0S000442

COMBAT RATION NETWORK

FOR

TECHNOLOGY IMPLEMENTATION

Quality Data Management System

Final Technical Report STP 1001

Results and Accomplishments (Nov 1996 - June 2000)

Report No: FTR 107

CDRL Sequence: A004

August 2000

CORANET CONTRACT NO. SP0103-96-D-0016

Sponsored by:
DEFENSE LOGISTICS AGENCY
8725 John J. Kingman Rd.
Fort Belvoir, VA 22060-6221

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

Principal Investigator:
Mr. Henderikus B. Bruins
Dr. Nabil Adam

Dr. John F. Coburn
Program Director

TEL: 908-445-6132
FAX: 908-445-6145

*A New Jersey Commission on Science and Technology Center

20001017 022

Read and Write

SCIENTIFIC AND TECHNICAL INFORMATION SYSTEM

1. Acc. Number DS000442	1A. Act. Code AQPO	2. Trans Type M Modify A Record	3. Effort Status C Completed	4. Perf. Meth. C Contract

Performance Type 5. R RDT&E Work Unit	6. Date of Summary 2000/09/25	7. Prec. Sum Date 1997/12/29	8. Start Date 1996/11/20	9. End Date 1998/01/01	Continuous End ☐

10. Sec. Class. U Unclassified	18. Distribution Code A Distribution Unlimited	19. Distribution Reason PB Public Release

20. Title
STP *1001, QUALITY DATA MANAGEMENT SYSTEM*

22. Linking Accession Number (Select Up to Two) DS 00441	23. Local Control Number *96003*	24. Search Data

25. Subject Categories

Code	Description
0608	**Food, Food Service and Nutrition**
1308	**Manufacturing and Industrial Engineering and Control of Production Systems**

26.1 Mission Area

Code	Description
5.1	**Logistics**

26.2 Function Code

Code	Description
7.2	**Manufacturing Technology**

26.3 Technology Code

Code	Description
4.4.2	**4.4.2 Computer-Aided Manufacturing, Inspection and Testing Technology**

27. Resp. Org. Source 432549	27.1 Responsible Organization Name **DEFENSE LOGISTICS AGENCY FORT BELVOIR VA**

27.2 Responsible Organization Component Name
TECHNICAL ENTERPRISE DIVISION

27.4 Responsible Individual

Last Name	First Name	MI
EGGERS	**RUSSELL**	**K**

27.5 Resp. Indiv. Off. Sys.	27.6 Resp. Indiv. Phone	27.7 Resp. Indiv. DSN No.
DLA-J-339	**703-767-1417**	**427-1417**

28. Perf. Org. Source	28.1 Performing Organization Name
410970	**RUTGERS - THE STATE UNIV PISCATAWAY NJ**

28.2 Performing Organization Component Name

CENTER FOR ADVANCED FOOD TECHNOLOGY

28.4 Performing Individual

Last Name	First Name	MI
COBURN	**JOHN**	**F**

28.5 Perf. Indiv. Off. Sys.	28.6 Perf. Indiv. Phone	28.7 Perf. Indiv. DSN No.
CORANET	**908-445-6130**	

28.8 Associate Investigator Names (Last, First, MI)   *ADAM, NABIL*
BRUINS, HENDERIKUS B.

35. Keyword Text

RATIONS
THERMOSTABILIZED
PROCESS
QUALITY
CONTROL
CIM ARCHITECTURE
MACHINE VISION
ROBOTICS
DUAL USE
FLEXIBILITY
INTEGRATION
DATA BASE
SHOP FLOOR
SHOP FLOOR DATA
USDA
FDA
VETCOM
ORACLE
DSCP-Subsistence
WEB

36.1 Objective
TO SET UP A SYSTEM FOR COLLECTION OF CURRENT DATA ON PRODUCT MANUFACTURED BY THE COMBAT RATION INDUSTRIAL BASE, INCLUDING QUALITY AND QUANTITY, FOR PURPOSES OF IMPROVING INVENTORY MANAGEMENT AND THE ABILITY TO DO RECALLS BY DSCP.

37.1 Approach
THE NEEDS OF DSCP WERE DEFINED AND DESCRIBED IN TERMS OF THE KIND OF DATA TO BE COLLECTED, THEN COMBINED WITH INPUT FROM USDA AND THE PRODUCERS AT EACH PLANT AND FROM THE VETCOM AT ASSEMBLY PLANTS; MODELED INTO AN ORACLE DATA BASE FOR PROMPT MANAGEMENT ACCESS, ANALYSIS AND DECISION MAKING. A SPECIAL SERVER AT DSCP WILL BE THE REPOSITORY FOR ALL OF THIS QUALITY DATA, PLUS MORE PRODUCTION INFORMATION TO BE ADDED IN THE FUTURE.

38.1 Progress

AT THIS TIME THE SOFTWARE IS IN PLACE, TESTED ON A SERVER AT DSCP, AND INSTALLED AT THE SERVER AT DSCP. EACH OF THE PRODUCERS HAS BEEN INTRODUCED AND TRAINED, AND DIFFICULTIES ARE BEING OVERCOME. THERE WAS A SIX-MONTH PERIOD OF ON-LINE SUPPORT, AND THE RESPONSIBILITY OF SUPPORT IS NOW TRANSITIONED TO DSCP-SUBSISTENCE AND THE LOCAL PC SUPPORT GROUP

30. Primary PE No.	31. 1st Contrib. PE No. 0708011S	32. 2nd Contrib PE No.
30A. Primary Proj. No.	31A. 1st Contrib. Proj. No. 96003	32A. 2nd Contrib. Proj. No.
30B. Primary Task No.	31B. 1st Contrib. Task No. STP #1001	32B. 2nd Contrib. Task No.

30. Primary Funding Data			31. 1st Cont. Funding Data			32. 2nd Cont. Funding Data			33. Contract Rollup Indicator
Fiscal Year	Dollars	Work Year	Fiscal Year	Dollars	Work Year	Fiscal Year	Dollars	Work Year	
C1	C2	C3	C1	C2	C3	C1	C2	C3	C
1996	34745	1							
D1	D2	D3	D1	D2	D3	D1	D2	D3	D
1995 7	14480	01							N
E1	E2	E3	E1	E2	E3	E1	E2	E3	E
1998	7898	1							
F1	F2	F3	F1	F2	F3	F1	F2	F3	F
2000	8777	1							
G1	G2	G3	G1	G2	G3	G1	G2	G3	G

39A. Product Set No.	39.1A Prod. Title Class Code	39.2A Product Title
1	Unclassified	

39.3A Product ID/Report No	39.4A Product AD No	39.5A Product Indicator

39B. Product Set No.	39.1B Prod. Title Class Code	39.2B Product Title
2	Unclassified	

39.3B Product ID/Report No	39.4B Product AD No	39.5B Product Indicator

39C. Product Set No.	39.1C Prod. Title Class Code	39.2C Product Title
3	Unclassified	

39.3C Product ID/Report No	39.4C Product AD No	39.5C Product Indicator

39D. Product Set No.	39.1D Prod. Title Class Code	39.2D Product Title	
4	Unclassified		
39.3D Product ID/Report No		39.4D Product AD No	39.5D Product Indicator

39E. Product Set No.	39.1E Prod. Title Class Code	39.2E Product Title	
5	Unclassified		
39.3E Product ID/Report No		39.4E Product AD No	39.5E Product Indicator

34. Contract/Grant/Trans. No.	34.1 Contract Effective Date	
34.2 Contract Expiration Date	34.3 Contract Face Value	34.4 Contract Cum. to Date
SP010396D1604	1996/11/20	
2000/06/30	659000	659000

40. Dom. Tech Trans.	HI High potential for civilian application
41. Study Category	6 Managment
42. Spec. Study Sub	OR Systems analysis, operations research analysis, policy analysis
44. Prim. Proj. Serial No	
45. Int. Sources Con.	N
46. Processing Date	
47. Receipt Date	

49. Thrust Indicator

Technology for Affordability

Focal Point
Russell Eggers

Author
Mark Glover

Status Code

DS000442

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 8/7/00	3. REPORT TYPE AND DATES COVERED Nov.96–June 2000

4. TITLE AND SUBTITLE Quality Data Management System	5. FUNDING NUMBERS SPO103-96-D-0016 PE - 7811s PR - 88003 96003

6. AUTHOR(S) Henderikus B. Bruins	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Rutgers, The State University of New Jersey The Center for Advanced Food Technology Cook College, NJ Agricultural Experiment Station New Brunswick, NJ 08903	8. PERFORMING ORGANIZATION REPORT NUMBER FTR-107

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Logistics Agency 8725 John, J. Kingman Rd. Ft. Belvoir, VA 22060-6221	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The QDMS is a database application that stores pertinent inspection information from producers, assemblers and government inspection agencies. The application includes analysis tools that allow the user to analyze the data for trends in defects and defective lots as function of time, product and/or producer.

QDMS is based on an Oracle relational database running on a Windows NT platform. The application was built using Oracle Developer Rapid Application Development tools to give users a highly visual interface to the database. The application is deployed in three-tier configurations where the front end tier is the client PC with a web browser, the middle tier an application server and the back end tier the database server. The user accesses the data via his desktop browser. To roll out a Web application, the user is given the application's URL. This distribution method reduces the time, cost, and complexity of deploying applications to a large or geographically-dispersed end user base, all without installing application software on their desktop machines. The only client-side requirement is a Java-enabled Web browser. This greatly reduces processor and memory requirements for end users' desktop machines.

14. SUBJECT TERMS			15. NUMBER OF PAGES 147
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

Abstract

For years DSCP's Operational Ration Business Unit lived with a problem of how to join multiple data bases at various locations into a centralized system that could permit quality specialists at our headquarters to view and archive quality records as they were produced. This was necessary since assembled operational rations such as the Meal, Ready-to-Eat were produced and inspected at various locations around the country before being shipped to military units or into long term storage. Since several million cases are procured per year at over $70 per case, any undetected quality problems could have major implications. The solution was the Quality Data Management System (QDMS).

The QDMS is a database application that stores pertinent inspection information from producers, assemblers and government inspection agencies. The application includes analysis tools that allow the user to analyze the data for trends in defects and defective lots as function of time, product and/or producer.

QDMS is based on an Oracle relational database running on a Windows NT platform. The application was built using Oracle Developer Rapid Application Development tools to give users a highly visual interface to the database. The application is deployed in three-tier configurations where the front end tier is the client PC with a web browser, the middle tier an application server and the back end tier the database server. The user accesses the data via his desktop browser. To roll out a Web application, the user is given the application's URL. This distribution method reduces the time, cost, and complexity of deploying applications to a large or geographically-dispersed end user base, all without installing application software on their desktop machines. The only client-side requirement is a Java-enabled Web browser. This greatly reduces processor and memory requirements for end users' desktop machines.

Besides the client graphical interfaces, QDMS has also several custom built interfaces that obtain automatically data from the USDA inspectors, the Army Veterinary Inspectors and Producer database systems. Synchronization with these database systems is performed at night.

To facilitate rapid response to client based analysis queries, a data warehouse system was developed and deployed. This data warehouse anticipates requests for data analysis and summarizes this data at night when the use of the system is minimal. The system is then able to generate a rapid response during daytime to users need for data analysis.

QDMS was developed as a result of a joint effort between the government, academia and industry. The team consisted of personnel from Defense Supply Center Philadelphia (DSCP), Rutgers University and MFG Systems Corp (a small business). The business concept was developed by DSCP and given to Rutgers University to develop the necessary software. It quickly became apparent that Rutgers would need assistance so they brought in MFG Systems Corp., which specializes in converting data records to Oracle for small companies. The QDMS now links six contractors and three government agencies with a common quality database which will identify negative quality trends early and reduce the cost of rejections and reworks for both government and industry.

2

Table of Content

1 Results and Accomplishments

1.1 Introduction and Background

This report describes short term project work (STP#1001) under the CORANET Contract SP0103-96-D-0016. The objective of this STP was to develop and implement a Quality Data Management System (QDMS) at DSCP.

For years DSCP's Operational Ration Business Unit lived with a problem of how to join multiple data bases at various locations into a centralized system that could permit quality specialists at our headquarters to view and archive quality records as they were produced. This was necessary since assembled operational rations such as the Meal, Ready-to-Eat were produced and inspected at various locations around the country before being shipped to military units or into long term storage. Since several million cases are procured per year at over $70 per case, any undetected quality problems could have major implications. The solution was a centralized database system that would receive, store and analyze inspection information from all organizations involved in the inspection of combat rations: the Quality Data Management System (QDMS).

The Quality Data Management System (QDMS) is based on an Oracle relational database management system that stores pertinent inspection information from producers, assemblers and government inspection agencies. Very application modules were developed that interact with this database system. The main module is the web module which enables the user to securely access the database from anywhere in the world using a standard web browser technology. This web module is by the producers to enter production and inspection information, as well as by DSCP personnel to analyze the data and generate summary reports on production and inspection activities. Three other modules were developed to facilitate automatic data transfers between data base systems from the USDA, the AVI and producers.

The design of the QDMS application includes all aspects of the food production system ranging from lot and production information to end-item inspection and shipment information. The design is such that it is not limited to combat rations but can easily be applied to other food products. Due to the open design of an Oracle database, integration with other existing or to be developed database systems within an organization is easy accomplished.

The core of the QDMS application are the analysis tools that allow the user to analyze the data for trends in rework instances, in-process losses, defects and defective lots as function of time, product and/or producer. No experience in SQL (Standard Query language) is required to conduct such analysis or is it necessary to understand the relational aspects of various tables. User friendly forms allow the inexperienced user to use the application and conduct analysis.

1.2 Results and Conclusions

QDMS is based on an Oracle relational database running on a Windows NT platform. The application was built using Oracle Developer Rapid Application Development tools to give users a highly visual interface to the database. The application is deployed in three-tier configurations where the front end tier, or user interface, is the client PC with a web browser, the middle tier an application server and the back end tier the database server. To roll out this Web application, the user is simply given the application's URL. This distribution method reduces the time, cost, and complexity of deploying and upgrading the application to a large or geographically-dispersed end user base, all without installing application software on their desktop machines. The only client-side requirement is a Java-enabled Web browser. The middle tier will run on a powerful Server and executes most of the application code on behalf of the client. This greatly reduces processor and memory requirements for end users' desktop machines.

4

Besides the client graphical interfaces, QDMS has also several custom built interfaces that obtain automatically data from the USDA inspectors, the Army Veterinary Inspectors and Producer database systems. This required standardization in data format among these various systems so that product lot and associate production and inspection information could be joined in a relational database system. Synchronization with these various database systems is performed as batch jobs at night.

To facilitate rapid response to client based analysis queries, a data warehouse system was developed and deployed. This data warehouse anticipates requests for data analysis and summarizes this data at night when the use of the system is minimal. The system is then able to generate a rapid response during daytime to users need for data analysis.

1.3 Recommendations

QDMS is based on an Oracle Relational Database Management System. The capacity of this database system goes well beyond the needs that are placed on it by the inspection data. Also, the design of the data base was done in such a way that it can easily be expanded in functionality by either joining it with other database system or taking over some tasks from existing legacy systems.

An example of such expansion in functionality could be a material tracking system for inventories of Government Furnished Materials (GFM). Producers would utilize QDMS' shipment information data entry module to record the receipt of GFM and record in the production module which of these materials were used to create assemblies. DSCP would have instant visibility of current inventory levels and be able to see in which assemblies contain GFM products.

