Software Architecture and

Design Document

Of the

Unloaded Shipping Container

Display Pages

Extension of Acme’s WMS System

Prepared for Acme, Inc.

Version 1.0.0

Prepared by Henry Landry

March 29, 2015

Contents

[1. Introduction 1](#_Toc415771127)

[1.1 Purpose 1](#_Toc415771128)

[1.2 Intended Audience 1](#_Toc415771129)

[1.3 Product Scope 1](#_Toc415771130)

[1.4 References 1](#_Toc415771131)

[1.5 *Reminders*: 1](#_Toc415771132)

[2. System Architecture 2](#_Toc415771133)

[2.1 Layer/Call Architecture Diagram 2](#_Toc415771134)

[2.2 Architecture Details of the Layers 2](#_Toc415771135)

[2.3 Design Rational 3](#_Toc415771136)

[2.4 Assumptions 3](#_Toc415771137)

[3. Component Design 4](#_Toc415771138)

[3.1 UI Layer 4](#_Toc415771139)

[3.1.1 WMSActivityByRefNbr.aspx Web page 4](#_Toc415771140)

[3.1.2 WMSCntrDetailsByRefNbr.aspx Web page 6](#_Toc415771141)

[3.2 Business Layer 8](#_Toc415771142)

[3.2.1 Service Method Relationship Diagram 8](#_Toc415771143)

[3.2.2 GetCntrDataByReferenceAndDate Web Service Method 8](#_Toc415771144)

[3.2.3 GetCntrDataByReferenceNbr Web Service Method 9](#_Toc415771145)

[3.2.4 SummarizeCntrDataByReferenceNbr Web Service Method 10](#_Toc415771146)

[3.2.5 CntrSummaryDataByReferenceType Class 11](#_Toc415771147)

[3.2.6 CntrDetailData Class 12](#_Toc415771148)

[3.3 General 13](#_Toc415771149)

[3.3.1 Links to new pages 13](#_Toc415771150)

[4. Other requirements common to all components 13](#_Toc415771151)

[4.1 Safety Requirements 13](#_Toc415771152)

[4.2 Software Quality Attributes 13](#_Toc415771153)

[4.3 Business Rules 14](#_Toc415771154)

[4.4 Server Database Requirements 14](#_Toc415771155)

[4.5 Product Operation Requirements 14](#_Toc415771156)

[4.6 Product Revision Requirements 14](#_Toc415771157)

[4.7 Deployment Requirements 15](#_Toc415771158)

[5. Appendix Notes: 15](#_Toc415771159)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Henry Landry | 3/29/15 | Initial version | 1.0.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to describe the architecture and component design of the Unloaded Shipping Container Display application extension of Acme’s WMS system. Continuing the design of the subsystem in a methodical and detailed manner will yield a high quality product.

## Intended Audience

This document is intended for designers/programmers of the WMS system that will implement this subsystem. The document describes in some technical detail the overall architecture of the system and relationships between the subsystems. Similarly, it also describes how the individual components should work internally and how they communicate with each other.

## Product Scope

The Unloaded Shipping Container Display application extension is a set of two webpages and supporting pieces that extend Acme’s existing internal WMS application. These new pages provide operators and managers easily accessible information about shipping container contents as they were unloaded and palletized for storage in the warehouse. As a note, reference number is really just an identifier on a set of pallets stacked and wrapped at the same time from the same instance of a source of goods. Reference number is further qualified with one of several possible values of a field called reference type. These reference types tell us the type of the source of the goods on the pallet. Therefore as a byproduct of this effort and just a little more work users will be able to look at data about other product besides shipping containers. One example is product that was palletized as a result of manufacturing. Most of the requirements and some of the specifications are written using shipping container as the reference type because that is what was requested. It also makes the document easier to read. At key points in documents there will be reference to displaying other reference types.

