|  |  |  |
| --- | --- | --- |
| Title Shop Runner Design Specifications | | Document ID |
| Author  Casmon Gordon | Approved by | Creation Date  10/1/2015 |
| Document Type  Form | | Modified Date  10/1/2015 |
| Function(s)  Shop Runner Product Feed | | System  Product Feeds |
| Build Information  1.7 | |  |
| **This is an ISO controlled document. Only the latest revision, which appears on the company database, may be used as a working document. Any printed copy is uncontrolled and is the responsibility of the user to verify validity prior to use.** | | Printed Date  9/9/2015 |

# **1.0 PURPOSE**

To document design specifications for development code changes. This document describes and outlines the code modifications and enhancements which will be performed in order to satisfy the requirement specification document.

# **2.0 SCOPE**

This form is used for any design specifications. It outlines the scope of the code modifications that will be made to meet that which was specified in the requirements.

# **3.0 RESPONSIBILITY**

It is the responsibility of the Dir. of Development or his delegate to use, maintain, and file this form (when completed).

# 4.0 AUDIENCE

The intended audiences for this document are the relevant stakeholders, the project development teams, technical architects, database designers, testers and vendors.

# 5.0 DESIGN OVERVIEW

This section describes the overall high level design of the feature or system.

The modifications made to meet the requirements for the Shop Runner Feeds basically involved cloning the iGo product data feed existing functionality and then modifying the code in order to produce some very unique features tat are specific to Shop Runner. These include:

1. The feed will only be needed to be called once in hour\_glass as opposed to the prior process of being called multiple times for each of the seven(7) brands
2. There is a requirement for each feed file for each brand to be sent to a specific sftp host. This differs from all other feeds where all the files are sent to the same sftp host.
3. There were additional information needed in the feed file which requires that columns be added to the xsql query xml file and the corresponding changes to accommodate the new fields.
4. The xsql query xml file was changed to accept the brand information the is being passed in so that the query results are narrowed which improves the speed and performance of the query.
5. The code changes were also done in a way to preserve the existing functionality of the other feeds.

# 6.0 ARCHITECTURE

Please see below for a more detailed description of the aspects and consideration made when designing the change required to fulfill the requirements for the Shop Runner project..

# 6.1 DESIGN CONSIDERATIONS

1. Implement the required changes for Shop Runner.
2. Preserve existing functionality and behaviour of the other feeds.
3. Improve the overall speed and performance of the feed process.
4. Make the feed more robust by implementing fail-safes and exception handling.

# 6.2 DESIGN PATTERNS

Data Access Objects.

Value Objects.

Business Delegate.

Model Set (Internal and External).

Data Transfer Objects.

# 6.4 TRADE-OFFS

Currently there are two versions of the product feed viz.

ProductDataFeed.java and MBPProductDataFeed.java.

There was the desire to rewrite the code so that only one implementation of the product feed would be needed. The concern was that doing that may need more time and resources to verify that the existing functionality would be maintained. The compromise for now is to keep both implementations as is and integrate the required changes to accomplish what is required for Shop Runner.

# 7.0 IMPLEMENTATION DETAILS

# 7.1 Java Changes

MBPProductDataFeed.java

ShopRunnerdataObject.java

ShopRunnerFeedConfig.java

AbstractPartnerDataFeed.java

XMLUtil.java

ProductFeedUtil.java

ProductFeedProperties.java

MBPProductRelationshipAccessor.java

IProductConstants.java

Product.java

ProductBase.java

ProductSku.java

BrandInfo.java

ConnectionInfo.java

SFTPProductFeedPoster.java

FTPProductFeedPoster.java

XQSLDataFeedDAO.java

# 7.2 XML and Properties Changes

|  |  |
| --- | --- |
| File | Comments |
| selectAvailableProdcts.xml | Added new shop runner columns. added/enabled the conditional 'brand\_ids' parameter/variable |
| XSQLConfig.xml | Made changes in order to test the new Shop Runner feed behavior. |
| productfeed.properties | Added properties for shop runner. Added pseudo fruit bouquets brand info for improving consistency in java logic. added more brand details. |
| mercent.xsd | Added the columns/fields for the new information that is required for Shop Runner. |
|  |  |