2 Program Management

The original project was awarded under SP0103-96-D-0016 on November 20, 1996 in the amount of $347,450. The proposal for this contract was based on Technical and Cost Proposals submitted by Rutgers University dated 20-August 1996 and a SOW dated 24 July 1995. Estimated completion date of the STP was January 1, 1998. Several modifications were issued to add tasks to the project and to extend the time frame for existing tasks due to delays or other reasons

Modification # 1: February 3, 1998. This modification extended the delivery order from January 1, 1998 to 31 March 1998 at no cost to the Government. This extension was required to complete the implementation of the software application at DSCP. The delay of implementation was caused by a delay in hardware and software acquisition by DSCP to support the application.

Modification # 2: 12 February 1998. This modification extended the contract till August 31, 1998 and added $144,800 to the delivery order for additional work against the QDMS project as described in the SOW December 14, 1997. This supplemental work was recorded as STP#1001B-"Quality Data Management System - Automated Data Input" This activity developed an additional module that could be used by the producers to submit their production and inspection information via a file transfer rather than manual data entry via the web forms module.

Modification # 3: 21 September 1998. This modification extended the performance period of STP#1001B from 31 August 1998 through 31 December 1998 at no additional cost, due to start up delays.

Modification # 4: December 23, 1998. The delivery order was extended from 31 December 1998 through 16 February 1999. This extension allowed time needed to develop a SOW to provide on-site training and to provide interim assistance to DSCP personnel in daily operations and maintenance of QDMS

Modification # 5: April 14, 1999. This modification added additional tasks and extended the performance period to December 31, 1999 for continued work against the QDMS project as described in Statement of Work dated March 23, 1999. The supplemental work is annotated as STP#1001-C -QDMS Maintenance,

Technical Support and Training during Transition. This modification added $78,980 to the delivery order, increasing the value of the contract from $492,250 to $571,230.

Modification # 6: December 30, 1999. The delivery order was extended from 31 December 1999 to 31 January 2000 at no cost to the government. This extension was needed to "buy" some time to formulate a new SOW for training of DSIO personnel and to upgrade the QDMS application to the latest Oracle tools.

Modification # 7: January 6, 2000. The performance period was extended until 30 June 2000 for continuation of QDMS maintenance, technical support and training. Additional funds in the amount of $10,000 were obligated, increasing the total value of the delivery order from $571,230 to $581,230

Modification # 8: March 17, 2000. Additional funds were committed to the project in the amount of $77,770, increasing the total value of the project to $659,000. The incremental funding were based on a Technical and Cost proposal titled "QDMS-Knowledge Transfer" submitted on February 28, 2000

3 Short Term Project Activities

3.1 STP#1001: Quality Data Management System

The Development and Implementation of QDMS was a joint effort between Rutgers University and MFG Systems Corporation. A sub contract (#832) between Rutgers and MFG was fully signed on 1/17/97 and was amended several times to reflect the extensions and modification of the main contract.

3.1.1 System Analysis

Several meetings were held with DSCP personnel and Industry representatives to conduct a in-depth software and system requirement specification. The input obtained from these meetings was used to formulate a systems requirement of the QDMS application. Based on this system requirement, a software system development test and evaluation plan was developed and issued as requested to the goverment.

In consultation with DSCP, Windows NT-4 platform was selected as the preferred hardware platform on which the application would be developed. The application would store it's data in an Oracle Relational Database and the interface with this database would be based on Oracle Developer and Designer 2000 Tools. This approach allowed for the deployment of the application either as a two tier system (client/server) for intranet use and/or as a three tier system via a web server for both intranet and internet use. The three tier deployment was at that time at an infancy stage, but promised to offer several advantages over a two tier deployment. For example, the user would not need to install special client run time tools on its machine but could instead use its own web browser and would be platform independent. Also, all data processing would be done on the middle tier (web server) thus cutting down on network traffic and increasing the overall performance of the system, especially in an Internet deployment configuration.

Also, in consultation with DSCP several pre-defined analysis were agreed upon:
- Defective Lot Analysis
- Defect Analysis
- In-Process Defects Analysis
- Variable Data Analysis
- Rework Instant Analysis

All analysis would display their data as function of producer, product or monthly time increments and would give the analyst the ability to fine tune the queried data based on a set of input parameters.
As it was anticipated that large amount of inspection information would be used to conduct the above analysis, and it was decided to implement a data warehouse type architecture on top of the production tables in which summary data would be stored for analysis purpose. These tables would be updated during the off hours with new and modified data that was collected during the day time. This architecture enhanced the response time to the user dramatically.

Data normalization was a serious concern during the analysis phase. Various agencies had their own "normalized" way to identify products, producers and defects. During the analysis phase various data inconsistencies were identified and recommendations were made to normalize these data elements among the various users such as the National Stock Number (NSN) for product identification. Similar, data elements such as contract numbers, defect id's, unit of measures, were normalized among the various agencies and producers that submitted data. Due to the extensive effort that was spend to normalize all these critical data fields, internal validation on these fields was built into QDMS before committing the data to the database. This would assure early detection in the use of non-normalized data.

3.1.2 USDA Data

Meetings were held with the USDA starting January 10, 1997 to discuss the integration of the USDA Quality Data base with QDMS. The USDA field inspectors collect their data into Paradox databases which export their data on a weekly basis to their head quarters in Washington. It was agreed that the USDA would create a custom file for QDMS purposes which would contain first inspection information. These files would be formatted according to an agreed upon specification and posted on their Bulletin board. QDMS would dial into this bulletin board on a nightly basis and retrieve these files for import into the QDMS data base. Validation of the data would be performed after the import in staging tables. Records that could not be validated would be moved to error tables for review by DSCP system analysts and records that could be validated would be added to the production tables. A special feature was built into this validation system that checked if a producing company was required to enter its own lot information into the system. The inspection header information from the USDA was used to create a lot in the system for those companies that were not required to enter data into QDMS. In all other cases, the USDA inspection header record needed to be linked to an existing lot record created by the producing company.

3.1.3 AVI Data

The AVI is the second government inspection agency that accepts products on behalf of the government at the assembly plants. The first meeting with AVI to discuss the integration with QDMS was held on February 11 &12, 1997. The AVI was in midst of switching their inspection record filing system to Lotus Notes. Each inspector in the field was going to have a local Lotus Notes Database which would replicate itself to their head quarters on a regular basis. In designing the architecture for the QDMS interface, it was decided that the Lotus Notes database from the head quarters would replicate the inspection information to a Lotus Notes database at DSCP. A data pump would then be used to replicate the data from this database to staging tables in the Oracle database. The design and implementation of this system was undertaken by the AVI and their sub-contractor, while the processing of the records from the staging tables to the production tables was done under this contract.
QDMS relies on the AVI system to validate the data before submitting it to the staging tables. For this purpose various validation tables were made available to the AVI. Also a specification on required inspection header and detail data fields and their format was issued to the AVI.
In the final implementation, lot records are made available to the AVI Lotus Notes Database. The AVI system will then check if it has new or updated inspection records for these lots and make this inspection information available in the Oracle staging table. During a nightly batch job, Oracle looks at the records in the staging tables and loads any new or changed inspection records into the production tables

3.1.4 Producer Data

The interface for the producers was based on Web Forms. This approach would simplify the implementation requirements for producers as they did not have to built their own database system and would greatly enhance QDMS ability to validate the data as it is entered. This approach of deployment would also enable us to display to the producer data that was entered, either via the forms, graphs or reports. By employing user specific database views we would be able to protect the producer data. During the deployment of QDMS, the reporting capability of QDMS was disabled for producers due to security concerns (report data could not be encrypted)

The producers did request the ability to submit their data electronically to prevent dual data entries in both their own system as well as the QDMS system. As this was beyond the original scope of the project, a

follow up project was submitted and approved which would establish standards for such electronic data submission.

3.1.5 Data Warehouse

In a relational database, one tries to normalize its design to avoid duplication of data fields and thus reducing the size of the database. Conduction a data analysis of data in such relational database requires however that several tables are to be joined. The more tables to be joined the slower, the response time of the query is. To improve the response time of a query, data can be stored in a de-normalized fashion thus reducing the number of tables to be joined. To take advantage of both the normalized relational database and fast response time to data analysis queries. it was decided to implement specific data warehouse tables which would contain de-normalized data that would contain just the data needed by the various pre-determined analysis tools. This data would be summarized on daily, monthly and yearly time increments. Therefore if the user was interested in all data for a specific month, it would not have to calculate this based on each individual record for that month, but would be able to go directly to the table which contains the summarized monthly data.

The data warehouse tables are being updated on a nightly basis and available to the user/analyst the following day. This design greatly improved the response time of the system.

3.1.6 Data Analysis

DSCP requested a total of five basic analysis tools. Each tool needed to display and calculate summary data for each producer, product and time increment.

Production Defect Analysis, is the analysis that determines the percent defective units that is removed from the original production lot. Inspections that lead to such removal are: pre retort inspection, post retort inspection, rework inspection and assembly inspection operations.

Lot Acceptance Analysis is the analysis that determines the ratio of lot inspection failures over inspections performed. Inspections that can lead to lot inspection failures are end item and receipt inspections

Inspection Defect Analysis is the analysis of defects found during the lot inspection operations (end item and receipt inspection). It calculates the estimated quantity of defects in the product based on the lot size inspected, the sample size taken and the number of defects found, and it displays the ratio of these estimated defects over total quantity inspected.

Variable Data Analysis is the analysis that calculates the average value of the variable and the average of the coefficient of variation of the variable data.

Rework Instance Analysis is the analysis that calculates the ratio of rework instances over lots produced. This calculation makes a distinction between lots that were offered and were not offered to the government. A lot is deemed "offered to the government" once an inspection record has been received from either the USDA or the AVI

3.1.7 Test and Validation

Test and validation of the various modules was done in accordance to the submitted test and evaluation plan. Testing on module level were performed as each module was completed. Systems testing were performed after all modules were completed and the final integration test was performed at DSCP after the deployment of the application was completed.

3.1.8 Implementation

The QDMS application was installed at DSCP in March 1998. Training in the use of the system was given to DSCP personnel and manuals provided to system administrators (TWP#205), DSCP analysts (TWP#203) and Producers (TWP#204)

3.1.9 User Manuals

The following manuals were prepared in the course of this project:
QDMS Software and System Description (restricted distribution)
QDMS Operators Manual (restricted distribution)
QDMS Database Administrator Manual (restricted distribution)
Specification QDMS File Based Data Transfer (TWP#202)
Quality Data Management System: User manual for DSCP Analyst (TWP#203)
Quality Data Management System: User manual for Producer Data Entry Clerk and Producer Analyst
(TWP#204)
Quality Data Management System: User manual for Administration and Maintenance (TWP#205)
QDMS File Transfer Utility User Manual (TWP#206)

3.1.10 Operational Support

Operational support was given to DSCP after completion of the implementation and continued through
subsequent project extensions till June 30, 2000 after which operational support was taken over by
DSCP/DSIO. Extensive support was given to DSCP during the September 1998 - January 1999 time frame.
During this period the QDMS system was relocated to the DSCP N-E Philadelphia location. Significant
problems were encountered, due to network changes and firewall implementation issues. Changes were
made to the system to allow access through the firewalls and calls from outside users were routed via a
different Internet Service Provider. During the time frame January-June, 2000 significant resources were
expended to train DSCP/DSIO personnel in the maintenance and deployment of the qdms application,
including the upgrade of the application to the newest Oracle software development and deployment tools.

3.2 STP#1001-B: -Quality Data Management System - Automated Data Input

3.2.1 System Analysis

The objective of this phase of the project was to develop one or more modules that would receive and
import data from producers via electronic file transfer. Various methods for data transfer were evaluated.
Data security and data integrity was a major factor in final recommendation. A questionnaire was sent to
the various producers on Feb 2, 1997 to survey each companies capability to electronically extract and
submit inspection data. It was determined that the industry was not using any standard software package to
store their production and inspection information. It was concluded that it would be best to develop a
standard for electronic record submission and that it would be the responsibility of each industry member to
develop a custom interface in order to submit their data in this standard format. On May 6, 1998, we
submitted to each producer for comments a draft specification, describing the proposed functionality of the
automated data input modules and the file/record format to be followed. No objections to the proposed
specifications were received and final recommendation to the Coranet JSG representatives were made on
August 10, 1998. On 28 August 1998 approval to proceed as recommended was obtained

3.2.2 Specifications

A document with final specification for file based transfer system was written and issued as TWP#202 and
send to all producers in electronic format. The system would consist of three major modules. The first
module, based on a web cartridge, would authenticate the user and receives the file(s). The second module
would process the received files and validate the records before adding them to the database in a manar
similar to the validation done in the web deployment. The third module generates error reports that are
automatically e-mailed to the producer.

3.2.3 Development of Software Modules

The various software modules were developed by MFG Systems using the Coranet Demo facilities
hardware and software.
The module that receives the files is a web cartridge that interacts with the client. First the client is being
requested to identify himself using the Oracle login id, password and connect string. Next it request the

user to submit a file. The file is then transferred to DSCP over the Internet using Secured Socket Layer for data encryption. The name of the transmitted file is then appended with the users login id and sequential number to avoid duplicate file name problems and to accommodate record validation against the user who submitted the records.

The second module runs at set intervals and processes the recent received files. First, each record is checked for errors. Records without errors are placed in the production tables and records with errors are placed in error table. The validation routines are equivalent to the validations performed in the web module.

The third modules runs also at set intervals and checks the error tables and sends to each vendor via e-mail an error report with then errors found since the last e-mail transmission. This module requires that a mapi compliant e-mail client is running on the qdms server

The development of the modules was completed by end of November 1998 after which the test and validation activities started

3.2.4 Test and Validation

To test and validate the functionality of the electronic data transfer modules, an Access database was developed that could generate the required files with records according to the specification listed in TWP#202). The design of the Access database application was later made available to both SOPAKCO and Wornick to assist them in the design of their relational database system

To test the validation and error trapping routines, records were on purpose altered to make records none compliant with the specifications. All tests and validation activities were successfully completed in January 1999

3.2.5 Implementation

The implementation of the automated file transfer module was completed in February 1999. System tests were performed to ensure full functionality.

3.2.6 User Manuals

A user manual, which includes the file specification, was written and electronically sent to all producers. This manual was later also issued as technical working papers TWP#206

3.3 STP#1001-C: QDMS Maintenance, Technical Support and Training

During this phase of the project, Rutgers and it's sub contractor maintained the QDMS system at DSCP with remote access software and when required via on-site support.

3.3.1 Maintenance and Performance Assurance

The System was maintained from January 1, 1999 till June 30, 2000
The maintenance tasks were:
❑   Maintain and adjust QDMS database tables
❑   Maintain and create users
❑   Maintain web server
❑   Maintain data loads
❑   Backup database and file system
❑   Track performance of qdms and consult/advice users regarding short comings

3.3.2 System Personnel Training

Training of DSCP/DSIO personnel was not completed during this phase of the project nor was the maintenance responsibilities handed over to them as originally anticipated. Reason for this was two fold. The maintenance of the Oracle based system requires personnel that are trained in the use of these tools. DSIO did not have these skills directly available and personnel were required to attend courses in these subjects. The second reason was the end of the millennium and the concerns of Y2K compliance. Significant resources at DSCP/DSIO needed to be dedicated to this effort and could not be made available

to take over the responsibility of the QDMS application. For the above reasons, training's activities for the DBA and application developers and maintenance personnel were delayed till January 2000 (STP#1001-D)

Training was however provided to the QDMS application administrator: Michael Pelligrino, who was trained in maintaining QDMS application from a users perspective such as maintaining setup information: product listings, defect assignments, contract information, etc as well in the aspects of data checking of USDA data imports. In addition Michael was trained in the use of the analysis and reporting tools. He then assisted the various DSCP analysts in the use these tools.