## References

This document references and/or should be used in conjunction with the following documents:

* Use case and activities diagrams document for this project. They will help provide clarity to this document. U28475331\_CEN4031\_RequrementsUseCase.docx
* The software requirements specifications document for the project will also help provide clarity. Landry-U28475331-CEN4031-SRS.docx .
* Acme Custom Programming Guide which is conventions and standards for all development.
* Various standards and conventions documents Acme’s IT has assembled about the WMS application web pages and WCF service methods
* Tools – Visual Studio, Internet Information Service, ASP.NET and C#

## *Reminders*:

* Throughout the requirements document and this document the data to display is described as shipping containers. This is the reference type requested by the stakeholders. However, shipping containers are just one reference type.
* The detail data for these webpages is usually palletization data describing how the goods were palletized. Sometimes the more generic term “Cntr” is used in the WMS system because technically these records can be applied to bins, tubs or other ways to store goods. Cntr and shipping container should not be confused, they are not the same.

# System Architecture

## Layer/Call Architecture Diagram

**WMSActivityByRefNbr** Page

**WMSCntrDetailsByRefNbr** Page

**GetCntrData**

**ByReferenceAndDate**

Exposed Web Service Method

**GetCntrData**

**ByReferenceNbr**

Exposed Web Service Method

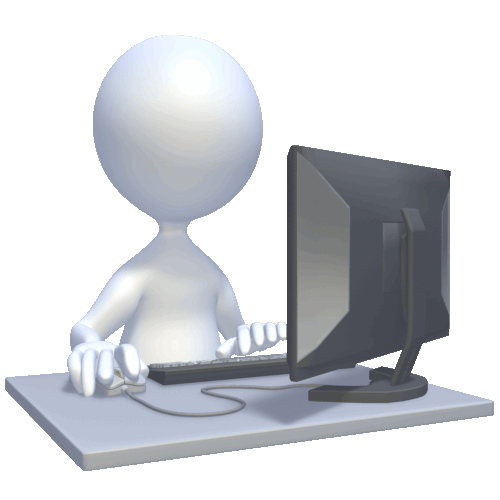
**shipmgmt**

Data Model/Database

**SummarizeCntrData**

**ByRefNbr**

Internal Web Service Method



WCF Net.TCP

HTTP

WCF Net.TCP

Internal Method

Internal Method

EDM Database Queries

**UI Layer**

**Business Layer (WMS Service)**

**Data Layer**

System Desktop user

* The UI layer of the Unloaded Shipping Container Display system is composed of two web pages. Users start at the WMSActivityByRefNbr.aspx page to see all of the unloaded containers for a time period they choose from a limited set of 1, 2, or 3 weeks back. They may click on one of the containers in the list to move to the WMSCntrDetailsByRefNbr.aspx page and then see how the goods from the container were palletized. Each page calls a corresponding WMS web service method to gather their data based on criteria selected by the user.
* In turn, the two exposed service methods in the diagram call the shipment EDM data layer to retrieve database data. The GetCntrDataByReferenceAndDate is for getting lists of reference numbers with summary data while the GetCntrDataByReferenceNBr is for getting one reference number with detailed data. These methods do some processing and organizing and also employ the helper internal service method to gather additional detail data and summarize the additional data. Using the internal helper method design pattern ensures the data is consistent plus code reuse savings is achieved.

## Architecture Details of the Layers

* Web pages for the UI layer
  + The webpages are the main target of this project because they provide the information requested by the stakeholders. Both of the webpages are written with ASP.NET and will be hosted on existing IIS servers at each installation of the WMS system. Per Acme standards server side processing and client side processing will be used for the best user experience and ability to use standard patterns available.
  + Web service calls made by the web pages use WCF Service Method technology and net.tcp protocols for efficiency.
* Web service methods for the business layer
  + This subsystem adds three methods to the existing WMS web service as they are named in the architecture diagram. Two methods are exposed and are called by respective web pages. A third is a helper method and used by both of the exposed methods for centralized processing and reuse of code. Of course all the methods are designed so they may be used in the future by other pages and clients.
  + The web services will return instantiations of two new classes which provides at least the extract data for these web pages. In fact the new classes contain more data than required in case other callers of the web service might desire to use them. The class CntrSummaryDataByReferenceType is the main class that is returned by both of the new methods and a List<> of CntrDetailData can be included with the main class, if required by the caller, in the CntrSummaryDataByReferenceType object. The caller would only request the detail data if needed for display. In this case the WMSActivityByRefNbr.aspx page does not require details but the WMSCntrDetailsByRefNbr.aspx page does.
* Data Layer
  + This subsystem is a reporting tool, therefore no data is written to the database.
  + As is typical for the WMS system data is stored in a database represented by the shipmgmt Entity Data Model. Also as is typical for the WMS system, queries are performed using LINQ to SQL.
  + The tables required for these web pages are S1301\_Cntr which represents pallets or more generally containers of any type and S1302\_CntrDetails which represents the contents of the container S1301.

