDomainName | Site Preference | String |  Steps Create a new common/google\_analytics.isml template to host Google Analytics code. Include common/google\_analytics.isml on pages we need to track through the Google analytics. Or reference it in the footer.isml if Solstice decides to track all pages. High level common/google\_analytics.isml implementation logic:  Try and use new code from Google analytics. TBD.   1. Get Google Analytics Java Script file from Google and write it on the page.   **e.g.**  <script type=*"text/javascript"*>  **var** gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");  document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));  </script>   1. Invoke tracker code using <Client Name> specific analytics id.   e.g.  <script type=*"text/javascript"*>  **try** {  **var** pageTracker = \_gat.\_getTracker("${dw.system.Site.getCurrent().getCustomPreferenceValue('googleAnalyticsId')}");  pageTracker.\_trackPageview();  // you can pass any pipeline data to Google Analytics using \_trackPageview..  } **catch**(err) {}  </script>   1. Define custom site preference to store the Google analytics account id.  Page Specific codeOn-Page Code for Google Analytics TO BE USED THROUGHOUT THE SITE - NOT ON ORDER CONFIRMATION PAGE: <!-- Start Google Analytics Code -->  <script type="text/javascript">  var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");  document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));  </script>  <script type="text/javascript">  try {  var pageTracker = \_gat.\_getTracker("${dw.system.Site.getCurrent().getCustomPreferenceValue('googleAnalyticsId')}");  pageTracker.\_setDomainName("${dw.system.Site.getCurrent().getCustomPreferenceValue('domainName')} ");  pageTracker.\_trackPageview();  } catch(err) {}  </script>  <! -- End Google **Analytics Code -->** ORDER CONFIRMATION PAGE – Single Item HERE IS THE GOOGLE ANALYTICS CODE FOR ORDERS WITH one ITEM  <!-- Google Analytics ROI Tracking Code -->  <script type="text/javascript">  var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");  document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));  </script>  <script type="text/javascript">  try {  var pageTracker = \_gat.\_getTracker("${dw.system.Site.getCurrent().getCustomPreferenceValue('googleAnalyticsId')}");  pageTracker.\_setDomainName("${dw.system.Site.getCurrent().getCustomPreferenceValue('domainName')}");  pageTracker.\_trackPageview();  pageTracker.\_addTrans(  "lastOrder.OrderNumber", // order ID - required  "", // affiliation or store name  "lastOrder.NetTotal", // total - required  "lastOrder.Tax", // tax  "lastOrder.Shipping", // shipping  "", // city  "", // state or province  "" // country  );    // add item might be called for every item in the shopping cart  // where your ecommerce engine loops through each item in the cart and  // prints out \_addItem for each  pageTracker.\_addItem (  "lastOrder.OrderNumber1", // order ID - necessary to associate item with transaction  "lastOrder.Sku1", // SKU/code - required  "lastOrder.ProductName1", // product name  "", // category or variation  "lastOrder.UnitPrice1", // unit price - required  "lastOrder.Quantity1" // quantity – required  "lastOrder.GiftBoxSelected1" // Yes/No – required  );  pageTracker.\_trackTrans(); //submits transaction to the Analytics servers  } catch(err) {}  </script>  <!-- End Google Analytics ROI Tracking Code --> ORDER CONFIRMATION PAGE – Multiple Items HERE IS THE GOOGLE ANALYTICS CODE FOR ORDERS WITH multiple ITEMS  <!-- Google Analytics ROI Tracking Code -->  <script type="text/javascript">  var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");  document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));  </script>  <script type="text/javascript">  try {  var pageTracker = \_gat.\_getTracker("${dw.system.Site.getCurrent().getCustomPreferenceValue('googleAnalyticsId')}");  pageTracker.\_setDomainName("${dw.system.Site.getCurrent().getCustomPreferenceValue('domainName')}");  pageTracker.\_trackPageview();  pageTracker.\_addTrans(  "lastOrder.OrderNumber", // order ID - required  "", // affiliation or store name  "lastOrder.NetTotal", // total - required  "lastOrder.Tax", // tax  "lastOrder.Shipping", // shipping  "", // city  "", // state or province  "" // country  );    // add item might be called for every item in the shopping cart  // where your ecommerce engine loops through each item in the cart and  // prints out \_addItem for each  pageTracker.