# 7.3 XML Beans Changes

<xs:schema>

<xs:complexType name="product\_base">

<xs:element ref="field3"/>

<xs:element ref="seosearchable"/>

<xs:element ref="canonical\_url" minOccurs="0"/>

</xs:complexType>

<xs:element name="canonical\_url" type="xs:string">

<xs:annotation>

<xs:documentation>bot friendly</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="field3" type="xs:string"/>

<xs:element name="seosearchable" type="xs:string"/>

</xs:schema>

# 7.4 Database Changes

There were no database changes made for this project.

# 7.5 Design and Code Change Highlights

1. Performance was always considered when making changes.
2. Shop Runner only needs to be called once with list of brands. Improves performance since expensive DB calls now being made once compared top previous multiple times.
3. No brands passed defaults to entire list of brands. Safety feature.
4. Refactor code where possible to avoid using literals in code such as “N” or “YES” etc. Made them constants (IProductConstants.java)
5. If a fixed list is always used to validate or verify a value, then the fixed list is made a constant set eg

public final Set<String> FRUITBOUQUET\_CLASS\_IDS = new

HashSet<String>(Arrays.asList(new String[] {"24","26"}));

1. Added specific feed level property to disable FTP feature (General) Default value is to send Ftp files but can be set to disable FTP
2. If brand info is missing then default to brand info provided in properties file.
3. Brand node in XML now displays user configurable brand description; no longer brand\_id info.
4. Added seosearcheable flag taken directly from catentry database table, and is no longer calculated.
5. Renamed xml data node from base\_code base\_code\_title to parent\_sku and product\_description. This was done by modifying the XmlCursor object directly at runtime and did not involve changing the XMLBeans implementation etc.
6. Descriptive brand name is now being maintained via the productfeed.properties file; e.g. <brand>1-800-Flowers</brand>
7. There is no longer the need for calculating what the canonical domain url for each brand should be as it is now also being maintained via the properties file.
8. Added ability to send multiple files to different sftp locations.
9. SFTPProductFeedPoster class was modified extensively to provide the ability to send the feed file for each brand to a specific predetermined sftp host location.
10. Pseudo Fruit Bouquet is implemented by creating a pseudo brand ‘1001FB’. This is needed because fruit bouquet uses the same brand id as the 18F brand and can only be identified by the class\_id field in CATENTRY table. During data processing of the brand the ‘1001FB’ pseudo brand is temporary rebranded as ‘1001’ and a Boolean flag indicating this special case is set so that the normal execution path is followed except when the output feed file is the be created; at which point the appropriate file name and sftp host will be retrieved and used.
11. FTPConnector class was modified so that a disconnect method added to explicitly close data connection after the file transfer is complete. This connection would previously remain open until the connection times out.
12. BrandInfo class was modified to now include an inner class with connection details. (ConnectionInfo.java) This simplifies the sftp process by having this information readily available during the processing of the feed files.
13. Each brand now has its own specific sftp host connection information.
14. There is also the ability to turn off the sftp feature for Shop Runner by setting the ‘shoprunnerDataFeed.transferFile=N’ property. This is useful when you do not want to risk affecting feeds that are running comfortably.
15. Filter products output by XCATENTRY.onsitesearchenabledflag value is true.
16. Filter product sku category information by seosearcheable flag is true.
17. Also added local flag to the output feed files.
18. Removed existing System.out.println/System.out.printf calls and replaced them with log4j equivalent which can be controlled at runtime via log4j API framework.