## Design Rational

This project is an add-on to an existing system. The benefit and trade-offs of this situation is that most of the architecture patterns are already in place and should be duplicated for consistency and ease of maintenance. One major deviation from the current pattern is in the web page architecture. There is an initiative at Acme to add more client side processing to web pages to keep in step with recent paradigms and improve the user experience of course. The WMSActivityByRefNbr page is to be a model for this initiative. More is discussed in the component design of the page.

## Assumptions

This document assumes the reader is familiar with Acme’s WMS system, especially the parts of the system that are used for unloading shipping containers and/or creating S1301\_Cntr records. The plan also assumes a heathy and installed WMS system exists to add this subsystem to.

# Component Design

This section details the components of the system. The new web pages are object oriented classes of their own by the nature of ASP.NET framework. The primary data components of the system are the new data classes returned by the service methods as detailed below and the existing entity data objects of the data model. As is typical in WCF web services, the web service methods are just new methods in a larger class, the web service itself. For clarity, the web service methods will be represented as classes for this document.

*Note: the design specification points each reference the requirements specification(s) they implement. The engineer should reference the SRS document to confirm they are implementing the requirement completely.*

## UI Layer

### WMSActivityByRefNbr.aspx Web page

Wireframe diagram of the page for reference (UCP-01, UCP-03**)**

|  |
| --- |
| Shipping Container data for site TN since 02/02/2015 |
| Since 2/2/15 Since 1/26/15 Since 1/19/15 Shipping Container PO ID  Refresh |
| Container Type Date Time Date Time User Item Description Good Damaged Expected Good Pallets  Number Began Began Ended Ended ID Qty Qty Qty Built |
| OSCL Container 2/2/15 10:10 2/2/15 12:15 sam 123456 Fan 2200 15 2215 50 |
| … |
| … |
| Footer (system diagnostic data) |