To assist DSIO and DSCP in maintaining the application from a systems administrators perspective, a document was prepared and send to DSCP that describes the functionality of the various elements of qdms: " QDMS Software and System Description". This document describes in detail the functionality of all the modules.

3.3.3 On-call support

This task was original planned for the last four month of 1999. However, due to reasons explained above, Rutgers and MFG, remained the primary organization to maintain QDMS system and application till the end of 1999.

3.4 STP#1001-D: QDMS-Knowledge Transfer

The technology transfer during phase C was not sufficient effective to support a final transition of the system and application to DSIO. Therefore phase "D" was added to this project which would primarily focus on the training of DSIO personnel while slowly transferring the responsibilities to DSIO. As the application was originally built in 1997, the application language used by Oracle was upgraded as well as the web server technology. To maintain an up-to-date application, it was decided to upgrade the QDMS application during this phase by using the latest Oracle Tools

3.4.1 Administration and Coordination

In coordination with DSCP POC meetings and demonstrations were organized and attended to ensure that all aspects of this project were properly executed and completed in a timely manner.

3.4.2 Technical Monthly Meetings

Technical meetings were attended as needed to review technical training's aspects and progress of the project as the responsibilities for maintenance of the QDMS application and its underlying system were slowly transferred to DSIO.

3.4.3 Maintenance and Technical Support

Maintenance and technical support, initially performed by Rutgers University and it's sub-contractor, were gradually transferred to DSCP/DSIO as various training's courses were completed. Rutgers University and it's sub-contractor. By the middle of May, DSIO took on the responsibility to monitor and maintain the system and Rutgers responsibilities were reduced to back-up support

3.4.4 Training

The main emphasis of this project phase was to train DSIO personnel in the various aspects of the QDMS system which included an Oracle data base, a web server, an application server and the actual application One training session was held at DSCP:
❑   January 28, 2000, DBA training in database and web server maintenance and monitoring
Four training's sessions were held at Rutgers:
❑   April 13, 2000, DBA training in database security
❑   April 18, 2000, Data Load training for the USDA, AVI and Producer data
❑   May 9, 2000. Developer Application Code training of the various forms, reports and graphs
❑   June 14, 2000, QDMS application administrator training

11

3.4.5 Upgrades/Modifications/Deployment

Application code was upgraded to Oracle Database version 8.15, Developer 6.0, Oracle Application Server 4.08, including the cartridge used for file based data transfer. Perl scripts were not converted to NT scripts at the request of DSIO

Installation of the upgrades on the test server ran into delays due to hardware problems on the test server. A new PC was configured and tools were successfully installed on this machine. During the interim a server was also configured at the Rutgers FMT facility to act as the test and validation platform of the upgraded components.

Once the new test server at DSIO was configured the application files were installed. The operational functionality of this server was tested for forms, graphs and reports.

On June 5, 2000 the production server was shut down and the database moved from the production server to the test server. Users were given instructions on how to login this server and how to install the new version of Jinitiator, a java add-in that runs on the client.

The production server was then rebuilt by DSCP/DSIO and the new Oracle tools installed, configured and tested.

3.4.6 Documentation

On May 16, 2000, a meeting was held with DSIO to cross reference the existing documents to the standard documents required by DSIO. Electronic copies of all documents were submitted

4 Appendix

    A. QDMS: Software and System Description
    B. QDMS: User Manual for Administration and Maintenance
    C. QDMS: File Transfer Utility User Manual
    D. QDMS: System Diagram

Appendix A
QDMS Software and System Description

Quality Database Management System

QDMS SOFTWARE and SYSTEM DESCRIPTION

Sponsored by:
Defense Logistics Agency

Contractors:

Rutgers, The StateUniversity of New Jersey
Food Manufacturing Facility
120 New England Avenue
Piscataway

MFG Systems Corporation
100 Davidson Avenue
Somerset, NJ

August 2000

1 GENERAL

1.1 Purpose of the Software and System Description

This software and system description for short term project # 1001: "Quality Data Management System" is written to provide:
- A description of the software modules that are being used by QDMS
- A description of the system
- A basis for the maintenance of the system and software.

1.2 Project References

A software requirement description was formulated in a statement of work issued on July 24, 1996 as part of the Combat Ration Network for Technology Implementation (CORANET) program. In response, a technical and cost proposal was issues on August 20, 1996, describing in more detail the various tasks that will be performed to fulfill the software requirements described in the statement of work. This proposal was approved on 20 November, 1996.

Under a previous contract with Rutgers, a prototype system was developed and installed on a database server at the Rutgers Food Manufacturing Technology Facility (FMTF) in Piscataway, NJ. Remote access was provided to DPSC in Philadelphia to allow initial trials and to uncover opportunities for improvement. This prototype system used an Oracle 7 database system to manage inspection data from the USDA as collected by their Operational Ration Database (ORDB) system.

The final technical report (Report No: CRAMTD STP #60 - FTR20.0) describes in detail the database model, the data entities and the functionality of the front end application. This system was used as the foundation to which new data sources were added and analysis modules that were custom tailored to DSCP's requirements.

The following two diagrams depict the logical flow of combat rations through the manufacturing and assembly system. Information that is collected from the various inspection stages can be stored in the QDMS

DPSC MRE PROCESS FLOW
Retorter..USDA

Bad Pouch (Rework)

NO (Rework)

Rework/Scrap/Reclassify

| Producer Pack (Retorter) | → | 100% Pre-Retort Screening by Producer (Data Collected) | —Good Pouch→ | Retort | → | 100% Post-Retort Screening by Producer and Cartoned (Data Collected) |

Pouch Fail Screening

| Scrap | | Reclassify for other use. |

| Sample Inspected by Vendor (Data Collected) | → | Was Inspection Passed? | —YES→ | Offered to USDA | → | Sample Inspected by USDA (Data Collected) | → | Was Inspection Passed? | —Yes→ | Shipped to Assembler (Data Collected) |

Good Pouch

Figure: Flow of Product at the Retorter

DPSC MRE PROCESS FLOW
Assembler..AVI

No (Rework)

| Assembler Receipt Inspection Performed (Data Collected) | → | Was Receipt Inspection Passed? | —YES→ | 100% Pre-Assembly Screening (Data Collected) | → | Assembly | → | Assembler End Item Sample Inspection (Data Collected) | → | Was Inspection Passed? |

Scrap

Yes (Ok for Assembly) Return to stage at which inspection was performed.

| AVI Warranty Inspection (Data Collected) | → | Was Inspection Passed? |

NO

| Scrap/ Reclassify or Return to vendor |

Yes

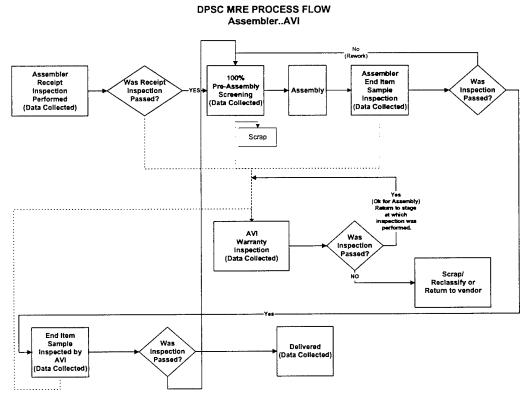| End Item Sample Inspected by AVI (Data Collected) | → | Was Inspection Passed? | → | Delivered (Data Collected) |

Figure: Flow of Product at the Assembler

5

1.3   Terms and Abbreviations

- AVI   Army Veterinarian Inspection Agency
- CORANET   Combat Ration Network for Technology Implementation
- CRAMTD   Combat Ration Advanced Manufacturing Technology Demonstration
- DPSC   Defense Personnel Support Center
- DSCP   Defense Support Center Philadelphia
- FMTF   Food Manufacturing Technology Facility
- INTEGRATOR   Programming Module in QDMS that adds records to the Data Warehouse
- LOV   List of Values
- MONITOR   Programming Module in the QDMS that monitors the entry, editing or deletion of records
- ORDB   Operational Ration Database
- PL/SQL   Programming Language/Sequential Query Language
- QDMS   Quality Database Management System
- STP   Short Term Project
- SQL   Sequential Query Language
- USDA   United States Department of Agriculture

2 SYSTEM SUMMARY

2.1 Background

Work under the Combat Ration Advanced Manufacturing Technology Demonstration Contract (CRAMTD) has resulted in a computer integrated manufacturing (CIM) system for combat rations. Included in the CIM system is a management system to identify material vendors, associating them with purchases, and tracking their relative performance over time in terms of applicable criteria. Initially the Vendor Evaluation System (VES) was developed with an architecture to meet the needs that an individual enterprise, such as a combat ration producer, would be facing.

That VES for a purchasing office was expanded to help DSCP manage Quality History Records, for evaluating vendors, and for tracking product information. Previously, no system existed which permits the integration of the various quality databases which are maintained on operational rations. These databases include contractor origin and assembly records. USDA origin records, and Veterinary records. Massive amounts of various computer and paper records are actually collected and stored manually at DSCP, but are very difficult to retrieve in a timely and useful fashion for effective decision making.

Under a previous contract with Rutgers (Short Term Project #60), a prototype computer-based VES application was developed and installed on the database server at the Rutgers Food Manufacturing Technology (FMT) Facility in Piscataway, NJ. The prototype Database Management System was a limited system that was developed as a stepping stone to test system functionality and performance, but was limited to handling only one data source. Remote access was provided to DPSC in Philadelphia to allow initial trials and to uncover opportunities for improvement. The prototype application used an Oracle 7 database system to manage inspection data from the USDA as collected by their Operational Ration Database (ORDB) system.

The current software development effort was performed as a Short Term Project (STP) under the CORANET Contract, SPO103-96-D-0015. The objective of this project was to complete the development of a Data Management System that contains Quality data from the USDA, AVI and Combat Ration Producers, and can serve as a management decision making tool for DPSC-HR quality and management personnel. This development activity was completed in February of 1998 at which time the software was installed on an NT server at DSCP. The AVI installed in April their Lotus Notes software on a second server near the QDMS server and installed also a data pump which replicated inspection records to Oracle staging tables. The system was validated in April and user accounts established.

Training was given to DSCP personnel in May 1998 in the use of QDMS and user manuals were made available. At the same time, instructions were provided to the producers and assemblers of MRE rations with instruction on how to access QDMS with their web browser.

After the initial deployment of QDMS, our efforts concentrated on file based data transfer from the producers/assemblers to QDMS. This method would enhance data entry into QDMS for those companies who had already a database application which contained the essential inspection data. This data can then be exported into comma delimited files and send to QDMS. This system was developed while maintaining the same security and validation as designed into the system via the web form based data entry.

2.2 Objectives

1) To implement a computer integrated database (QDMS) to track information pertaining to quality performance of vendors, routinely sent in by the various data generators (producing plants, USDA inspectors, Army Veterinarian inspectors, etc.)., to DPSC-HR quality and management people.
2) To provide expanded analysis and presentation capabilities in a manner useful as a management decision making tool, to help management select preferred vendors while assuring control over product quality, lead time and costs.
The system consist of the following functional areas:
Importing records from the USDA ORDB database
Importing records from the AVI Inspection database

7

Importing records from Producers and Assemblers
Storing the records in a relational database
Implementing Data Warehouse tables to improve the response time of the system
Analyzing, Displaying and Reporting the data in a manner useful as a management decision making tool

2.3 System Definition

See section 3.3.1 for a detailed graphical display of the system

2.4 System Diagrams

See section 3.3.1 for a detailed graphical display of the system

2.5 Computer Program Identification

File Transfer USDA
Data Import USDA
Record Transfer AVI
Data Import AVI
Data Normalization USDA
Data Warehouse Monitor
Data Warehouse Integrator
Dynamic Web Page Generator

2.6 Assumptions and Constraints

The following assumptions were made:

DSCP:

hardware and software supplied and available as specified in section 3.1 and 3.2 at time
of integration
technical resources made available to identify:
required data from producers and assemblers
analytical data requirements
network issues, protocols, firewalls, etc.

USDA:

implementing a bulletin board that supports Zmodem protocol for data file transfers
implementing requested upgrades of database to prevent erroneous data entry
make bulletin board available for down loads

AVI:

install Lotus Notes database at DSCP for record transfers to QDMS
setup Lotus Notes data pump to transfer records to QDMS staging tables
implementing requested upgrades of database to prevent erroneous data entry
adhere to data format storage with will comply with QDMS requirements

Producers:

DSCP assistance in identifying data type/format to be stored
record data in format that is universal among the industry
can obtain access to Internet

Assemblers:

DPSC assistance in identifying data type/format to be stored
record data in format that is universal among the industry
can obtain access to Internet

The capability of the system will be constraint by:

The system will only be useful if the producers and assemblers upload their data on a regular
basis.
The system will only be useful when setup information such as material descriptions, contract
description. defect descriptions are supplied and maintained on a timely basis.
The system can only be implemented and maintained if the required hardware and software is

obtained, installed and maintained on a timely basis.
If material tracking needs to be accomplished from processors to assemblers and from assemblers to government warehouses then a reliable and unique lot coding scheme needs to be put in place and used by all organizations.
The system will only be reliable if the following support is given:

person assigned to maintain database and backups
person assigned with Oracle skills who can perform regular maintenance, users and user access
person assigned who can maintain the File Server and Oracle Web Server on the network

3 ENVIRONMENT

3.1 Equipment Environment

The hardware requirements are
Database Server Options:
   a) A Pentium Pro 200 INTEL processor machine with a capacity for additional processors.
   b) Operating System: Windows NT Server
   c) Memory: 64 MB
   d) 2 * 2 Gigabyte SCSI Hard Drives
   e) 2-4 GB tape drive backup unit.
   f) 10Mbps/100Mbps Fast Ethernet card.
   g) CD-ROM drive (SCSI)
   h) 17" Monitor
   i) High speed (Partial T1) full time Internet connection
   j) Modem: 28,800 baud

Client Options.
   a) A Pentium 100 Mhz processor machine or comparable speed machine.
   b) A 250 MB Hard Drive
   c) A network connection with access to the Server (DSCP), or
      A modem (28,800 baud) for dial up access to the server or Internet provider, depending on
      which means of access are implemented at the producers sites
      A SVGA monitor with 640x800 screen resolution

3.2 Support Software Environment

Server
   a) Windows NT Server 4.0 or higher.
   b) Oracle7.3 Workgroup Server for NT with a minimum of 20 user licenses.
   c) Oracle Developer/2000, version 1.6 for Windows NT
   d) Oracle Developer/2000 Server, version 1.6 for Windows NT.
   e) Reflection 2 Software to access and download files from USDA
   f) PC-Anywhere for remote maintenance of the system

Client
   a) Windows 95, 98 or Windows NT Operating System
   b) Web Browser Software: Netscape 4.0 or Windows Internet Explorer 4.0 or higher
   c) Jinitiator from Oracle as plug-in as Java Run Time Engine
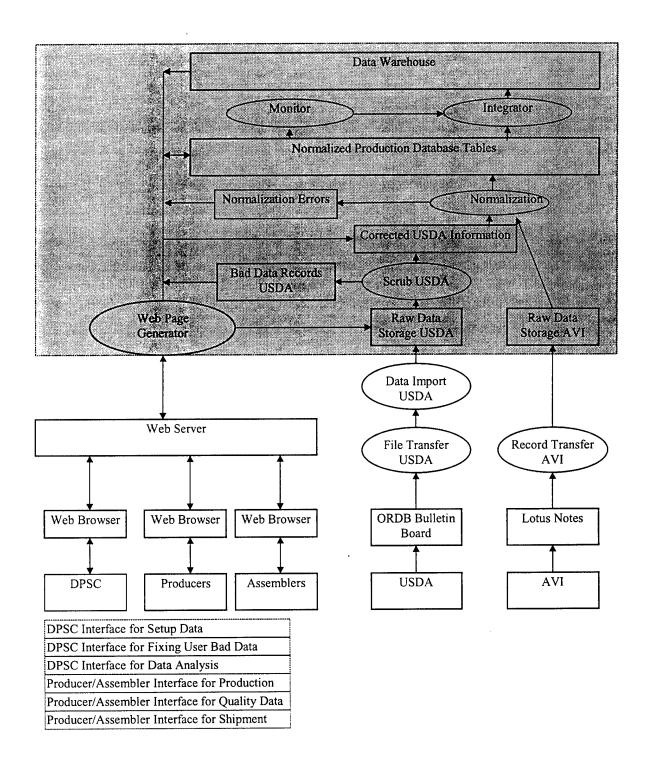
3.3 Interfaces

- All client machines (DSCP, Producers & Assemblers) must be connected to the server via a network connection that can communicate to the Web Server via TCP/IP.
- An interface to USDA in order to transfer inspection Data from their ORDB Application into the QDMS application.
- An interface to the AVI Lotus Notes Server in order to transfer inspection Data of assembled lots into the QDMS application.
- An interface between the web server and the database. This communication protocol of this interface will be a none TCP/IP protocol.