\_addItem(  "lastOrder.OrderNumber1", // order ID - necessary to associate item with transaction  "lastOrder.Sku1", // SKU/code - required  "lastOrder.ProductName1", // product name  "", // category or variation  "lastOrder.UnitPrice1", // unit price - required  "lastOrder.Quantity1" // quantity - required  "lastOrder.GiftBoxSelected1" // Yes/No – required  );  pageTracker.\_addItem(  "lastOrder.OrderNumber2", // order ID - necessary to associate item with transaction  "lastOrder.Sku2", // SKU/code - required  "lastOrder.ProductName2", // product name  "", // category or variation  "lastOrder.UnitPrice2", // unit price - required  "lastOrder.Quantity2" // quantity - required  "lastOrder.GiftBoxSelected2" // Yes/No – required  );  pageTracker.\_trackTrans(); //submits transaction to the Analytics servers  }  catch(err) {}  </script>  <!-- End Google Analytics ROI Tracking Code --> JavaScript On/Off BehaviorBrowser Support Matrix Browser Matrix - Desktop and Tablets   |  |  |  | | --- | --- | --- | | BROWSER | RESPONSIVE DESIGN | DESKTOP VERSION (FULL SITE) | | IE 8 | No | Yes | | IE 9 |  |  |  Mobile Both brand sites would launch with mobile storefronts. DW will leverage the responsive design implementation built into SiteGenesis reference storefront application.  For specific requirements, check the mobile FSDs and the mobile UI drops.   * Responsive Mobile Design Summary * Comps * Mobile Device Specs * Devices and Viewports  Accessibility Please specify any Special accessibility requirements Scheduled Jobs Jobs would be scheduled in Commerce Center (Business Manager); following diagram shows the interaction of various components for each job.  The table below shows the list of jobs their corresponding details.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Name** | **Duration/Time** | **Source** | **I/O** | **Environment** | **Resources** | **Description** | | ImportCatalogs | Nightly | SFTP | Data fed into DW | Staging | Catalog and Product | Feed imported into staging then replicated to production | | ImportPricebooks | Nightly | SFTP | Data fed into DW | Staging | Pricebook | Feed imported into staging then replicated to production | | OrderExport | Every 10 minutes | DW | Data fed to Client | Production | Order | Feed exported directly from production. | | ImportInventory | Every 4 hours | SFTP | Data fed into DW | Production | Inventory | Feed imported directly into production | | ImportOrderStatus | Every 4 hours | SFTP | Data fed into DW | Production | Order | Delta Feed imported directly into production. | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  Manual Jobs The following data entities will be setup or imported in commerce center manually.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Entity Name** | **Source** | **I/O** | **Environment** | **Resources** | **Description** | | GeoLocations DB | SFTP | Data fed into DW | Staging | GeoLocation (long/lat zip code mapping) | Feed imported into staging then replicated to production | | Stores (setup job, monthly?) | SFTP | Data fed into DW | Staging | Store | Feed imported into staging then replicated to production | | Customers | SFTP | Data fed into DW | Production | Customer | Feed imported directly into production before going live. | | Tax Table | SFTP | Data fed into DW | Staging | Tax | Manual setup or a feed into staging and then replicated to production. See sample below. | | Shipping Methods | Manual |  | Staging | Shipping Method | Manual setup and then replicated to production. See sample below. | | Payment Methods | Manual |  | Staging | Payment Method (VI, MC, AX, Discover) | Manual setup and then replicated to production. |  Data Model Extensions For implementing the <Client Name> storefront deviations from SiteGenesis, certain system objects will be extended and new custom objects will be created. The following sections describe these extensions. System ObjectsProduct Check the Data Mappings spreadsheet to get the complete list of product and category attributes. All the system object definitions (custom) should be identified in here or referred as an attachment to this. Category Check the Data Mappings spreadsheet to get the complete list of product and category attributes. Order  |  |  |  | | --- | --- | --- | | Custom Attribute | Type | Description | |  |  |  |  Global and Site Preferences  |  |  |  | | --- | --- | --- | | Custom Attribute | Type | Description | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  |  Custom ObjectsFeed Job Configuration  |  |  |  | | --- | --- | --- | | Custom Attribute | Type | Description | | Id | String | ID | | description | String | Description of the SFI job | | fromEmail | Email | From email address for the SFI job email | | tasksXML | Text | Task xml has the details for each job i.e. remote server location, credentials, dw