***Content Design of WMSActivityByRefNbr****.****aspx***

* + This page uses the site.master asp.net master page as its base with the title in the Title Content placeholder and the rest of the content in the Main Content placeholder. The master page will take care of UCP-07 by providing a footer. SQIP-2 also applies.
  + There is considerable interaction between the various controls on this page as the user works with them. Therefore the page should be heavily invested in client side JavaScript and JQuery rather than send trips back to the server for server side processing where not necessary. **(**SQIP-1**)**
  + Requirements UCP-03, UCP-04, UCP05, UCP-08 dictate the interaction two radio sets and the Refresh button. The content text of the radio sets can be calculated when the server side page is rendered, making 7 days back into date, for example. However, clicks on these radio sets cause the Refresh button to change text and color to warn the user the current selections of the radio sets do not match the data contents of the screen. JavaScript and CSS classes should be used to make this interaction smooth and easy to change colors.
  + Global variables and functions should be added as necessary to event handling and values of the radio sets in order to promote code reuse.
  + Requirement UCP-06 dictates a data grid structure or HTML table. This can also be implemented smoothly with a JQuery click handler on the Refresh button. The click event would call JavaScript methods to retrieve the data from a WebMethod contained on the same web page class and also create the data table.
  + This new client side function GetData() can use the JQuery ajax method to retrieve JSON list of data from the WMSActivityByRefNbr.aspx page’s WebMethod GetReferenceList(). This implements UCP-09 and SQT-2. The same client function can apply error data if any is returned by the WebMethod. The error information can be placed in a textblock control in the data area instead of a grid structure. (UCP-12)
  + Client side function BuildDetailDataTable(data) creates the grid structure and fits the data to the structure including appropriate CSS styles on each element. It also must add a click event on rows in the table so the user can click on a row and drill into palletization details.
  + The click event on the grid structure implements UCP-10. It formats a query string with container type, days back and the container number then passes that to WMSCntrDetailsByRefNbr.aspx. The query string data is also used by the child page to bring the same criteria back if the user clicks return on that page.
  + The WMSActivityByRefNbr.aspx page takes optional query string data for reference type and days back (these are defined below). When present the values in the query string should be used to set days back and reference type. If the query string variables are not present ShippingCntr and 7 days back are the default values (UCP-11)
  + Various helper methods for converting data such as dates, times and reference types to user friendly screen values should be included.
  + Styles (UCP-02)
    - The page references a WMSActivityByRefNbr.css stylesheet to provide the attributes of assigned styles. This sheet should inherit from MetricsPages.css to get common styles for the website.
    - Styles must be defined for these controls so their behavior and appearance are uniform plus can be changed easily:
      * Controls in the Radio sets
      * Refresh button
      * Header row of the grid structure
      * Data rows of the grid structure
      * Overrides of the site-wide styles when necessary.
      * Different CSS styles should exist to handle the control of the refresh button changes (text and color) such that the JavaScript applies different styles as the user clicks rather than directly altering the attributes of the controls.
  + The web page does not require paging as per UCP-14.
  + The performance requirements of the page are in (SQE-1)
  + Browser compatibility requirements are in WCR-01 and WCR-02
* ***Server side processing (ASP.NET)***
  + The page inherits from WMSBasePage.cs to provide the usual functionality for WMS web pages. It should perform all the usual processing required of an WMS web page including registering with device state, logging, SetupPage() overrides, registering web service clients, and having a page ID assigned. Automatic page refreshing should be turned off. SQT-1 through SQT-5.
  + Query string variables “reftype” and “timeperiod” should be tested for null and allowed values then converted to proper data as .Net property types. They are used to optionally populate the radio controls, usually if the user is returning from the detail page.
  + The server side code can implement the values and display text of the radio controls as they are always value 7, 14, 21 days back from today and the text would be the actual calendar date in short date format. Default value is 7.
  + The reference type radio set can be hardcoded in aspx with the 3 valid screen values of Shipping Container, PO and ID. The corresponding database values of these are shipcntr, po and id which are defined as an enum in the S1012\_CntrRefType object. The page should provide static converter methods to work between the two sets of values.(UCP-06)
  + The primary server side data processing will be contained in a static WebMethod contained in the WMSActivityByRefNbr code behind called **GetReferenceList** that is called from the client side using ajax. This method takes time period and reference type from the client call. It calls the WMSWMSWebservice GetCntrDataByReferenceAndDate method to get data from the database. The web service returns a List<CntrSummaryDataByReferenceType> which can be directly returned to the client for display.
  + The CntrSummaryDataByReferenceType class is to represent a shipping container (or other reference type) and its corresponding pallets. It is discussed in more detail below. (UCP-09)
  + The GetReferenceList web method should catch any exceptions from the web service and log them as is usual for the project. It should also format the exceptions into user friendly messages contained in a SoapException to be thrown back to the client. (UCP-12)

### WMSCntrDetailsByRefNbr.aspx Web page

#### Wireframe diagram of the page (PDP-01, PDP-03, PDP-10)

|  |
| --- |
| Palletization Data for Container OSCL123453 |
| Return  Refresh |
| Pallet ID User Location Bin Number Status Date Created Time Created Item Description Quantity |
| 0460131234456 sam 1 1 Verified 2/2/15 10:20 123456 Fan 55 |
| … |
| … |
| Footer (system data) |

#### Content Design of WMSCntrDetailsByRefNbr.aspx

* This page has far less interaction of controls by the user so it can be implemented almost entirely on server side with traditional ASP.NET methods instead of JavaScript.
* The page should use the site.master master page as its base with the title in the Title Content placeholder and the rest of the content in the Main Content placeholder. The master page will take care of PDP-01, PDP-05 and also SQIP-2 by providing a footer.
* PDP-03 means the title section should be dynamically created on the server based on the query string variables passed in. If the query string variables are in database format converter methods on the WMSActivityByRefNbr page can be used to convert to display values
* Requirement PDP-04 dictates a data grid structure or HTML table. It can be designed and populated on the server side using traditional a Gridview control and LoadGridData method. The grid is automatically populated using data generated with query string variables (PDP-06).
* Various helper methods for converting data such as dates and times to appropriate screen values should be included.
* Styles (PDP-02)
  + If possible the page can reference a WMSActivityByRefNbr.css stylesheet to provide the attributes of assigned styles and stay consistent with its master page. If a different style sheet is implemented it should inherit from MetricsPages.css to get common styles for the website.
  + Styles must be defined for these controls so their behavior and appearance are uniform plus can be changed easily.
    - Button area
    - Header row of the grid structure
    - Data rows of the grid structure
    - Overrides of the site-wide styles when necessary.
* The performance requirements of the page are in SQE-2.
* Browser compatibility requirements are in WCR-01 and WCR-02.