3.3.1 Interface Block Diagram

The data from the AVI and USDA is initially imported and stored in staging tables. From there procedure are run to validate the data and add it to the production tables. Data entered via the Web forms will directly

be stored in the production tables. Updated and new records in the Operational database will be propagated upwards into the Data warehouse tables. Data in the Data warehouse tables can be accessed by the end users via the Analysis modules.

Data Warehouse

Monitor — Integrator

Normalized Production Database Tables

| Normalization Errors | ← Normalization |

| Corrected USDA Information |

| Bad Data Records USDA | ← Scrub USDA |

Web Page Generator — Raw Data Storage USDA — Raw Data Storage AVI

Web Server

Data Import USDA

File Transfer USDA

Record Transfer AVI

Web Browser	Web Browser	Web Browser

| ORDB Bulletin Board | Lotus Notes |

DPSC	Producers	Assemblers

| USDA | AVI |

DPSC Interface for Setup Data
DPSC Interface for Fixing User Bad Data
DPSC Interface for Data Analysis
Producer/Assembler Interface for Production
Producer/Assembler Interface for Quality Data
Producer/Assembler Interface for Shipment

3.3.2 Interface Definition

The following interfaces with other systems are identified in the above diagram:

File Transfer and Data Upload USDA
A file transfer program was developed using Z-modem protocol to automatically upload files from the USDA bulletin board. This programming uses Reflection 2 as the communication package.

Record Transfer AVI
Lotus Notes data pump is used to pump data from the Lotus Notes database into the QDMS staging tables.

Web Server and HTML pages
The QDMS application is developed using Developer/2000 tools for the forms, graphs and reports. Oracle Developer/2000 server is used to dynamically convert these forms and graphs into Java applets and serve them to the user. Oracle/2000 server converts the report output into "pdf" files and sends them to the user. The communication between the web server and the database is a none TCP/IP protocol, while the communication protocol between the user and the web server will be TCP/IP.

3.4 Security and Privacy

1. All users are provided with a unique User-name and Password in order to gain access to the system. The web server will interpret the login and password to identify the user and use this data to restrict the data that can be accessed and viewed, thus preventing a vendor from having access to another vendor's data. Each user can be assigned to one or more user groups.

2. The communication protocol between the web server and the database is a non- TCP/IP protocol.

3. The following user groups will be created

Administrator	(full database access)
Maintenance	(access to setup tables, waivers, error log, contracting)
DSCP Analyst	(access to all analysis and reporting functions)
Producer Data Entry	(access to lot, receipt and inspection data entry forms)
Producer Analyst	(access to analysis tools, restricted records)

   Each User of the system will be assigned to one or more of these rolls.

   In addition, "VIEWS" are used to restrict data that can be accessed by the user.

   For example, the data entry clerk of company xyz has been assigned to the roll of Producer Analyst. This person will be restricted to a 'VIEW", which restricts that person to see only certain records. In case of inspection data, QDMS restricts this user to see only inspection data that either belongs to the material lot that was manufactured at that location or to inspection records that were entered into the system by this location. This security will be applied to all analysis and reports. In addition, the data entry clerk of this company has access to the form in which new records can be entered or old records edited. However, QDMS will only allow the clerk to edit records that are not older than 300 days.

   Producer Analysts are allowed to see all inspection data that belongs to a lot that was produced at their location via a detailed report or graphical analysis.

   Producer Clerks are allowed to see lot and inspection records via forms. They are allowed to edit these records up to a fixed time period after the record was generated (was 30 days, is now 300 days).

DPSC users are allowed to see unrestricted analysis and reporting data no matter who produced or who inspected the product.

The USDA and the AVI can be assigned to the DPSC user group, which will give the same privileges as DPSC, or if they are not assigned to this group, they will have the same privilege as the Producer Analyst.

Due to security concerns Producer Analyst cannot use the report function of QDMS at the current time, as report output is not encrypted.

4 DETAILED CHARACTERISTICS AND REQUIRMENTS

4.1 Specific Performance Requirements

4.1.1 Accuracy and Validity

- Any calculation is accurate to the first decimal
- No requirements are placed on this system regarding the accuracy of the entered and imported data. Error checks are done on the incoming data to make sure that it complies with the normalized format that is used in the database.
- The analysis results are accurate to the first decimal and displayed as such.

4.1.2 Timing

No timing requirements were specified by the customer. Timing will be, in part, a function of the hardware (processor speed) acquired by the customer and the traffic on the network.

4.2 Computer Program Functions

4.2.1 Scheduling and Batch Program for Importing USDA Data (USDAIMP)

4.2.1.1 Program Description

This program module is a scheduling program that executes various modules in a predefined order at specific time periods. The initial scheduling trigger for USDAIMP is initiated by "Windows NT Scheduler" (USDAIMP.job). This job will start a batch file (D:apps\qdms\perl\dtrans.pl) to download the files from USDA, load the staging tables, download the files from USDA, archive the data, and populate the production tables after validation. Once "dtrans.pl" is started it will execute the program modules in the following order: USDASEND, USDAIMPO, USDANORM. Dtrans.pl is written in "Perl", a scripting language.

4.2.1.2 Detailed Functional and Performance Requirements

- Windows NT Scheduler starts USDAIMP which executes dtrans.pl
- Dtrans.pl executes the following modules
  - execute USDASEND (Reflection script)
  - execute USDAIMPO (perl script)
  - execute USDANORM (sqlplus)
- terminate

4.2.1.3 File Transfer USDA to DPSC (USDASEND)

4.2.1.3.1 Program Description

The USDA stores updates to their Operational Ration Database (ORDB) on a bulletin board which can be accessed by means of a modem given the correct login name and password. The files are producer specific and contain updated quality records from the USDA inspection agency. Two files are specifically formatted to the needs of QDMS and contain inspection header information (*.gen) and inspection detail information (*.fai). Both files are ascii comma delimited files.
USDASEND is a Reflection-2 script (USDA.R2W), that retrieves these files from the USDA and stores them locally in a directory "download". USDASEND will automatically be started by the batch program, "dtrans.pl", written in "Perl". USDASEND will logon, identify and download any new files and automatically disconnect after the download has been completed.

4.2.1.3.2 Detailed Functional and Performance Requirements
- automatic dial-in and logon at specific time of day
- identify new files for down load
- request download
- store files on file server
- disconnect

4.2.1.3.3 Special Requirements
The above functionality requires that the USDA supports Zmodem file transfer protocol, and generates ascii files for both inspection header information (*.gen) and inspection detail information (*.fai).

4.2.1.4 USDA Data Import (USDAIMPO)

4.2.1.4.1 Program Description
The USDAIMPO module is the program module that imports the USDA records from the flat files to staging tables in the Oracle database. This program is a "Perl" script and part of the file "dtrans.pl". It will execute after the completion of USDASEND module and checks for new files that were downloaded by USDASEND. It will accomplish this by comparing the files located in subdirectory "Download" with the files in subdirectory "UsdaArchive". It will assume that a file is "new" if is located in "Download" but not located in "UsdaArchive". It will copy these "new" files to a subdirectory "UsdaWorking". Once this task is completed, it processes the files located in this working directory and import each record into an Oracle staging table: "FAILINSP" or "GENINFO". It uses SQLloader for this process. After the files have been processed, the files are moved from the "UsdaWorking" to the "UsdaArchive" directory. After all *.gen" and "*.fai" files are processed, the working directory should be empty and the USDAIMPO application will terminate.

4.2.1.4.2 Detailed Functional and Performance Requirements
- run application after USDASEND has been completed
- check usda download directory
- move new files to working directory
- import inspection header records to usda staging tables
- move inspection header files to usda archive directory
- import inspection detail records to usda staging tables
- move inspection detail files to usda archive directory
- terminate after all files are processed

4.2.1.5 Data Scrubbing and Normalization USDA (USDANORM)

4.2.1.5.1 Program Description
USDANORM module is the Data Scrubbing and Normalization procedure that is executed as a sqlplus session (cp_err.sql) and is activated by the Perl script "dtrans.pl" after completion of the USDAIMPO module. USDANORM module consist of various sql database procedures.

The first procedure ("copy_err_gen_fail") checks first if there is any data records in the USDA Error tables ("GEINFO_ERRS" and "FAIL_ERRS"). If there is an error record it will move this record into the respective staging tables unless the staging table already contains the records, at which time it will delete the record (It assumes that the new record from the USDA might have corrected an error that was found previously). It will also delete the record if the record is older than 30 days since the first time the day that it was first processed

The second procedure ("val_gen" and val_fail) will populate any key fields in the GENINFO and FAILINS tables with not null values.

16

The third procedure ("val_geninfo" and "val_failinsp") will validate the records in the GENINFO and FAILINS tables and move any records that fail to the GENINFO_ERRS and FAIL_ERRS tables. If during this validation it is found that a lot does not exist. it will check if the producing company is on-line or off-line. If the producing company is off-line, this script will generate a lot record in the PRODUCT_LOT table with the information supplied by the USDA and a few assumptions such as contract number= 999999-99-9-9999, product date = inspection date and production quantity = inspection quantity. (This contract is for a fictitious MRE ration (NSN:9999-99-999-9999), and all products listed in QDMS are made a component to this ration)

The fourth procedure in this module ("load_lot_insp_head" and "load_end_it_def") will load the records from the staging tables to the production tables LOT_INSPECTION_HEADERS and END_ITEM_DEFECTS. Before a new record is moved from the staging table to the Operational Database, the normalization procedure will check if a record with the same key fields already exist in the Operational Database. If the record already exist, the procedure will update the record. If no record could be found with the same key fields then a new record will be created in the Operational Database. New or updated records that are placed in the Operational Database will be flagged (DW flag) for the MONITOR. This flag is reset by the INTEGRATOR once the record has been used to update the data warehouse.

4.2.1.5.2 Detailed Functional and Performance Requirements

A sqlplus session is started (cp_err.sql) and the following procedures are executed.
- First the records from the error tables are copied to the staging tables. This is done by the procedure copy_err_gen_fail.(This procedure is created in Q:\copy_err.sql).
- The key fields are populated with not null values by executing the procedure val_gen.(code for creating this procedure in Q:\val_gen.sql)
- The duplicate rows are removed from geninfo
- Check for errors and put the bad records in error tables by executing the procedure val_geninfo.(The code for creating this procedure is in Q:\valid_dat_gen.sql)
- The key fields are populated with not null values by executing the procedure val_fail.(code for creating this procedure in Q:\val_fail.sql)
- The duplicate rows are removed from failinsp
- Check for errors and put the bad records in error tables by executing the procedure val_failinsp.(The code for creating this procedure is in Q:\valid_dat_fail.sql)
- The procedure load_lot_ins_head is executed.(This procedure is created using Q:\load_lih.sql)
- The procedure load _end_it_def is executed (This procedure is created using Q:\load_eid.sql)

4.2.2 Record Transfer from Lotus Notes to QDMS (AVISEND)

4.2.2.1 Program Description

The AVI stores their quality records on a Lotus Notes server at the VetCom in San Antonio. A replication process is used to copy the quality records to a Lotus Notes server at DPSC. The Lotus Notes server at DPSC has installed a "Lotus Notes Pump", an application that can transfer data between an Oracle database and Lotus Notes applications. This data pump application (AVISEND) will retrieve any new, updated or deleted records from the Lotus Notes server and copy them to Oracle staging tables: "AVI_HEADER" and "AVI_ENDITM" which are located on the QDMS server at DPSC. This process will occur at scheduled intervals and is controlled by the Lotus Notes Server. The staging tables will contain at all times the identical information as contained in the Lotus Notes database. Therefore deletions will get propagated into the staging tables. Constraints are enforced on the staging tables, which means that only valid data can be transferred. Invalid data will result in a message back to the Lotus Notes Administrator indicating why a record could not be propagated. The software code "AVISEND" was developed by the AVI and is located on the Lotus Notes server. The code will be maintained by the AVI.

The Lotus Notes pump is bi-directional and will maintain the following tables on the Lotus Notes server with information from the QDMS application:
- Product_Lots Table
- Products Table
- Defects Table

Lotus Notes pump communicates with the QDMS database over Sequal_Net using the "Name Pipes" protocol.

4.2.2.2 Detailed Functional and Performance Requirements

- replicate the AVI Lotus Notes quality database to QDMS staging tables
- replicate product setup tables and product lot tables to the AVI Lotus Notes database.

4.2.3 AVI Data Import (AVIIMPO)

4.2.3.1 Program Description

This program module is a scheduled program that executes various modules in a prededined order at specific time periods. The initial scheduling trigger for this program module is initiated by Windows NT scheduler (AT2.job). This job will start a batch file (D:apps\qdms\perl\avi.pl) which calls a sqlplus procedure: "avi_imp.sql " This procedure calls three database procedures: avi_lot_ins_head, avi_det and del_rec_lih. These procedures extract any changed or new records and propagate these records into designated production tables of the QDMS application : "LOT_INSPECTION_HEADERS" and "END_ITEM_DEFECTS".

Note: Deletions of records will propagate into the production tables if this deletion occurs within 90 days of the date that the record was initially submitted to QDMS. This functionality allows the AVI to archive their data after 90 days. Archiving of Lotus Notes data before the 90 day time limit would mean the deletion of records from their Lotus Notes database and from the QDMS staging and production tables. Archiving of Lotus Notes data after the 90 day time limit would mean the deletion of records from the Lotus Notes database and the QDMS staging tables but not from the production tables.

New or updated records that are placed in the Operational Database will be flagged (DW flag) for the MONITOR. This flag is reset by the INTEGRATOR once the record has been used to update the data warehouse.