impex folder, file name, notification email list. |  Cartridge Structure Cartridge is a container that contains all the resources for the particular storefront or integration module. A cartridge contains the following folders   * Forms – form field definitions and validation * Pipelines – business process flows * Scripts – scripts for executing a particular business process * Static – all static content for the storefront e.g. images, css, javascripts etc. * Templates – all page templates for storefront * Webreferences – web service definition files   <Client Name> will be built using the following cartridges as described below:   |  |  | | --- | --- | | Cartridge Name | Description | | <Client>-core | Core cartridge that will contain all SiteGenesis resources, along with all changes that will be made for <Client Name> that can be used across all their storefronts. | | <Brand1> | Brand1 NA site specific code and static resources. | | < Brand2> | Brand2 NA site specific code and static resources. | | int\_cybersource | Cartridge that contains all resources for integrating with Cybersource. | | int\_simplefeeds | Cartridge that contains all resources that automates importing feeds. | | int\_buildmanager | Cartridge that contains all resources to help code builds on staging. | |  |  |  Environment and Build Process Following diagram displays the <Client Name> environment    Build process for <Client Name> will be automated. The following steps would be performed by a build process   * Download code from <Client Name> SVN * Configure the build process. * Build process will create the cartridge and upload it to the staging server * Build process will activate the version of the uploaded code * Build process will send out emails about the new build  Build Cartridge (Optional)Build FrequencyCentral Build Server (Optional)Build ResponsibilitiesAliases DW platform version 2.10.2 onwards, redirects can be configured in Aliases section which is under <Site> and Site URLs section within Commerce Center (Business Manager). Following is a sample configuration for <Client Name> –  *{*  */\**  *\* The file is used to configure host names for a site.*  *\* \_\_version must be set to "1"*  *\*/*    ***"\_\_version" = "1",***    */\**  *\* Settings section is used to configure main HTTP and HTTPS hostnames for a site.*  *\**  *\*/*  ***"settings" = {***  ***"http-host" = "www.brand1.com",***  ***"https-host" = "secure. brand1.com"***  ***}****,*    */\**  *\* Host name definitions.*  *\* The following section allows to define additional hostnames associated with the site.*  *\* With each hostname it is possible to define set of redirect rules.*  *\**  *\*/*    */\**  *\* Host redirect from SiteGenesis.com to www.sitegesis.com.*  *\* All URLs coming to SiteGenesis.com (http://SiteGenesis.com/\*) will be permanently redirected (301) to www.SiteGenesis.com*  *\*/*  ***" brand1.com": [***  ***{***  ***"host": "www.Brand1.com",***  ***"path":"/",***  ***}***  ***],***    ***"new.Brand1.com": [***  ***{***  ***"host": "www.Brand1.com",***  ***"path":"/",***  ***}***  ***],***  */\**  *\* Host redirect to specific pipeline.*  *\* Hostname-only URL (e.g. http://electronics.SiteGenesis.com/) will result in a call of Search-Show pipeline.*  *\**  *\* "try.Brand1.com": [*  *\* {*  *\* "pipeline": "Try-Brand1",*  *\* "params" : {"cid"="try-Brand1"}*  *\* }*  *\* ],*  *\*/*  */\**  *\* Host redirect based on user agent header.*  *\* With each rule it is possible to associate precondition based on content of "user-agent" http header.*  *\* In the example below an HTTP requests with "User-Agent" header containing "iphone" or "ipod" will be redirected (301) to apple.SiteGenesis.com*  *\* and those with "blackberry" to bb.SiteGenesis.com.*  *\**  *\* "www.SiteGenesis.com": [*  *\* {*  *\* "if-agent-contains": ["iphone","ipod"],*  *\* "host": "apple.SiteGenesis.com",*  *\* },*  *\* {*  *\* "if-agent-contains": ["blackberry"],*  *\* "host": "bb.SiteGenesis.com",*  *\* }*  *\* ],*  *\*/*    *}* Multiple StorefrontsStore BM front URLsCartridge StructureCartridge DefinitionsStyle SheetStyle GuideDevelopment UX Studio / Sandbox - Special SettingsCartridge StructureCoding StandardsNaming ConventionsBusiness ManagerRedirects DW platform version 2.10.2 onwards, redirects can be configured in Aliases which is under each storefront and Site URLs section within Business Manager. Following configurations were done for the <client> –  *{*  */\**  *\* The file is used to configure host names for a site.*  *\* \_\_version must be set to "1"*  *\*/*    ***"\_\_version" = "1",***    */\**  *\* Settings section is used to configure main HTTP and HTTPS hostnames for a site.