***Server side processing (ASP.NET)***

* The page inherits from WMSBasePage.cs to provide the usual functionality for WMS web pages. It should perform all the usual processing required of an WMS web page including registering with device state, logging, SetupPage() overrides, registering web service clients, and having a page ID assigned. SQT-1 through SQT-5.
* Query string variables “reftype” and “RefNbr” should be tested for null and allowed values then converted to proper data as .Net property types. See the error processing section below for invalid values (PDP-08). The query string variable timeperiod is also accepted and used only to return to the master page.
* The GridView control is populated in the LoadGridData method. This calls the WMSWebservice GetCntrDataByReferenceNbr method to get data from the database. The web service returns a single CntrSummaryDataByReferenceType which contains a List<CntrDetailData> for the pallets on the container. This List is bound to the GridView. The CntrSummaryDataByReferenceType class is to represent a shipping container (or other reference type) and its corresponding pallets and is discussed in more detail below. (PDP-07)
* The click event on the return button implements PDP-10. It formats a query string with reftype and timeperiod variables then passes that to WMSActivityByRefNbr.aspx.
* The click event on the refresh button just reposts the same criteria to the server side page in case data has changed, such as a container currently being unloaded.
* In case of web service failure, see the error processing section below.
* Error processing: in event of errors that prevent loading data such as invalid query string or web service failure the GridView should be hidden and a label should display a user friendly error in place. The buttons may remain, especially to return to the master page. PDP-08, SQT-2

## Business Layer

*<<Service Method>>*

|  |
| --- |
| GetCntrDataByReferenceAndDate |
| See below for input parameters |
| Web page calls this to get a list of all reference nbr for criteria |

<<Service Method>>

|  |
| --- |
| SummarizeCntrDataByRefNbr |
| See below for input parameters |
| Helper method to do the summarizing |

<<Interface>>

|  |
| --- |
| CntrSummaryDataByReferenceType |
| No Cntr Details |

Summarized data about the all matching containers

*<<Service Method>>*

|  |
| --- |
| GetCntrDataByReferenceNbr |
| See below for input parameters |
| Web page calls this to get a list of details for a since reference nbr |

<<Interface>>

|  |
| --- |
| CntrSummaryDataByReferenceType |
| Cntr Details Included |
| Summarized data about a single selected containers, with details |

Ship-mgnt

EDM

Web Page

Web Page

Summarize the Pallets

Summarize the Pallets

Query to get list of reference nbrs

Query to get single reference nbr

### Service Method Relationship Diagram

### GetCntrDataByReferenceAndDate Web Service Method

##### Purpose

This exposed method returns a list of reference numbers with summary data about the pallets built with each number. As an example it would return a list of all received shipping containers received between specified dates and say how many pallets were built in total along with other summary data. The summary is contained in a list of CntrSummaryDataByRefenceType classes. See the design of that class below for details of what data is summarized.

* The common web service method requirements for this project are SQE-3, SQE-4

##### Input Parameters

* + refType – (String) the reference type to query with, usually “shipcntr” but could be “id” or “po”. Technically it must be the string form of one of the values of S1012\_CntrRefType.RefTypes.
  + startdate – (String) inclusive start date to query for.
  + enddate – (String) inclusive end date to query for. It should be greater than the start date.
  + IncldDetails – (Bool) include the CntrDetailData list in the return or not. These are sizable datasets so it can be a useful performance increase to leave these out if not needed
* Return
  + A list of SummarizeCntrDataByRefNbr objects unless there were no matches, then it would be an empty list. These represent totals of the pallets built from the source reference number and type.
* Logic
  + The method should first validate the parameters and return an exception if any are invalid.
  + Then it queries the database for all pallets (S1301\_Cntr objects with related S1302\_CntrDetails records) matching the requested criteria, grouped by reference number.
  + Each group of same-reference-number records is passed to the SummarizeCntrDataByRefNbr method which totals and otherwise summarizes the data, returning a single CntrSummaryDataByReferenceType class which also contains the list of CntrDetailData (the detailed pallets).
    - Per the setting of the IncldDetails parameter the list of CntrDetailData may be dropped or saved after it is returned from SummarizeCntrDataByRefNbr
  + The returned CntrSummaryDataByReferenceType class is added to a running list which will be returned to the caller
  + Failures in the query, the summarizing method or adding to the list to be returned should results in a FaultException<WMSException> with logging and a user friendly message. (SQRE-1, SQRE-2)
* Exceptions
  + Exceptions will be of the usual WCF FaultException<WMSException> with a user friendly message contained.
  + Any exceptions thrown within the method should be caught and logged per the usual WMS standards.
* Performance requirements for the web page using this method are in SQE-1.