18

4.2.3.2 Detailed Functional and Performance Requirements

- NT Scheduler start application at a pre-selected time: avi_imp.sql

- Avi_imp.sql calls first the database procedure named "avi_lot_ins_head". The code for this procedure can be found in "load_avi_lih.sql". This procedure creates or updates the inspection headers records in the "Lot_Inspection_Headers" table based on records from AVI staging table "AVI_HEADER".
  New records are based on the following:
  - ◆ Lot Number
  - ◆ Split Lot Id
  - ◆ NSN
  - ◆ Producer 3 letter code
  - ◆ Date Time Inspection
  - ◆ Inspected By = AVI
  The combination of the above fields will determine unique inspection header records. When new records are inserted into inspection headers, a unique Inspection ID is assigned to the record.
  Records are processed if "Date Modified" is after "Date Processed". This flags that a record has been updated. New records will have "Date Modified" = "null" and will also be processed. If a inspection header record is updated then the corresponding defect records belonging to the "old" record are deleted and updated by the following procedure

- After the previous procedure has processed all inspection headers, Avi_imp.sql calls the database procedure named "avi_det". The code for this procedure can be found in "load_avi_det.sql". This procedure creates or updates the inspection defect records in the "End_Item_Defects" table. To insert detail records the following steps must be performed.
  - ◆ Get QDMS assigned inspection ID corresponding to AVI header record.
  - ◆ Check for duplicates in QDMS end item defects for the Inspection ID retrieved
  - ◆ If no record found insert a new record, else update the record.
  - ◆ Records are processed if "Date Modified" is after "Date Processed". This flags that a record has been updated. New records that have "Date Modified" = "null" and will also be processed.
    Unique records are based on Inspection ID, and Inspection Detail ID. The Inspection Detail Id in the record is the same as the detail ID in avi_enditm.

- Next, Avi_imp.sql calls a database procedure named "del_rec_lih". The code for this procedure can be found in "del_rec_lih.sql". This procedure checks for the records in avi_header which exist in lot_inspection_headers(avi records). If the record does not exist in avi_header and the record is less than 90 days old in lot_inspection_headers that record will be deleted from the lot_inspection_headers. The corresponding detail records in end_item_defects will be deleted too if they exist.

4.2.3.3 Special Requirements

4.2.4 QDMS File Based Transfer Module

4.2.4.1 Introduction

This module allows producers to send their lot, receipt and inspection information to QDMS via file based data transfer. A specification was developed for each record to be transmitted, which is described in Technical Working Paper #202. A user manual for this module was issued send to the producing companies in March 1999 which describes how to use this module.
This document describes the functionality of the various modules that were developed that receive and process the data.

4.2.4.2   Program Description

4.2.4.2.1 Data Receipt

The user can access the encrypted data transfer module via URL:
> https://www2.dscp121.dscp.dla.mil/wa/owa_dba/owa/imgload.show_form.

A secure socket layer (SSL) is used to encrypt the communication between the users browser and the qdms server. Therefore the user needs to use a browser that supports SSL.

The web server is automatically started via the general batch start up file that starts other web services. The server needs to have an updated security certificate which is typically valid for only one year.

The following program contains the logic to receive the files, rename them and store them in a directory
> OWAREPL.DLL,   11/6/98
> OWAREPL.EXE,   11/6/98

The source code for these program modules is stored in the following directory:
> D:\orant\oawarepl\owarepl\src

The transmitted files are renamed to UserID+Sequential_Number.txt and stored in the following directory:
> D:\orant\owarepl\download

A log file of each connection is maintained in D:\orant\owarepl with a name similar to:
> logWRB_DBAUTH_19535024.log

4.2.4.2.2 Data Loading to Staging Tables

The following database tables are being used to receive the data that is being transmitted by the producers
- STAGING_PRODUCT_LOTS
- STAGING_SUB_LOTS
- STAGING_LOT_COMPONENTS
- STAGING_LOT_INSPECTION_HEADERS
- STAGING_INPROCESS_ATTRIBUTE
- STAGING_END_ITEM_DEFECTS
- STAGING_VARIABLE_DATA
- STAGING_RECEIVING_HEADER
- STAGING_RECEIVING_DETAILS

Program module that loads data from ASCII files into staging tables
Upload.pl	main program modules that runs at set intervals and call varies subroutines
Trunc_stables.sql	sub routine called to clear (truncate) staging tables
Qdms1.cntl	subroutine called to load text files into staging tables
update_filename.sql	subroutine called to insert file name/username into staging table

A record of loading the staging tables is maintained *in o:\apps\qdms\file_upload.*
The processed file are moved to the q:\upload_archive directory for archival purposes
Staging tables are cleared before new files are processed for uploading instead of clearing after completion of copying the record to either the error table or the production tables

4.2.4.2.3 Data Validation

This Program module validates data in staging tables and adds the data to the production tables or generates an error record
> **Upload.pl,**   main program modules that runs at set intervals and call varies subroutines

20

Up_err.sql, routine that is called to execute the loading of varies production tables
- truncate table file_upload_errors:
- execute Load_prod_table.load_product_lots;
- execute Load_prod_table.load_lot_components;
- execute Load_prod_table.load_sub_lots;
- execute Load_prod_table.load_lot_inspection;
- execute Load_prod_table.load_inprocess_attribute;
- execute Load_prod_table.load_variable_data;
- execute Load_prod_table.load_end_item_defects;
- execute Load_prod_table.load_receiving_header;
- execute Load_prod_table.load_receiving_details;

Upload_prod_tables.sql, file that contains the stored procedures: "Load_prod_table" that are called by "Up_err.sql". This procedure validates first the record by calling nested procedures listed in "File_upload.sql" and then either inserts or updates the record in the production table

File_upload.sql & File_upload1.sql, files that contain stored procedures: "upload_errors" that are called by "load_prod_table" procedure. These nested procedures validate each non processed record in the staging table and marks it if error found. It also writes a record to the error table: "file_upload_errors" if error found.

4.2.4.2.3.1 Error Coding

The following error checks are made on data that is located in the indicated table:

4.2.4.2.3.1.1   Table name: Staging_Product_lots

1) Check for valid location_3letter_code
   Location_3letter_code must be in table organization_locations **Error code = 300**
2) Lot_number should not be null **Error code = 110**
3) Split_Lot_Id should not be null. **Error code = 115**
4) Check for valid NSN
   NSN must be in table products. **Error code = 200**
5) Check for valid packaging_method **Error code = 120**
   Packaging_Method must be in table Packaging_method.
6) Check for valid contract_number **Error code = 310**
   Contract_Number must be in table Contract_headers.
7) Check for valid contract_item **Error code = 320**
   (ContractNumber,Contract item) must be in table Contract_products
8) Lot_quantity should not be null **Error code = 210**
9) Date produced should be a valid format 'DD-MON-YYYY' **Error code = 101**
10) Location_3letter_code must be equal to User_Company **Error code = 910**
11) One additional check is making sure that NSN is valid for the contract item or is a sub
    component of the contract item NSN **Error code = 130**
    NSN in
    (select component_nsn
     from product_structure ps
     connect by prior ps.component_nsn = ps.nsn
     start with ps.nsn = (select cp.nsn
        from contract_products cp
        where cp.contract_number = :prodlot.contract_number
        and   cp.contract_item   = :prodlot.contract_item)
    union
    select cp.nsn

```
                from contract_products cp
                where cp.contract_number = :prodlot.contract_number
                and   cp.contract_item   = :prodlot.contract_item)
```

4.2.4.2.3.1.2 Table_name: Staging_Sub_lots

1) Check for valid location_3letter_code. **Error code = 300**
2) Lot_Number should not be null. **Error code = 110**
3) Split_Lot_Id should not be null. **Error code = 115**
4) Check for valid NSN. **Error code = 200**
5) Ensure columns1..4 exist as a valid product lot in product_lots table. **Error code = 400**
6) Sub_lot_id should not be null **Error code = 140**
7) Sub_lot_quantity should not be null **Error code = 150**
8) Check for valid unit_of_measurement in UOM table **Error code = 920**
9) Location_3letter_code must be equal to User_Company **Error code = 910**

4.2.4.2.3.1.3 Table_name: Staging_Component_Lots

1) Check for valid location_3letter_code **Error code = 300**
2) Lot_Number should not be null. **Error code = 110**
3) Split_lot_id should not be null. **Error code = 115**
4) Check for valid NSN. **Error code = 200**
5) Ensure columns1..4 exist as a valid product lot in product_lots table. **Error code = 400**
6) Check for valid Component_3letter_code **Error code = 300**
7) Component_Lot_No should not be null. **Error code = 110**
8) Component_Split_Lot_Id should not be null. **Error code = 115**
9) Check for valid Component_NSN **Error code = 200**
10) Ensure columns 6..9 exist as a valid product lot in product_lots table. **Error code = 400**
11) Component_Lot_Quantity must not be null. **Error code = 220**
12) Location_3letter_code must be equal to User_Company **Error code = 910**

4.2.4.2.3.1.4 Table_name: Staging_Lot_Inspection_Headers

1) Check for valid Inspection_Type . Make sure that column Inspection_Type exists in table Inspection_Type. **Error code = 510**
2) Make sure Vendor_Inspection_Id is not null. **Error code = 520**
3) Check for valid location_3letter_code. **Error code = 300**
4) Lot_Number should not be null. **Error code = 110**
5) Split_lot_id should not be null. **Error code = 115**
6) Check for valid NSN . **Error code = 200**
7) Ensure columns1..4 exist as a valid product lot in product_lots table. **Error code = 400**
8) Date_Inspected entered must be of format (DD-MON-YYYY). **Error code = 100**
9) Check for valid Inspected_by column. **Error code = 300**
10) Lot_Quantity_Inspected must not be null. **Error code = 900**
11) Lot accepted must valid for inspection_type in ('1','6') . **Error code = 515**
12) Inspected_by must be equal to User_Company. **Error code = 910**

4.2.4.2.3.1.5 Table: Staging_Inprocess_Attribute

1) Vendor_Inspection_Id must be not null column **Error code = 520**
2) Column Inspection_Type must exists in table Inspection_Type **Error code = 510**
3) Date_Inspected must be of right format (i.e., DD-MON-YYYY). **Error code = 100**
4) Check for valid Inspected_By column. **Error code = 300**
5) Vendor_Inspection_Id, inspection_type, date_inspected, inspected_by must be a unique inspection_id in table lot_inspection_headers. **Error code = 530**
6) Inspection_Detail_Id must be not null column **Error code = 540**
7) Defect_Id must be in table Inspection_Defects. **Error code = 800**

```
        FROM  INSPECTION_DEFECTS INSPDEF2
```

```
WHERE  INSPDEF2.RESULT_TYPE = 'IP_Attribute'
and    INSPDEF2.DEFECT_ID in
(select id.defect_id
 from inspection_defect_xref id
 where id.inspection_type = :LIH.inspection_type
 and  id.defect_id in (select pc.defect_id
       from product_class_defects pc
       where pc.product_class = (select pr.product_class
                     from products pr
                     where pr.nsn = :LIH.nsn))
)
```

8) Defect_Id must be valid for inspection_type **Error code = 810**

9) Defect_Id must be valid for product_class. **Error code = 820**
10) Number_Defects must not be null column. **Error code = 700**
11) Inspected_by must be equal to User_Company. **Error code = 910**

4.2.4.2.3.1.6   Table: Staging_End_Item_Defects

1) Vendor_Inspection_Id must be not null column. **Error code = 520**
2) Column Inspection_Type must exists in table Inspection_Type. **Error code = 510**
3) Date_Inspected must be of right format (i.e., DD-MON-YYYY). **Error code = 100**
4) Check for valid Inspected_By column. **Error code = 300**
5) Vendor_Inspection_Id, inspection_type, date_inspected, inspected_by must be a unique inspection_id in table lot_inspection_headers. **Error code = 530**
6) Inspection_Detail_Id must be not null column. **Error code = 540**
7) Defect_Id must be in table Inspection_Defects. **Error code = 800**
8) Defect_Id must be valid for inspection_type **Error code = 810**
9) Defect_Id must be valid for product_class. **Error code = 820**
10) Sample_Size_Taken must be a not null column. **Error code = 600**
11) Number_Defects must be not null column. **Error code = 700**
12) Inspected_by must be equal to User_Company. **Error code = 910**

4.2.4.2.3.1.7   Table: Staging_Variable_data

1) Vendor_Inspection_Id must be not null column. **Error code = 520**
2) Column Inspection_Type must exists in table Inspection_Type. **Error code = 510**
3) Date_Inspected must be of right format (i.e., DD-MON-YYYY). **Error code = 100**
4) Check for valid Inspected_By column. **Error code = 300**
5) Vendor_Inspection_Id, inspection_type, date_inspected, inspected_by must be a unique inspection_id in table lot_inspection_headers. **Error code = 530**
6) Inspection_Detail_Id must be not null column. **Error code = 540**
7) Defect_Id must be in table Inspection_Defects. **Error code = 800**
8) Defect_Id must be valid for inspection_type **Error code = 810**
9) Defect_Id must be valid for product_class. **Error code = 820**
10) Sample_Size_Taken must be a not null column . **Error code = 600**
11) Average must be not null. **Error code = 610**
12) Std_Dev must be not null. **Error code = 620**
13) Result_UOM must be not null and exists in table UOM. **Error code = 920**
14) Ensure Result_UOM is valid for specified defect . **Error code = 925**
15) Inspected_by must be equal to User_Company. **Error code = 910**

4.2.4.2.3.1.8   Table: Staging_Receiving_Header

1) Document_Number should not be null. **Error code = 930**
2) Date_Received must be in right Format (i.e.. DD-MON-YYYY) and

must be not null **Error code = 940**
3) Check for valid Received_By Column. **Error code = 300**
4) Date_shipped must be of Format DD-MON-YYYY and must be not null. **Error code = 950**
5) Document_type cannot be null. **Error code = 970**
6) Document_date must be of format DD-MON-YYYY and must be not null. **Error code = 960**
7) Received_by must.be equal to User_Company. **Error code = 910**

4.2.4.2.3.1.9 Table: Staging_Receiving_Details

1) Make sure Document_Number exists in Receiving_Header . **Error code = 930**
2) Date_Received must be in right Format (i.e., DD-MON-YYYY). **Error code = 940**
3) Check for valid Received_By Column. **Error code = 300**
4) Check for a valid Location_3letter_code. **Error code = 300**
5) Lot_number should not be null. **Error code = 110**
6) Split_lot_Id should not be null. **Error code = 115**
7) Check for a valid NSN. **Error code = 200**
8) Ensure columns 4..7 exist as a valid product lot in product_lots table. **Error code = 400**
9) Check for a valid Contract_number. **Error code = 310**
10) Check for a valid Contract_Item. **Error code = 320**
11) Check for valid Delivery_Order_no .
12) Check for valid Delivery_line_Item
13) If Delivery_order_no is not null then check fields 9..12 in
    Contract_Delivery_Schedules table. **Error code = 990**
14) Received_Quantity must be not null. **Error code = 980**
15) Check whether the producer is online and the received lot is in the system . **Error code = 400**
    Location_3letter_code must be equal to User_Company. **Error code = 910**

4.2.4.2.4 Error Report

This program module sends at set intervals an e-mail to a designated e-mail address with a report attached that indicates if any errors and what type of errors were found since the last data transmission. This module needs to be in sync with the data processing as the error table is cleared before new records are processed.

It is important the mail program such as Outlook is running and is logged onto the mail server with a valid mail server account.

4.2.5 Data Warehouse Monitor (MONITOR)

4.2.5.1 Program Description

The datawarehouse is a data storage for data that most likely and frequently are accessed and queried by the end users. Data stored in the datawarehouse is highly summarized, semi-summarized or not summarized at all. The datawarehouse consist of tables designed in such a way that it would help and fasten end-user query. The datawarehouse is populated and updated from data or records originated from the Operational database through the mean of the Monitor and the Integrator.

The Monitor will flag records that are new, changed or deleted. The code of the Monitor is part of the various modules that initiate the insertion of new records, updates existing records. The code which flags a deleted record is a database trigger. Records in the tables "Product_Lots", Lot_Inspection_Headers", End_Item_Defects", "Inprocess_Attribute_Defects" and "Variable_Data" have a field named "Record_Changed" which identifies if the record was new/updated since the last Data Warehouse Update. The table "Deletion_Log" contains information on records that were deleted by identifying the table from which a record was deleted and the lot production of the lot to which this record belonged.