*  *\**  *\*/*  ***"settings" = {***  ***"http-host" = "www.<sitename>.com",***  ***// "https-host" = "secure.sitegenesis.com"***  ***}****,*    */\**  *\* Host name definitions.*  *\* The following section allows to define additional hostnames associated with the site.*  *\* With each hostname it is possible to define set of redirect rules.*  *\**  *\*/*    */\**  *\* Host redirect from sitegenesis.com to www.sitegesis.com.*  *\* All URLs coming to sitegenesis.com (http://sitegenesis.com/\*) will be permanently redirected (301) to www.sitegenesis.com*  *\*/*  ***"xxxxxxx.com": [***  ***{***  ***"host": "www.xxxxxxx.com",***  ***"path":"/",***  ***}***  ***],***    ***"new.xxxxx.com": [***  ***{***  ***"host": "www.xxxxxx.com",***  ***"path":"/",***  ***}***  ***],***  */\**  *\* Host redirect to specific pipeline.*  *\* Hostname-only URL (e.g. http://electronics.sitegenesis.com/) will result in a call of Search-Show pipeline.*  *\**  *\* "try.sheercover.com": [*  *\* {*  *\* "pipeline": "Try-xxxxxx",*  *\* "params" : {"cid"="try-xxxxxx"}*  *\* }*  *\* ],*  *\*/*  */\**  *\* Host redirect based on user agent header.*  *\* With each rule it is possible to associate precondition based on content of "user-agent" http header.*  *\* In the example below an HTTP requests with "User-Agent" header containing "iphone" or "ipod" will be redirected (301) to apple.sitegenesis.com*  *\* and those with "blackberry" to bb.sitegenesis.com.*  *\**  *\* "www.sitegenesis.com": [*  *\* {*  *\* "if-agent-contains": ["iphone","ipod"],*  *\* "host": "apple.sitegenesis.com",*  *\* },*  *\* {*  *\* "if-agent-contains": ["blackberry"],*  *\* "host": "bb.sitegenesis.com",*  *\* }*  *\* ],*  *\*/*    *}* Site redirect URLStatic mappingSearch Engine Optimization  * Work with solution strategist on 301 redirections which are supported exclusively through business manager. * Work with Robots txt file to understand capabilities  AliasesSource ControlSVN Link and Accounts / Team ProcessBackup ScheduleSource/ Metadata Update/ Recovery ProcedureMonitoringSecurityProvisioning of 2-level AuthenticationSSL Certificates for Launch and Development needsPOD Control CenterAccounts / Access ListPOD DetailsOutgoing address of PODSingle Sign-OnGlobal Error Codes and MessagesInternationalization / LocalizationSite PerformanceKPIsConversionsData VolumesAPI QuotasPlatform QuotasCanonicalization To ensure Search Engines don’t penalize website for pages that can be reached from different navigations, following line of code will be added to product detail page.  *<link rel="canonical" href="${URLUtils.url('Product-Show','pid',<* *pdict.Product.ID>, 'cgid', <pdict.Product.primaryCategory.ID>)}">*  Similarly add canonical links for Category Landing page as follows.  *<isif condition="${pdict.CurrentHttpParameterMap.cgid.submitted}">*  *<link rel="canonical" href="${URLUtils.url('Search-Show','cgid', pdict.CurrentHttpParameterMap.cgid)}" />*  *<iselseif condition="${pdict.CurrentHttpParameterMap.q.value!=null}">*  *<link rel="canonical" href="${URLUtils.url('Search-Show','q', pdict.CurrentHttpParameterMap.q.value)}" />*  *</isif>*  For more information on canonicalization visit [http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=139066](file:///C:/Users/jmoody/AppData/Local/Microsoft/Windows/Temporary%20Internet%20Files/Low/Content.IE5/AIKYGGSJ/http:/www.google.com/support/webmasters/bin/answer.py?hl=en&answer=139066) Data Replication Data replication will occur automatically at 4:00AM. Replication will occur from staging to development and from staging to production. The following items will be included in automatic replications:   * Site catalogs * Content assets * Content slots * Price Books * Indexes * Campaigns * Promotions  Resource Usage EstimatesCyber Monday Estimates  * Peak visits / hr - 3,853 visits. * Peak page views/hr - 20,306 page views * Peak orders/hr - 1000 orders  Expected Averages    |  |  | | --- | --- | | Estimates | Year/Season/Day Averages | | Order per hour |  | | Visits per day/year |  | | Page views per hour |  | | Abandoned Carts |  | | Registrations |  | | Conversion Rate |  | | Number of searches (optional) |  | | Order History / Detail Requests |  | | Total Product Variants |  | | Total Feed File Sizes /Row Counts per Feed |  | | Total Custom Objects |  |  Migration Estimates Please obtain this information from the client   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Item** | **Import Size** | | **Description** | | | Customer Accounts | | 100,000,000 Records, 230MB; | |  | | Products | 106,000 Rows / 25MB | |  | | | Price Books | 6,000 Rows / 25MB | |  | | | Inventory | 90,000; 6MB | | There will be multiple files per store updated hourly | |   Any unusual volume in migration plan needs to be addressed with available solutions working with Demandware. Email Accounts Please identify all the sender /destination email accounts needed for the implementation. This could be email setting for Quota Alerts, Contact Us and all such triggered notifications.   |  |  | | --- | --- | | **Email IDs** | **Purpose** | |  |  | |  |  | |  |  |  Transaction Emails  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Email** | **Send By DW** | | **Comments** | | | Order ACK | |  | |  | | Shipping Confirmation |  | |  | | | Welcome Email |  | |  | | | Forgot Password |  | |  | | | Send To A Friend |  | |  | | | CC for STF |  | |  | | | WishLIst |  | |  | |  Key Performance Indicators Some KPIs are listed below :   * Site traffic * Unique visitors versus returning visitors * Time on site * Page views per visit * Day part monitoring (when site visitors come) * Newsletter subscribers * Texting subscribers * Chat sessions initiated * Facebook, Twitter, or Pinterest followers or fans * Pay-per-click traffic volume * Blog traffic * Number and quality of product reviews * Brand or display advertising click-through rates * Affiliate performance rates  Static File Optimization PlanData Retention  |  |  | | --- | --- | | **Element** | **Clean-up plan** | | Orders (after export) | Delete objects older than x days. All order history is on external system. | | Back in stock notifications: | Delete objects older than x days | | Payment | Delete objects older than x days | | Profile | Delete objects that have not logged in in over x days |  Caching Strategy The site will mostly use similar caching as SG default. The changes are as follows:   * Partners and Platform pages will be cached for 24 hrs. * Order history and order details pages will not be cached since we are using direct web service calls  Cartridge Structure A Cartridge is a container that includes all the resources for a particular storefront or integration module. A cartridge contains the following folders:  Forms – form field definitions and validation  Pipelines – business process flows  Scripts – scripts for executing a particular business process  Static – all static content for the storefront e.g. images, css, javascripts etc.  Templates – all page templates for storefront  Webreferences – web service definition files  For configurations of the cartridges, see the integration sections earlier in this document. <Site Name> will be built using the following cartridges described below:   |  |  |  |  | | --- | --- | --- | --- | | **CARTRIDGE NAME** | **DESCRIPTION** | **LINK CARTRIDGE?** | **LINK VERSION** | | **core\_<sitename>** | Main Cartridge, holding core business logic and templates | NO | - | | **app\_<sitename>\_richUI** | This cartridge overrides some of the ISML templates from the core. The most powerful enhancements here are a completely new app.js (without a separate product.js) and a responsive.js file that extends the app namespace. | NO | - | | **app\_<sitename>\_us** | US specific content. Inherits most information from app\_<Client>\_core | NO | - | | **app\_<sitename>\_uk** | UK specific content. Inherits most information from app\_<Client>\_core | NO | - | | **int\_bazaarvoice** | BazaarVoice Reviews, Ask & Answer | YES https://xchange.demandware.com/docs/DOC-10627 | 5.2 | | **int\_cybersource** | Cybersource Integration | YES https://xchange.demandware.com/docs/DOC-10535 | 1.1.0 | | **int\_paypal** | PayPal Integration | YES  https://xchange.demandware.com/docs/DOC-10617 | 1.0.2 | | **int\_digitalriver** | Digital River payment method | NO | - | | **int\_analytics** | Google Analytics | NO | - | | **int\_grayskull** | Grayskull OMS | NO | - | | **int\_salestax** | Sales Tax Office | NO | - | | **int\_ups** | UPS Rating | NO | - |  Site Cartridge Paths  |  |  | | --- | --- | | **Site Name** | **Cartridge Path** | | **< Site Name> US** | int\_ups:int\_bazaarvoice:int\_grayskull:int\_order\_export:int int\_cybersource:int\_salestax: app\_< Site Name> \_us:app\_< Site Name> \_richUI:core\_< Site Name> | | **< Site Name> CA** | int\_order\_export: int\_digitalriver: app\_**< Site Name>** \_fr:app\_< Site Name> \_richUI:core\_< Site Name> | | **< Site Name> CN** | int\_grayskull:int\_order\_export: int\_digitalriver: app\_< Site Name>\_uk:app\_< Site Name> \_richUI:core\_< Site Name> |  Coding Standards For any standards not defined below, follow Demandware Coding Standards, available at <https://documentation.demandware.com/display/DOC131/Storefront+coding+standards+and+conventions> |
|  |