### GetCntrDataByReferenceNbr Web Service Method

##### Purpose

This exposed method returns a single CntrSummaryDataByRefenceType which represents a summary of the pallets built under the reference number and type. The object also contains a detailed list of the pallets built. See sections about the CntrDetailData and CntrSummaryDataByRefenceType for the data that is contained.

##### Common web service method requirements for this project are SQE-3, SQE-4

##### Input Parameters

* + Refnbr – (String) the reference number to query with.
  + refType – (RefTypes enum) the reference type to query with, one of the values of S1012\_CntrRefType.RefTypes.
  + startdate – (String) inclusive start date to query with. This is not technically a criteria input but more of a safety criteria since reference numbers may be reused. This can be used by the caller to prevent relatively old data from being returned.
* Returns
  + A single SummarizeCntrDataByRefNbr object unless no matches were found then it would a null object. These represent totals of the pallets built from the source reference number and type. This object will always contain a loaded list CntrDetailData objects.
* Logic
  + The method should first validate the parameters and return an exception if any are invalid.
  + Then it queries the database for all pallets (S1301\_Cntr objects with related S1302\_CntrDetails records) matching the requested criteria.
  + The S1301\_Cntr dataset records are passed to the SummarizeCntrDataByRefNbr method which totals and otherwise summarizes the data, returning a single CntrSummaryDataByReferenceType class which also contains the list of CntrDetailData (the detailed pallets).
  + The returned CntrSummaryDataByReferenceType class is returned to the caller
  + Failures in the query, the summarizing method or adding to the list to be returned should results in a FaultException<WMSException> with logging and a user friendly message. (SQRE-1, SQRE-2)
* Exceptions
  + Exceptions will be of the usual WCF FaultException<WMSException> with a user friendly message contained. (SQRE-1, SQRE-2)
* Performance requirements for the web page using this method are in SQE-2.

### SummarizeCntrDataByReferenceNbr Web Service Method

##### Purpose

The purpose of this method is to do the real work of summarizing the List of Cntr objects passed in. The summarized data is compiled into a SummarizeCntrDataByRefNbr object. This object also contains a list of CntrDetailData objects which is a list of records corresponding to the input Cntr objects. See the sections on each of these classes in this document for the exact data that is contained.

##### Parameters

* + Refnbr, the reference number that is being summarized
  + refType – (RefTypes enum) the reference type of the number being summarized, one of the values of S1012\_CntrRefType.RefTypes.
  + List of S1301\_Cntr objects with their related S1302\_CntrDetail records attached
* Logic
  + The method should first validate the parameters and return an exception if there are any invalid.
  + It must retrieve the damage location number for the current site from the appropriate WMS parameter method. In event of failure the exception must be logged and a default location “99” used. The method does not fail because of this.
  + If an expected quantity of the reference number applies to the reference type the method should find the service’s ERP service client and attempt to find the expected quantity using the appropriate method for that type. Shipcntr, po and lot are reference types that might have expected quantities. The expected quantity is a total for the reference number, not by pallet. The method should not fail because of an error finding expected quantity though a value of ‘error’ should be entered as the expected quantity.
  + The method should loop though all of the S1301 records with a twofold purpose. First it will be building the summary SummarizeCntrDataByRefNbr object to be returned plus it will be building a CntrDetailData object for each S1301/S1302 object. There can be more than one CntrDetailData if there is more than one S1302 record for an S1301.
    - While building the SummarizeCntrDataByRefNbr object the method will sum appropriate data such as quantity or taking the earliest or latest for data such as dates, etc. The details on how to handle each field in the object is in the section for that object.
  + Errors calculating the objects to be returned should result in a complete failure with a log entry and FaultException<WMSException>. It should contain a user friendly message. (SQRE-1, SQRE-2)
* Returns
  + A single CntrSummaryDataByReferenceType summarizing the S1301 records passed in
* Exceptions
  + Exceptions will be of the usual WCF FaultException<WMSException> with a user friendly message contained. (SQRE-1, SQRE-2)