The Integrator will process new, changed or deleted records on a scheduled basis, update the data warehouse tables and set the Monitor flag back to "N".

4.2.5.2 Detailed Functional and Performance Requirements

- Set "Record_Changed" to "Y" if new record was inserted or existing record was changed
- Insert record in Deletion_Log table to identify a record that was deleted from a production table, record will contain the name of affected table and the production date of the lot to which this record belonged

4.2.5.3 Special Requirements

4.2.6 Data Warehouse Integrator (INTEGRATOR)

4.2.6.1 Program Description

The Integrator will process new, changed or deleted records on a scheduled basis, update the data warehouse tables and set the Monitor flag back to "N". It will update various data warehouse tables which contain highly summarized, semi-summarized or not summarized data. Summarization is done on production date time frames. Summarization is done on a daily, monthly and yearly basis. Data analysis queries are designed to use this summarized data in the most effective manner to fasten the end-user query.

The integrator can run in two modes. In the first mode, it cleans and updates all data warehouse tables irregardless if the underlying records were altered or not. In the second mode, it updates only those records in data warehouse that were affected by the new, changed or deleted record in the underlying production tables. At the current time the first mode is scheduled to run once a week and is triggered by a scheduled job, which calls "weekly_sched.sql". The second mode is called every night and is triggered by a scheduled job, which calls "new_sched.sql". Various database procedures are called by these modules to collect necessary information for update from the operational database. If the needed information is found in the operational database, the integrator will then perform the necessary calculation and summarization to update the data warehouse.

The Integrator can also be triggered manually via the form "Update WH Tables"

4.2.6.2 Detailed Functional and Performance Requirements
- Pickup the records identified by the "Record_Changed" flag

25

- Perform the necessary summarization calculations
- Update the warehouse
- Reset the "Record_Changed" flags for the processed records

4.2.6.3 Special Requirements

The following tables are the main Data Warehouse tables:
DEFECTIVE_LOT_DAILY
DEFECTIVE_LOT_MONTHLY
DEFECTIVE_LOT_YEARLY
INSPECTION_DEFECT_DAILY
INSPECTION_DEFECT_MONTHLY
INSPECTION_DEFECT_YEARLY
LOT_QUANTITY_INSPECTED_DAILY
LOT_QUANTITY_INSPECTED_MONTHLY
LOT_QUANTITY_INSPECTED_YEARLY
PRODUCTION_DAILY
PRODUCTION_MONTHLY
PRODUCTION_YEARLY
REWORK_DAILY
REWORK_MONTHLY
REWORK_YEARLY
TOTAL_LOT_DAILY
TOTAL_LOT_MONTHLY
TOTAL_LOT_YEARLY
VARIABLE_DEFECT_DAILY
VARIABLE_DEFECT_MONTHLY
VARIABLE_DEFECT_YEARLY

The database procedures used by the Integrator are listed in the following sql files:
Newpackagebody.sql
Del_handler_body.sql

4.2.7 Dynamic Web Page Generator

Each form, graph or report that interacts with the database, either for data entry, data analysis or data reporting will be dynamically generated by the Web Page Generator and made available to the user via an Intranet (DPSC) or Internet (Other Users).

4.2.7.1 Program Description

In the following paragraphs we will discuss the detailed functionality of each of the menu items. The source code for the forms that are called by the menu is stored in "fmb" files, while the executable code resides in "fmx" files. The source code for the menu itself is stored in the "Ves_menu.mmb", including the access rights of each user group. The following table indicates the file name that belongs to each menu item.

Menu Item	File Name (*.fmb)
Administration	
Users	people
Error Codes	errors
Lot Activity Types	lot_activity
Setup	
Products	products
Defect Types	defects

Inspection Types and their Defects	insptype
Product Classes and their Defects	prdclass
Organizations	orgs
Organizations and Locations	orgloc
Analysis Defects	anal_def
Container Types	container
Family of Rations	rations
Packaging Methods	pack_method
Product Processes	prodproc
Unit of Measure	uom

DPSC Maintenance

Contracts	contracts
Waivers	waivers
Geninfo Errors	generrs
Failinsp Errors	failerrs
Update WH Tables	new_whup

Lot Transactions

Create Lots	lots
Lot Receipts	receipt
Finished Lots/Receipt Inspections	insp
Inprocess Inspections	inp_insp

Analysis

Data Analysis

Production Analysis	Pr_NewAn
Inspection Defect Analysis	ID_NewAn
Variable Defect Analysis	Vd_NewAn
Defective Lot Analysis	Dl_NewAn
Rework Analysis	Rw_NewAn

Trend Analysis

Production Trend	Pr_NewTr
Inspection Defect Trend	ID_NewTr
Variable Defect Trend	VD_NewTr
Defective Lot Trend	DL_NewTr
Rework Trend	RW_NewTr

Analysis Reporting

Production Detail	prodrpt
Inspection Defect Detail	insderpt
Variable Defect Detail	varderpt
Defective Lot Detail	delotrpt
Rework Detail	rewrpt

Reports

Contracts	ctr_rpt
Products	prod_rpt
Product Structure	pro_str_rpt
Lot Activity	lot_rpt
Product Class/Defects	prdclrpt
Insp. Type/Product Class/Defects	defr
USDA Inspection Header Errors	gen_rpt

USDA Inspection Detail Errors	er2_rpt
Organization/Locations	orgs_rpt
Organization/Products	org_pro_rpt
Product Lot	prdl_rpt
Lots Produced	venlotrpt
Receipts	rec_rpt
Waivers	waiv_rpt

Note: Certain forms make calls to other forms. graphs and/or reports

4.2.7.2 Detailed Functional and Performance Requirements

4.2.7.2.1 Users

This module is used to describe the various users that have access to QDMS, including the plant location for which they report or view data. Assignment of Login ID's and Passwords are a DBA function and can not be created and/or edited via this form.

4.2.7.2.2 Error Codes

This module is used to edit the description of error codes that are being used by the QDMS application

4.2.7.2.3 Lot Activity Types

This module is used to describe the different types of activities that can be performed on a lot. Some of the activity types might include Vendor Inspection, USDA Inspection, AVI Inspection, Reworked by Vendor, Shipped By Vendor, etc. These lot activity id's are used within the software code to build a log table of various activities that are performed on lots.

4.2.7.2.4 Products

The product form is a two block form to enter product descriptions and their relationships to component products into the system.

The first block is used to enter information such as National Stock Number, Three letter code, Product Description, Military Specification, Product Type, Process Type, and Container Type. Product Type information is validated through the foreign key link to the Product Type Table. Process Type information is validated through the foreign key link to the Process Type Table. Container Type information is validated through the foreign key link to the Containers type table. The product type, process type and container type fields are used during the data analysis phase in which "groups" of products can be analyzed rather than a singular product.

The second block is used to establish a relationship between "assembled" products and their components. This relationship is important during the validation of contract numbers and the products that can be produced under these contracts. A component product is validated against a contract line item for an assembled product to which this component belongs. The user can also identify the quantity for the component that is to be used to make the assembled product. If the quantity is less than "1" or "null" then the component will not show in the LOV of products that can be produced for the assembly. This feature can be used by DSCP to limit the LOV to active components.

4.2.7.2.5 Defects

This function is used to enter all possible defects into the system that can be used by the QDMS application. Each defect will be assigned a "Result Type". There will be four possible result types: "In Process Attribute", "In Process Variable", "Finished Product Attribute" and "Finished Product Variable". Each defect description will be given a defect id. Both the USDA and the AVI need to cross reference their defect description to a defect id used by the QDMS application. To keep the defect descriptions organized the following defect id ranges should be used for the following result types:

28

```
FP_Attribute:    1000 – 3999
FP_Variable:     4000 – 4999
IP_Variable:     5000 – 5999
IP_Attribute:    7000 – 7999
```

Each defect id can be cross-referenced to a higher level of defect description: "Analysis Defect". This can be used to perform drill down analysis in which various defects are grouped together at a higher level. For example all the various product defects can be cross-referenced to an Analysis defect "Product Defect" and an analysis could be performed on all defects that are "Product Defects"

4.2.7.2.6 Product Classes

This function is used to identify various product classes that can be used by the QDMS application. Each product will be assigned to one of these product classes. A total of eight (8) product classes have been assigned so far:

1	Thermostabilized MRE Pouch Product
2	Thermostabilized Tray Can Product
3	Component Type Products
4	Bakery Products
5	Spread Type Products
6	Dehydrated Products
7	Assembled Meal Bags
8	Other Type Products

Each product class will be cross-referenced to a distinct set of defects that are applicable for that particular product class regardless of the inspection location.

4.2.7.2.7 Inspection Types

This function is used to identify the various inspection types/locations that can be used in the QDMS application. There will be six inspection types used by the system:

1	End Item Inspection
2	Pre Retort Inspection
3	Post Retort Inspection
4	Rework Inspection
5	Assembly Inspection
6	Receipt Inspection

Inspection ID's 1 and 6 are "lot inspections" leading to an acceptance or rejection of the lot. All other inspection types are considered "in-process inspections" or screening processes in which non-conforming units are removed from the lot, but not necessary lead to an acceptance or rejection of the lot.

A cross-reference is made between each inspection type and the allowable defects that can be recorded for this inspection type, regardless of the product class.

4.2.7.2.8 Organizations

This function is used to create or edit information regarding Organizations. Organizations are required as Contracts are awarded to Organizations. One Organization can have multiple Plant Locations. This information is managed by the form "Organizations and Locations"

4.2.7.2.9 Organizations and Locations

This function is used to identify organizations to whom contracts can be awarded and the various (production) locations that can be identified for this organization. Each organization-location will be assigned a unique three letter code that will be used to record quality records in the QDMS database, and will be used to identify unique product lots. The identical three letter organization-location codes have to be used by all inspection organizations such as the USDA and the AVI when they send inspection

information.
This form has also a field "Online". This field is used to identify if a company enters data into QDMS. If this field is set to "Y" then the USDA records will not be imported till a lot has been created by the vendor. If this flag is set to "N" then the USDA record will be used to generate a lot which will then facilitate the import of the USDA record

4.2.7.2.10 Analysis Defects

This function can be used to identify a higher level defects to be used by users of the QDMS application who want to perform more advanced analysis.. Analysis defects are meant to perform higher-level defect analysis in which various defects are grouped together. The cross-reference between defects and analysis defects are described in paragraph: "Defects".

4.2.7.2.11 Container Types

This function is used to identify various container types and sizes that can be used by the QDMS system. Each product is assigned to one of these container types.

4.2.7.2.12 Family of Rations

This function is used to identify the various family of rations that can be used by the QDMS system. The Family of Ration is used in the contracting function where each line item is assigned to a family of ration. During data analysis, the "family of ration" variable can be used to select just those process records that belong to a certain family of ration. Note: the family of ration was not assigned to specific products as products can be used in multiple rations.

4.2.7.2.13 Packaging Methods

This function is used to identify various packaging methods that can be used by the QDMS system. The main intent is to differentiate between horizontal form fill seal and vertical form fill seal quality records or between quad laminate and tri-laminate packages. Therefore, each time that a producer generates a new lot in the system, he needs to identify the packaging method he employed to create that lot.

4.2.7.2.14 Product Processes

This function is used to setup various process types that can be used by the QDMS system. The objective of a process type variable is to divide product classes into smaller categories, such that analysis can be done on this sub category. Each product can therefore be assigned to one of these process types. For example Thermal Stabilized MRE's can be divided into pumpable products, placeable products, hot fill products, etc.

4.2.7.2.15 Unit of Measure

This module is used to setup various units of measures that can be referred to by other modules.

4.2.7.2.16 Contract Data Entry

The contract module is used to enter contracts into the system. The entry form has three main blocks. The first block is the contract header block which is used to enter contract numbers, the contract grantor and the contract grantee and the contract start and end dates. Any Reference to grantors or grantees is validated through the foreign key link to the organizations table. Additional field in the contract header is: "Research Contract". This field is used to differentiate between quality information generated under production type contracts and those records which are associated with research type contract. Therefore data analysis could be performed to exclude quality records that are associated with research type contracts.

The second block is the contract detail block. This block is used to enter line item information for each product that is contracted for, including the minimum and maximum quantity. Any reference to Product in the line item detail level is validated through the foreign key link to the products table. Additional field in this block is the "Family of Ration". Each contract line item is assigned to a specific "Family of Ration". This information is used during the analysis phase, by allowing the user to select the records for one

30

particular family of ration.

The last block is the contract delivery schedule block. This block is used to enter delivery order information for each contract line item. Fields will be available to record the quantity of requested delivery, the time frame for the delivery, the shipper and the destination.

4.2.7.2.17 Waivers

The function of this module records any waiver that is granted by DSCP. The module requests information such as lot description, data of the waiver, person granting the waiver and any comments regarding the waiver.

4.2.7.2.18 Geninfo Errors

Whenever the Data-Normalization procedures for the USDA data find an error in records stored in the Geninfo staging table, it will mark the records as Bad-Data and store the record in Geninfo_Errs. It is the task of this module and an authorized end-user to look at these erroneous records and fix them. The corrected record will be normalized and imported into the Operational Database during the next cycle.

4.2.7.2.19 Failinsp Errors

Whenever the Data-Normalization procedures for the USDA data find an error in records stored in the Failinsp staging table, it will mark the records as Bad-Data and store the record in Fail_Errs. It is the task of this module and an authorized end-user to look at these erroneous records and fix them. The corrected record will be normalized and imported into the Operational Database during the next cycle.

4.2.7.2.20 Update Date Warehouse Tables

This module allows the user to manually update the data warehouse tables either incremental update data warehouse tables or to completely clean and regenerate all data warehouse tables

4.2.7.2.21 Create Lots

The producer and assembler which need to create a lot in the QDMS system will use this module. This module will require certain information describing that lot such as lot id, NSN, date of production, quantity produced, etc. Also this table will have a field in which producer can identify a code for packaging type used (this will be used to identify horizontal form fill seal versus vertical form fill seal). Due to the fact that producers are allowed to split a lot, each lot needs to be identified with a split lot id. Split lot identifier "A" will be assigned to lots that are not identified with a split lot identifier. This also means that the producer has to mark the second "slit lot" with an other identifier than "A.

In addition to some production information, the producer is also required to enter the contract number that the lot is offered against. Internal validations will be performed by QDMS to assure that a valid contract number and line item number is used and that the product code is valid under that contract line item.

Each lot has a flag that identifies if the lot was offered to the Government. This flag is set to "N" by default and set to "Y" after a inspection record for that lot from either the AVI or USDA has been received. This feature assumes that each lot will be inspected and accepted by an Government Inspection Agency

A sub form (Sub Lots) can be called from the main form in which the producer can record which sub lot codes belong to the lot.

A second sub form (Component Lots) can be called from the main form in which the producer can record the lot codes of various components/ingredients that were utilized to manufacture the lot.

4.2.7.2.22 Lot Receipt

This module is used to record the receipt of a lot by a specific company. Companies can only conduct

inspections on lots produced by them or on lots that were received by them. The form is of the master detail type. The header is used to record items such as shipping information. The detail block is used to record information on each item that was received during that shipment. This information will be cross checked against the contract information.

If a company receives a lot which is not existing in the database then the user will be asked if it should create the lot.