# 8.0 ASSUMPTIONS AND CONSTRAINTS

This section describes any assumptions and constraints that were made about both the infrastructure and general direction for technology.

|  |  |
| --- | --- |
| # | Assumptions and Constraints |
| 1 | User Acceptance Test / System Acceptance Test Environment will be available for performing Usability testing, User Acceptance testing, Installation & Configuration testing and Performance Testing. |
| 2 | General Architecture principles based on past experiences and 1800flowers.com best practices & methodologies will be used in designing the solution. |
| 3 | The basic TCP/IP (HTTP) protocol will be the only one used to access the application |
| 4 | The web browser will be the primary client used by employees and public users |
|  |  |

# 9.0 APPENDIX A: Acronyms, Abbreviations, Terms and Definitions

|  |  |
| --- | --- |
| WAS | IBM Websphere Application Server |
| WCS | IBM Websphere Commerce |
| Solr | This is based on Apache Solr and provides integrated search management tooling for business users in the Management Center. |
| CMC | IBM Websphere Management Center |
| API | Application Program Interface |
| BO | Business Object |
| DAO | Data Access Object |
| DHTML | Dynamic Hypertext Markup Language |
| TCP | Transfer Control Protocol |
| IP | Internet Protocol |
| DMZ | De-Militarized Zone – Term for the portion of the network between the external Internet and the internal private network. The DMZ is protected from the outside by a Firewall. |
| GUI | Graphical User Interface |
| EAR | Enterprise Archive |
| EJB | Enterprise JavaBeans |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol over SSL |
| SSL | Secure Socket Layer |
| LAN | Local Area Network – Communications network confined to the same physical building. |
| SMTP | Simple Mail Transport Protocol – Standard method of delivering internet email messages |
| J2EE | Java Enterprise Edition |
| JAR | Java Archive |
| JCA | Java Connector Architecture |
| JDBC | Java Database Connectivity |
| JRE | Java Runtime Environment |
| JSP | Java Server Pages |
| JVM | Java Virtual Machine |
| POJO | Plain Old Java Object |
| SQL | Structured Query Language |
| UML | Unified Modeling Language |
| WAR | Web Archive |
| XML | Extensible Markup Language |
| XLST | Extensible Style Language Transformation |
| IDE | Integrated Development Environment |

# 10.0 APPENDIX B: Products & Tools

The following software components will be utilized in the architecture being designed. New versions of software may be released during the development of the system. The implementation of these new versions will be evaluated on an individual basis in determining if and when they will be implemented. Cross compatibility issues must be addressed before implementing any new versions of software products.

|  |  |  |  |
| --- | --- | --- | --- |
| Software/Tool | Version | Source | Description |
| J2EE |  |  | Java Enterprise Edition for Enterprise services |
| SVN |  |  | Subversion - source code management and control. |
| Eclipse (RAD) |  |  | Integrated Development Environment |
| J2SDK |  |  | Java SDK for API |
| Oracle 11g |  | http://www.oracle.com | Oracle Server for database persistence |

|  |  |  |  |
| --- | --- | --- | --- |
| Infrastructure Software | Version | Source | Description |
| Java JDK |  |  | Java Runtime for Portal and Application Server |
| Websphere Application Server |  |  |  |
| Websphere Commerce |  |  |  |
| Oracle 11g |  | http://www.oracle.com | Oracle Server for database persistence |
| SVN |  |  | Subversion - source code management and control. |

# 11.0 APPENDIX B: Configuration files

Below are some of key configuration files used in Online Screening Tool -

**Application Configuration File**

Below are three key application configuration files -

a) web.xml - The Web application descriptor provides the application server with information about the Web resources in the application.

b) wc-server.xml - The wc-server.xml file is the deployment descriptor for the Websphere Commerce application container. (Location= /opt/IBM/WebSphere/AppServer/profiles/MBPDev01/installedApps/WC\_MBPDev01\_cell/WC\_MBPDev01.ear/xml/config)

c) wc-search.xml - The wc-search.xml file is used to configure the Solr integrated search functionality. (Location = /opt/IBM/WebSphere/AppServer/profiles/MBPDev01/installedApps/WC\_MBPDev01\_cell/WC\_MBPDev01.ear/xml/config)

|  |  |  |
| --- | --- | --- |
| **Approved By Name** | **Approved By Title** | **Approved on Date** |
| Casmon Gordon | Development Lead |  |
|  | Development Director |  |