### CntrSummaryDataByReferenceType Class

Purpose

This class is used to return data from WMS Web Service methods to clients that call appropriate methods. Basically is a summary of a set of S1301\_Cntr/S1302\_CntrDetails records related by reference number. Data in the set is summed or the latest/earliest value is picked to represent the entire set of records. See field details below on how to summarize. In addition, one property of the class is a list of CntrDetailData objects which represent each S1302\_CntrDetail record in the set. The class would be defined in existing file IWMSService.DataTypes.

* Note: The S1302\_CntrDetails objects passed in must also have their S0501\_Items record attached in order to determine item number and description.
* Constructor
  + Only one base constructor is included which takes reference number and reference type. Data is not enforced for the class otherwise because it is a simple translation from database records.
* Field and descriptions
  + RefNbr – This String field represents the reference number the dataset is built on.
  + RefType – This String field represents the reference type the dataset is built on.
  + RefVerified – This Boolean field clarifies whether the reference number was verified to be valid at the time the pallet was built.
  + Quantity – This Decimal field is the total quantity across all if the records in the dataset.
  + NumTagsUnDamaged – this integer field is the number of S1301\_Cntr records that are not located in the damaged location.
  + NumTagsVerified – this integer field is the number of S!301\_Cntr records that were applied and verified to be on a pallet.
  + CreateDate – the DateTime the reference number was started, which technically is the timestamp of earliest S1301 record.
  + EndDate – the DateTime the reference number was completed, which technically is the timestamp of latest S1301 record.
  + Cntrs – this is a List<CntrDetailData> of the data making up the summary. See the section for CntrDetailData class.
  + UserID – this String is the UserID that created the first S1301 record
  + GoodQty – A decimal number representing the sum of all quantities in the dataset of all the S1301 records not located in damage location.
  + DamagedQty -- A decimal number representing the sum of all quantities in the dataset of all the S1301 records not located in damage location.
  + Expectedty – The Decimal total quantity of all items expected for this reference number.
  + DamageLoc – The string value of the damage location for this site, it is a single value not related to the dataset. It is the damage location used to determine if quanties and tags were damaged or not.
  + ItemNbr – If the entire dataset contains only a single item then this field will contain that item number. If the dataset contains more than one item this field will be empty string.
  + ItemDesc – If the entire dataset contains only a single item then this field will contain that item’s description. If the dataset contains more than one item this field will be the word “Multiple”.

### CntrDetailData Class

* Purpose

This class is used to return data from WMS Web Service methods to clients that call appropriate methods. At a high level is a different representation of S1301\_Cntr/S1302\_CntrDetails records that is more useful on web pages that focus on reference numbers. In fact these records can be thought of as unique Cntr-Items since there can be multiple records if it is a multi-item pallet. See field details below, each field has an exact corresponding field in S1301\_Cntr or S1302\_CntrDetail or S0501\_Item. The class would be defined in existing file IWMSService.DataTypes.

* Constructor
  + Only one base constructor is included which takes Cntr ID, Reference Number and Reference Type. Data is not enforced for the class otherwise because it is a simple translation from database records.
* Field and descriptions
  + CntrNbr – This String field represents the Cntr ID or pallet ID of the record
  + RefNbr – This String field represents the reference number the cntr was associated with when it was built.
  + RefType – This String field represents the reference type the cntr was associated with when it was built.
  + ItemNbr – String item number contained in the S1302 record.
  + ItemDesc – String description in the database for the item number of the S1302 record.
  + QtyOnPallet – This Decimal field is the quantity of this item on this Cntr.
  + CreateDate – the DateTime the Cnt was created
  + UserID – this String is the UserID that created the Cntr
  + CurrLocation – this string is the current location ID the Cntr is located in. Note this may not be the location the Cntr was built in.
  + CurrBin – this string is the current location ID the Cntr is located in. Note this may not be the location the Cntr was built in.
  + CntrStatus – A string representing the current status of the Cntr
  + Verified – Boolean variable indicating if the Cntr was verified or not

## General

### Links to new pages

Per requirement UCP-14, a link to WMSActivityByRefNbr.aspx should be added to the menu website of web pages for all Acme locations.