4.2.7.2.23 Finished Lot/Receipt Inspection

This end item inspection module consist of a header in which details are given on who performed an inspection, the lot id, the quantity inspected, the date of the inspection and the result of the inspection. The module has two detail tables in which one will record details regarding defects found during the inspection process and any variable data.

4.2.7.2.24 Inprocess Inspection

This in-process inspection module consist of a header in which details are given on who performed an inspection, the lot id, the quantity inspected and the date of the inspection. The module has two detail tables in which one will record details regarding defects found during the inspection process and any variable data.

4.2.7.2.25 Production Defect Analysis and Reporting

This analysis will display to the user the defect data that is recorded for lots that were produced during a specified time period. This defect data will only pertain to the so-called "in_process" inspection points (inspection type: 2,3,4,5). The results will be displayed as a bar chart displaying the average defect ratio (total defects found during the various inspections/total lot quantity produced as recorded in the lot header).

The sources of the analysis are the DWH table:
PRODUCTION_DAILY
PRODUCTION_MONTHLY
PRODUCTION_YEARLY
&
TOTAL_LOT_DAILY
TOTAL_LOT_MONTHLY
TOTAL_LOT_YEARLY

The DWH tables contain summarized data based on time increments. The DWH summarization is done based on producing company, NSN, product family, inspection type, inspected by, and defect id. Summarized will be the total number of defects found, the total number of lots produced, and the total number of units produced (including those lots, which did not have that particular defect). Calculated and stored with each DWH record will be the percent defect, as calculated by the ratio of total number of defects and total units produced.

The analysis module will compute the total number of defects and total units produced over a given time period based on DWH data and production table data and compute and display the percent defect. The data will be grouped for analysis purposes "by producer", "by product" or " by defect". If the user, which is not a DPSC analyst, can only see those records that were either generated by that location or belong to a lot that was produced by that location. Analysis based on "by time increments" will be explained under "Trend Analysis".

The analysis form will have several sections:
Section A is used to setup the query input variables: Company Code, NSN, Product Family, Product Class, Container Type, Process Type, Defect, Production Start Date, Production End Date, Inspection Type.
Section B is used to select the type graph (grouped by) is desired
Section C is the actual display of the graphical analysis result based on the input parameters. The graphics

will display the percent defects found on the Y-axes and display on the X-axes the "grouped by" parameter. Indicator for the "grouped by", will be the Location_3Letter_Code, or the Products_Three _Letter_Code, or the Defect_Description

Section D is a spreadsheet, which displays the summarized data, used for the graphics display and is based on the input parameters.

Three buttons will be displayed on this form. The first button will execute the query, generate the graphics and the spreadsheet based on the parameters set in section A and B. The second button will generate a hard copy report of the analysis. The third button is a "list" button which can be used by the user to get a suggested list box for the input variable in which the cursor is located. If the cursor is located in a field for which there is no look up list available then messages to that extend will be displayed.

An analysis report will be included which lists a summary record for each producing company, for each NSN, for each inspection type, and for each found defect. The summary report will display the total number of units produced for the group: company/nsn, and report the total number of defects found for each product group, broken down by inspection type. The analysis report will only display those records that were requested via the forced or unforced query input parameters.

In addition to the analysis report, a detailed report can be generated that will list all in_process defect records contained in the database. The report will have several input variables that can be selected to limit the report (see: Production Detail Report).

4.2.7.2.26 Production Variable Data Analysis and Reporting

This analysis will display to the user the variable data that is recorded for lots that were produced during a specified time period. The results will be displayed as a two dimensional x-y chart displaying the grand average of the variable selected and the average of the coefficient of variation for that particular selected variable.

The sources of this analysis are the DWH tables:
VARIABLE_DEFECT_DAILY
VARIABLE_DEFECT_MONTHLY
VARIABLE_DEFECT_YEARLY
&
TOTAL_LOT_DAILY
TOTAL_LOT_MONTHLY
TOTAL_LOT_YEARLY

The DWH tables contain summarized data based on time increments. The DWH summarization is done based on producing company, NSN, product family, inspection type, inspected by, and variable id. Summarized will be the grand average of the variable data and the average of the coefficient of variation of each variable data point.

The analysis module will compute the grand average and average coefficient of variation over a given time period based on DWH data and production table data and display this in a Line Chart with two Y-Axes. The data will be grouped for analysis purposes "by producer" or "by product". Grouping "by defect" will not be enabled, as analysis will be only possible for specific defects. If the user is not a DPSC analyst, he can only see those records that were either generated by that location or belong to a lot that was produced by that location. Analysis based on "by time increments" will be explained under "Trend Analysis".

An analysis report will be included which lists a summary record for each producing company, for each NSN, for each inspection type, and for each found defect. The summary report will display the total number of units produced for the group: company/nsn, and report the total number of defects found for each product group, broken down by inspection type. The analysis report will only display those records that were requested via the forced or unforced query input parameters.

An analysis report will be included which lists a summary record for each producing company, for each NSN, for each inspected by, and for each inspection type. The summary report will display the grand average and the average coefficient of variation. The analysis report will only display those records that were requested via the forced or unforced query input parameters.

In addition to the analysis report, a detailed report can be generated that will list all variable data records contained in the database. The report will have several input variables that can be selected to limit the report (see: Detailed Variable Data Report).

4.2.7.2.27 Rework Analysis and Reporting

This analysis will display to the user the rework information that is recorded for lots that were produced during a specified time period. This rework data will only pertain to one particular "in_process" inspection point (inspection type: 4). This analysis will display to the user the ratio of rework instances of lots that were produced during a specified time period. Separate data will be maintained for lots that were offered to the USDA or the AVI, and for lots that were never offered to the USDA or AVI..

The sources of this analysis are the DWH tables:
REWORK_DAILY
REWORK_MONTHLY
REWORK_YEARLY
&
TOTAL_LOT_DAILY
TOTAL_LOT_MONTHLY
TOTAL_LOT_YEARLY

The DWH tables contain summarized data based on time increments. The DWH summarization is done based on producing company, NSN, product family, inspection type, and inspected by. Summarized will be the total number of rework instances for the total number of lots that were offered to the Government and for lots that were never offered to the Government.

This analysis will display to the user the ratio of rework instances of lots that were produced during a specified time period and either offered or not offered to the USDA or the AVI. The results will be displayed as a barchart and grouped by either vendor, product or based on time increments. The DWH table will contain summarized data based on time increments. An analysis report will be included which lists a summary record for each producing company and NSN and list the number of lots offered to the government and the number of rework records recorded for these products.

In addition a separate report will be generated that will list all rework records contained in the database. The report will have several input variables that can be selected to limit the report.

4.2.7.2.28 Defective Lot Analysis and Reporting

This analysis will display to the user the defective lot data that is recorded for lots that were produced during a specified time period. This defect data will only pertain to the so-called "end-item" or receipt inspection points (inspection type: 1, 6). The results will be displayed as a bar chart displaying the average ratio of failed inspections and total inspections performed.

The sources of this analysis are the DWH tables:
DEFECTIVE_LOT_DAILY
DEFECTIVE_LOT_MONTHLY
DEFECTIVE_LOT_YEARLY
&
TOTAL_LOT_DAILY
TOTAL_LOT_MONTHLY
TOTAL_LOT_YEARLY

The DWH tables contain summarized data based on time increments. The DWH summarization is done based on producing company, NSN, product family, inspection type, and inspected by. Summarized will be the total lots inspected, the total number of inspections performed and the total number of inspection failures.

The analysis module will compute the total number of failed lot inspections and the total number of inspections performed over a given time period based on DWH data and compute and display the ratio of these as a percent defective lots. The data will be grouped for analysis purposes "by producer", "by product" or " inspected by". If the user which is not a DPSC analyst can only see those records that were either generated by that location or belong to a lot that was produced by that location. Analysis based on "by time increments" will be explained under "Trend Analysis". It should be noted that all records are included in this analysis, even of those lots which were never offered to the government.

The analysis form will have several sections:
Section A is used to setup the query input variables: Company Code, NSN, Product Family, Product Class, Container Type, Process Type, Production Start Date, Production End Date, Inspection Type, Inspected By:.
Section B is used to select the type graph (grouped by) is desired
Section C is the actual display of the graphical analysis result based on the input parameters. The graphics will display the percent defects found on the Y-axes and display on the X-axes the "grouped by" parameter. Indicator for the "grouped by", will be the Location_3Letter_Code, or the Products_Three_Letter_Code, or the Inspected By 3Letter Code
Section D is a spreadsheet, which displays the summarized data, used for the graphics display and is based on the input parameters.
Three buttons will be displayed on this form. The first button will execute the query, generate the graphics and the spreadsheet based on the parameters set in section A and B. The second button will generate a hard copy report of the analysis. The third button is a "list" button which can be used by the user to get a suggested list box for the input variable in which the cursor is located. If the cursor is located in a field for which there is no look up list available then messages to that extend will be displayed.

An analysis report will be included which lists a summary record for each producing company, for each NSN, for each inspection type, and for each inspection organization. The summary report will display the total number inspections performed for the group: company/nsn, and report the total number failed inspections found for each product group, broken down by inspection type. The analysis report will only display those records that were requested via the forced or unforced query input parameters.

In addition to the analysis report, a detailed report can be generated that will list all lot inspection header records contained in the database. The report will have several input variables that can be selected to limit the report (see: Defective Lot Detail Report).

4.2.7.2.29 Defect Analysis and Reporting

This analysis will display to the user the defect data from lot inspections that are recorded for lots that were produced during a specified time period. This defect data will only pertain to the so-called lot inspection points (inspection type: 1,6). The results will be displayed as a bar chart displaying the average defect ratio (fraction of total estimated defects/total quantity inspected) grouped by defects, product, producing company or inspection agency. The total estimated quantity of defects would be calculated as "the ratio (number of defects over sample size taken) times the lot quantity inspected. The total quantity inspected will be determined by adding all inspected quantities, even of those lots, which did not have that particular defect

The source of the analysis are the the DWH table:
LOT_QUANTITY_INSPECTED_DAILY

LOT_QUANTITY_INSPECTED_MONTHLY
LOT_QUANTITY_INSPECTED_YEARLY
&
INSPECTION_DEFECT_DAILY
INSPECTION_DEFECT_MONTHLY
INSPECTION_DEFECT_YEARLY
&
TOTAL_LOT_DAILY
TOTAL_LOT_MONTHLY
TOTAL_LOT_YEARLY

The DWH tables contain summarized data based on time increments. The DWH summarization is done based on producing company, NSN, product-family, inspection type, inspected by, and defect id. Summarized for that combination will be the total number of inspections performed, the total number of units inspected (including inspections on that group of lots which did not have that particular defect) and the total number of estimated defects found in that group of lots by that inspection agency. Calculated and stored with each DWH record will be the estimated percent defect, as calculated by the ratio of total number of estimated defects and total units inspected.

The analysis module will compute the total number of estimated defects and total units inspected over a given time period based on DWH data and compute and display the percent defect. The data will be grouped for analysis purposes "by inspection agency", "by product", by producing company or " by defect". If the user which is not a DPSC analyst can only see those records that were either generated by that location or belong to a lot that was produced by that location. Analysis based on "by time increments" will be explained under "Trend Analysis".

The analysis form will have several sections:
Section A is used to setup the query input variables: Company Code, NSN, Product Family, Product Class, Container Type, Process Type, Defect, Production Start Date, Production End Date, Inspection Type.
Section B is used to select the type graph (grouped by) is desired
Section C is the actual display of the graphical analysis result based on the input parameters. The graphics will display the percent defects found on the Y-axes and display on the X-axes the "grouped by" parameter. Indicator for the "grouped by", will be the Location_3Letter_Code, or the Products_Three _Letter_Code, or the Defect_Description
Section D is a spreadsheet, which displays the summarized data, used for the graphics display and is based on the input parameters.
Three buttons will be displayed on this form. The first button will execute the query, generate the graphics and the spreadsheet based on the parameters set in section A and B. The second button will generate a hard copy report of the analysis. The third button is a "list" button which can be used by the user to get a suggested list box for the input variable in which the cursor is located. If the cursor is located in a field for which there is no look up list available then messages to that extend will be displayed.

An analysis report will be included which lists a summary record for each producing company, for each NSN, for each inspection type, for each inspection agency and for each found defect. The summary report will display the total number of units inspected for the group: company/NSN/inspection agency, and report the total number of defects found for each product group, broken down by inspection type. The analysis report will only display those records that were requested via the forced or unforced query input parameters.

In addition to the analysis report, a detailed report can be generated that will list all inspection defect records contained in the database. The report will have several input variables that can be selected to limit the report (see: Detailed Lot Inspection Defect Report).

4.2.7.2.30 Trend Analysis and Reporting

Trend analyses are identical to the previous discussed analysis with the exception that the data is grouped per month or per year . Time is based on the production date of the lot.

4.2.7.2.31 Detailed Production Defect Report

This report will show lots that exist in the database which have inspection results in the "Inprocess_Attribute_Defect" table. The report can be restricted based on the input query parameters or the "View" settings based on the user privileges.

The following query parameters can be used:
    Producer
    Product Three Letter Code
    Product Class ID
    Container Type ID
    Process Type ID
    Family of Product
    Inspection Type ID
    Inspected by:
    Defect Description
    Start Date and End Date of Production
Information displayed for each lot are:
    Date of Production
    Vendor Location Code
    Product Three Letter Code
    Lot Number + Split Lot ID
    Lot Quantity (optional)
Information displayed for each In_Process Inspection_Record
    Inspection Date
    Inspection Type
    Inspected by
    Inspected Quantity (optional)
    Defect Description
    Number of Defects Found

Note: The report will be sorted in the same order as the variables above listed

4.2.7.2.32 Detailed Lot Inspection Defect Report

This report will show lots that exist in the database which have inspection results listed in the "End_Item_Defects" table. The report can be restricted based on the input query parameters or the "View" settings based on the user privileges.
The following query parameters can be used:
    Producer
    Product Three Letter Code
    Product Class ID
    Container Type ID
    Process Type ID
    Family of Product
    Inspection Type ID
    Inspected by:
    Defect Description
    Start Date and End Date of Production
Information displayed for each lot:
    Date of Production
    Vendor Location Code
    Product Three Letter Code

Lot Number + Split Lot ID
Lot Quantity Produced
Information displayed for each End Item Inspection Record
Inspection Date
Inspection Type
Inspected by
Inspected Quantity
Defect Description
Sample Size
Number of Defects Found

Note: The report will be sorted in the same order as the variables above listed

4.2.7.2.33 Detailed Variable Data Report

This report will show lots that exists in the database which have variable data inspection results listed in the "Variable_Data" table. The report can be restricted based on the input query parameters or the "View" settings based on the user privileges.
The following query parameters can be used:
Producer
Product Three Letter Code
Product Class ID
Container Type ID
Process Type ID
Family of Product
Inspection Type ID
Inspected by:
Variable Description
Start Date and End Date of Production
Information displayed for each lot:
Date of Production
Vendor Location Code
Product Three Letter Code
Lot Number + Split Lot ID
Information displayed for each Variable Data Inspection Record
Inspection Date
Inspection Type
Inspected by
Variable Description
Sample Size
Variable Average
Variable Standard Deviation
Unit of Measure

Note: The report will be sorted in the same order as the variables above listed

4.2.7.2.34 Detailed Lot Acceptance Report

This report will show lots that exists in the database that have inspection header records listed in the "Lot Inspection Headers" table, where Inspection Type = "1" or "6". The report can be restricted based on the input query parameters or the "View" settings based on the user privileges.
The following query parameters can be used:
Producer
Product Three Letter Code
Product Class ID
Container Type ID

Process Type ID
Family of Product
Inspection Type ID
Inspected by:
Start Date and End Date of Production
Lot Acceptance
Information displayed for each lot:
Date of Production
Vendor Location Code
Product Three Letter Code
Lot Number + Split Lot ID
Information displayed for each Inspection Record
Inspection Date
Inspection Type
Inspected by
Inspected Quantity
Lot Accepted (Y/N)

Note: The report will be sorted in the same order as the variables above listed

4.2.7.2.35 Detailed Rework Report

This report will show lots that exists in the database which have an inspection header with the inspection_type= "4". The report can be restricted based on the input query parameters or the "View" settings based on the user privileges.
The following query parameters can be used:
Producer
Product Three Letter Code
Product Class ID
Container Type ID
Process Type ID
Family of Product
Inspection Type ID = Rework
Inspected by:
Start Date and End Date of Production
Information displayed for each lot:
Date of Production
Vendor Location Code
Product Three letter Code
Lot Number + Split Lot ID
Information displayed for each End Item Inspection Record
Inspection Date
Inspection Type
Inspected by:
Inspected Quantity

Note: The report will be sorted in the same order as the variables above listed

4.2.7.2.36 Contract Report

This report will show in a report form the contracts header and details as listed in the database.
Input parameters for this report are:
Awarded to:
Contract Number:
Contract Start Date Period: (Begin/End Date)
Information displayed from the contract header:
Contract Number

Awarded to
Research Contract (Y/N)
Information displayed from the contract detail table:
Contract Item
NSN
Product Description
Family of Ration
Min. Quantity
Max. Quantity

Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.37 Product Report

This report will show in a report form the product information as listed in the database.
Input parameters for this report are:
NSN
Product Class
Process Type
Container Type
Information displayed from the product table:
NSN
Product Description
Three Letter Product Code
Label Weight
Unit of Measure
Process Type
Container
Specification
Number in Case
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.38 Product Structure Report

This report will show in a report form the various assembled products and the components that could be used in the assembly. This information is listed in the database "Product_Structure" table.