# Other requirements common to all components

## Safety Requirements

* + **SQU-1:** User documentation will be provided in a manner consistent to other modules of the WMS system. A white paper will explain how to use the pages within Acme’s environment. The white paper will be posted along with similar white papers for other parts of the system.

## Software Quality Attributes

* + **SQP-1:** The new software modules will follow standards and conventions in the Acme Custom Programming Guide except as noted below.
  + **SQP-2**: The new software modules will follow standards and conventions in the various specifications documents about the WMS system except as noted below. Coding and user experience will be consistent with the rest of the system.
  + **SQP-3:** The page will not time out or auto refresh as is common in other parts of the system. This is an exception to SQP-2
  + **WCR-02**: It is not the goal of the project use techniques that limit the web pages to IE; however in order to save time this is allowed with permission from the Application Development Manager.
  + **SQT-1**: Serious errors that deny the user data or might provide bad data to the user must fail with a message to the user in the data grid as per SQRE-1 and SQRE-2. The server side web page operation must log the exception to the user’s device log file. The web service method must log its exception to the windows event log appropriate to the web service with an Error designation. The logging is details in the specifications for the WMS system.

## Business Rules

* + **SQIP-1:** Web pages are written in ASP.NET with JavaScript (JQuery preferred) client side scripting when appropriate.
  + **SQIP-2:** Web pages are children of the WMS system’s site.master page which controls headers, footers and page-wide global settings.
  + **SQIP-3:** Web page formatting is based on CSS style sheet files. If necessary each new page can have its own style sheet but the sheet must inherit from the SiteWebPages.css style sheet.

## Server Database Requirements

* **SDR-01**: The web service method shall collect data using the shipmgmt entities data model which connects to the appropriate database.
* **SDR-02**: The webpages are presentation layer only and do not access the database directly. Only the WMS web service shall access the database on behalf of the web pages.
* **SDR**-**03**: The database is the standard SQL Server already included with the WMS product. No additions or changes necessary since these pages display existing data.

## Product Operation Requirements

Correctness

* + **SQC-1:** The project shall include unit tests and system tests detailed in another document.

## Product Revision Requirements

Maintainability

* + **SQM-1:**  The software shall be stored in the appropriate source control system with check outs required to make changes.
  + **SQM-2**: All supporting documents shall be stored with Acme’s other WMS documentation on the project’s website.

Testability

* + **SQT-1:** The software shall implement diagnostic logging per usual Acme WMS standards as below:
  + **SQT-2**: Serious errors that deny the user data or might provide bad data to the user must fail with a message to the user in the data grid. The server side web page operation must log the exception to the user’s log file. The web service method must log its exception to the windows event log appropriate to the web service with an Error designation.
  + **SQT-3:** Less serious errors that do not invalidate user data can assume appropriate default actions and continue as long as the problem does not affect data accuracy or the user is informed. The situation still needs to be logged as in SQT-2 with a warning designation. The action taken should be noted in the log.
  + **SQT-4:** The web pages will make the usual Device State entries that all workflows and web pages in the WMS system does. This operation is handled automatically by inheriting the new pages from the system’s base page.
  + **SQT-5:** Testing is to be performed first on the system’s development server. With approval by management it will be promoted to the system’s test server and tested again. When this stage is approved the software may be deployed by itself or in conjunction with other features and fixes as a schedule determined by IT and each warehouse.

## Deployment Requirements

**SD-1**: The product will be deployed by Acme IT personnel using the usual methods of WMS upgrades and installation which are to hand copy required files into the production system on a scheduled developed with users of the system (if there will be a system interruption). Automated methods of deployment are on the backlog but have never made it to top of the priority list.

# Appendix Notes:

* As a nonfunctional requirement, these new pages were intended to be a complete model for using CSS tables and JQuery client side scripting in future system web pages. Time limitations may restrict the completeness of the model. The pages should still function as planned, although the underlying technique may be traditional APS.NET server side scripting.