Input Parameters for this report are:
NSN

Information displayed in this report:
Header:
NSN
Product Description
Detail:
Component NSN
Component Description
Component Quantity

4.2.7.2.39 Lot Activity Report

This report will show in a report form the history of a lot as listed in the database.
Input parameters for this report are:
Production Date Time Frame
NSN
Product Class
Process Type

Container Type
Information displayed from the lot history table:
Product Lot (concatenate lot id and split lot id)
Producer Code
NSN
Activity Date
Activity
Quantity
Performed by:
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.40 Product Class Defect Report

This report will show in a report form the defects that are assigned to a product class as listed in the database.
Input parameters for this report are:
Product Class
Information displayed from the product class defect table:
Product Class
Defect ID
Defect Description
Result Type
Defect Category
Unit of Measure
Analysis Defect ID
Analysis Defect Description

Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.41 Inspection Type/Product Class Defects Report

This report will show in a report form the defects that are assigned to a product class as listed in the database.
Input parameters for this report are:
Product Class
Inspection Type
Result Type
Information displayed in this report:
Header:
Product Class
Detail:
Inspection Type
Defect ID
Defect Description
Result Type
Unit of Measure
Analysis Defect ID

Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.42 USDA Error Report: Lot Header

This report will show in a report form USDA Gen_info error table that lists all the errors that were incured during the import process of the Gen_info information.
Input parameters for this report are:
Producer Code

NSN
Information displayed from this table:
    Company Code
    Error Code
    Error Description
    Lot Code (concatenate lot id and split lot id)
    NSN
    Date Initial Inspection
    Inspection No
    Quantity Inspected
    Lot Passed
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.43 USDA Error Report: Defects

This report will show in a report form USDA Fail_Insp error table that lists all the errors that were incurred during the import process of the Fail Insp information.
Input parameters for this report are:
    Producer Code
    NSN
Information displayed from this table:
    Company Code
    Error Code
    Error Description
    Lot Code (concatenate lot id and split lot id)
    NSN
    Inspection No
    Defect ID
    Number of Defects
    Sample Size
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.44 Organization/Locations Report

This report will show in a report form the organization and its location information as listed in the database.
Input parameters for this report are:
    Organization Name
    Location 3 Letter Code
Information displayed from the organization and location tables:
    Organization Name
    Organization ID
    Location 3 Letter Code
    Location Description
    Address (concatenate address1+address2+address3+city+zip)
    Cage Number
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.45 Organization Products Report

This report will show in a report form the products that were produced by an organization as listed in the database "Organization_Product" table. This report can also be used to show which companies havbe produced a certain product in the past.

Input Parameters for this report are:
    Organization

42

NSN

Information displayed in this report:
Header:
Organization Name
Detail:
NSN
Product description

4.2.7.2.46 Production Lot Report

This report will show in a report form the product lot information as listed in the database.
Input parameters for this report are:
Producer code
Time frame of production
NSN
Product Class
Product Type
Container Type
Contract Number
Contract Line Item
Family of Ration
Information displayed from the product table:
Producer Code
Lot number (concatenate lot id and split lot id)
NSN
Product Description
Packaging Method
Contract Number
Contract Line Item
Lot Quantity
Offered to Government
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.47 Receipt Report

This report will show in a report form the product receipt information as listed in the database.
Input parameters for this report are:
Received by
Receipt time frame
Contract Number
NSN
Product Class
Process Type
Container Type
Information displayed from the product table:
Date Received
Received by
Document Number
Document Type
Document Date
Contract Number
Contract Item
Delivery Order Number

Delivery Line Item
Received Quantity
Comments
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.2.48 Waiver Report

This report will show in a report form the waiver information as listed in the database
Input Parameters for this report are:
Lot Production Time Frame: Start Date and End date
Producer Code
NSN

Information displayed in this report:
Lot Number and Split Lot ID
Producer Code
NSN
Date of waiver
Waiver Granted by
Comments
Note: The report will be sorted in the same order as the output variables above listed

4.2.7.3   Special Requirements

4.3   Inputs-Outputs

4.3.1   USDA

The USDA data records are coming from the following tables in the Operational Ration Database: "GENINFO" and "FAILINSP". These records first put into staging tables "GENINFO" AND FAILINSP" and the records are then used to populate the identified tables and fields in the QDMS database

4.3.1.1   GENINFO

Lot_number	N	LOT_INSPECTION_HEADERS	LOT_NUMBER
Sublot_ID	A10	LOT_INSPECTION_HEADERS	SPLIT_LOT_ID
Company_Code	A3	LOT_INSPECTION_HEADERS	LOCATION_3LETTER CODE
Inspection_Number	A1	LOT_INSPECTION_HEADERS	VENDOR_INSPECTION_ID
NSN_Stock_Number	A16	LOT_INSPECTION_HEADERS	NSN
Quantity_inspected	N	LOT_INSPECTION_HEADERS	LOT_QUANTITY_INSPECTED
Date_init_inspection	D	LOT_INSPECTION_HEADERS	DATE_TIME_INSPECTION
Lot_Accepted	A3	LOT_INSPECTION_HEADERS	LOT_ACCEPTED

4.3.1.2   FAILINSP

Lot_Number	N	*LINK TO "LOT_INSPECTION_HEADERS"*	
Sublot_ID	A10	*LINK TO "LOT_INSPECTION_HEADERS"*	
MRE_number	N	not used	
Company_code	A3	*LINK TO "LOT_INSPECTION_HEADERS"*	
Product_Code	A3	not used	
Defect_reason	A55	not used	
Inspection_Number	A1	*LINK TO "LOT_INSPECTION_HEADERS"*	
Defect_id	A4	END_ITEM_DEFECTS	DEFECT-ID
NSN_Stock_Number	A16	*LINK TO "LOT_INSPECTION_HEADERS"*	
Date-on-hold	D	not used	
Comments	A255	not used	
Date-updated	D	not used	
Date-reinspection	D	not used	
Pass-reinspection	A3	not used	
Quantity_reinspected	N	not used	
Reason-fail-reinspection	A55	not used	
Date-reworked	D	not used	
Number_Defects	N	END_ITEM_DEFECTS	NUMBER_DEFECTS
Sample_Size	N	END_ITEM_DEFECTS	SAMPLE_SIZE_TAKEN

4.3.2   AVI

Lot Header (to be filled out by producer or assembler)
    Producer location code (three letter)
    Product code (NSN)
    Lot Number (Julian Date)
    Split Lot Designator (A/B/C/....) or Sub Lot Code
    Contract Number (DPSC contract number if available)
Inspection Header:
    Source of Data
    Inspection Type (Warranty or End_Item)
    Inspection Date
    Lot Size Inspected

    UOM
    Inspection Result (Accept/Reject or Blank, calculated based on detail)
Inspection Detail:
    Defect Type
    Number of Defective Units
    UOM

4.3.3 Component Producer

4.3.3.1 Production Information (General)

Lot Header:
        Producer location code (three letter code)
        Product code (NSN)
        Lot Number
        Split Lot Designator (A/B/C/....)
        Contract Number (DPSC contract number if available)
        Contract Item
        Date of production
        Lot Quantity
        Offered to Government (Y/N)
        Package Material Type (HFFS/VFFS/NA)
Sub Lot Detail: (Optional)
        Sub Lot ID
        Sub Lot Quantity

4.3.3.2 In-Process Inspection

Inspection Header:
        Producer location code (three letter code)
        Product code (NSN)
        Lot Number
        Split Lot Designator (A/B/C/....)
        Inspected By (Company Code)
        Inspection Type (Pre or Post Retort or Rework)
        Inspection Date
        Lot Size Inspected
        Comments
In_Process_Inspection Attribute Detail:
        Defect ID
        Number of defective units removed
        Comments
Production Variable Detail
        Variable ID
        Number of samples taken
        Average
        Standard Deviation
        Result UOM
        Comments

4.3.3.3 End Item Inspection

Inspection Header:
        Producer location code (three letter code)
        Product code (NSN)
        Lot Number
        Split Lot Designator (A/B/C/....)
        Inspected By (Company Code)
        Inspection Type (End_Item)
        Inspection Date

          Lot Size Inspected
          Lot Accepted (Yes/No)
          Comments
End Item Inspection Detail:
          Defect ID
          Sample Size
          Number of Defective Units
          Comments

4.3.4 Assembler

4.3.4.1 Receipt Information

Receipt Header:
          Document Number
          Date Shipped
          Date Received
          Document Date
          Document Type (DD250/BL)
          Received By
Receipt Detail:
          Lot Number
          Split Lot Identifier
          Material Code (NSN)
          Producer Location Code (three letter code)
          Document Number
          Contract No
          Contract Item
          Delivery Order
          Delivery Line Item
          Quantity Received
          Comments

4.3.4.2 Receipt Inspection

Inspection Header:
          Producer location code (three letter code)
          Product code (NSN)
          Lot Number
          Split Lot Designator (A/B/C....)
          Inspected By
          Inspection Type (End_Item)
          Inspection Date
          Lot Size Inspected
          Lot Accepted (Yes/No)
          Comments
Receipt Inspection Detail:
          Defect ID
          Sample Size
          Number of Defective Units
          Comments

4.3.4.3 Assembly Information (General)

Lot Header:
  Assembler location code (three letter code)
  Product code (NSN)
  Lot Number
  Split Lot Designator (A/B/C/....)
  Contract Number (DPSC contract number if available)
  Contract Item
  Date of Production
  Lot Quantity
  Offered to Government (Y/N)
  Package Material Type (HFFS/VFFS/NA)
Component Detail:
  Component Producer location code (three letter code)
  Component Product code (NSN)
  Component Lot Number
  Component Split Lot Designator (A/B/C/....)
  Component Lot Quantity Used

4.3.4.4 Assembly In-Process Inspection

Inspection Header:
  (inspection header will refer to the component lot that is inspected on-line)
  Producer location code (three letter code)
  Product code (NSN)
  Lot Number
  Split Lot Designator (A/B/C/....)
  Inspected By
  Inspection Type (In_Process_Assembly)
  Inspection Date
  Lot Size Inspected
  Comments
In_Process_Inspection Attribute Detail:
  Defect ID
  Number of defective units removed
  Comments

4.3.4.5 End Item Inspection

Inspection Header:
  (inspection header will refer to the assembled lot)
  Producer location code (three letter code)
  Product code (NSN)
  Lot Number
  Split Lot Designator (A/B/C/....)
  Inspected By
  Inspection Type (End Item)
  Inspection Date
  Lot Size Inspected
  Comments
End Item Inspection Detail:
  Defect ID
  Sample Size
  Number of Defective Units

Comments

4.3.5 DPSC

4.3.5.1 Contract Information

Contract Header
    Contract Number
    Grantor Organization
    Grantee Organization
    Contract Start Date
    Contract End Date
    Research Contract (Y/N)
Contract Line Items:
    Product ID (NSN)
    Family of Ration Type
    Minimum Qty
    Maximum Qty
Contract Delivery Schedule:
    Quantity Required
    UOM
    Date Required
    Destination
    Delivery Order No.

4.3.5.2 Product Setup Information

Products
    NSN Number
    Product Code
    Product Description
    Product Specification Number
    Components Product (Y/N)
    Product Class
    Process ID
    Container Type
    Label Weight
    Weight_UOM
    Number per Case
    Units_of_Contract
Product_Structure
    NSN
    Component_NSN
    Component_Quantity

4.3.5.3 Company Setup Information

Organization Locations
    Location 3 Letter Code
    Organization Location Description
    Organization_ID
    Cage_No
    Address1
    Address2
    Address3
    City
    State
    Zip

4.3.5.4 Inspection Setup Information

Inspection Defects
        Defect_ID
        Defect_Description
        Result_Type (IP_Attribute; IP_Variable; FP_Attribute)
Product_Class_Defects
        Product_Class
        Defect_ID
        Defect_Catagory
        UOM_ABR
Inspection_Defect_Xref
        Inspection_Type
        Defect_ID

4.3.5.5 Waiver

        Waiver ID
        Date of Waiver
        Person Name who granted Waiver
        Producer location code (three letter)
        Product code (NSN)
        Lot Number (Julian Date)
        Split Lot Designator (A/B/C/....)
        Comments

4.4 Data Characteristics

No information on data characteristics and expected growth of the data and related components was provided by the client at this time

4.5 Failure Contingencies

4.5.1 Back-up

It will be the responsibility of the database administers to perform database backups at a frequency that minimizes any data loss.

4.5.2 Fallback

In case of a system failure, no new records can be added to the database. The USDA and AVI data will be maintained at their respective facilities and can be requested as an hard copy. Any attempt to enter data via a Web page will fail until the system is up. Until such time, the providers of "web-page-data" should maintain their own records so that DPSC can call upon them to provide this data manually.

4.5.3 Restart

In case of a system restart, the system will acquire any data that was "buffered" by the USDA and AVI. The providers of web-page-data" will need to be notified that the system was re-enabled so that they can enter their data as soon as possible.

4.6 Design Requirements

All design and analysis will be done with the use of Oracle 2000 products. These products allow for database designs that are functional in architecture and modular design and standard end user interfaces.

Interface program will be written in Visual Basic whenever possible, using standard control objects.

4.7 Computer Security Requirements

Access to any data in the database must require a login and password.

4.8 Human Performance Requirements

No specific human performance requirements were specified by the client.

5 TEST AND QUALIFICATION REQUIREMENTS

5.1 Introduction

In order to test the software, all hardware and supporting software should be installed as identified in previous sections. Unit tests will be performed in house as various modules are completed. The intergation tests will be performed at DPSC after all the hardware and software is installed. and prior to the Software System DT&E test

Table: Test and Qualification Cross-Reference Index

SRS Par Ref	Requirement Name	Qualification Method	Qualification Test Level	SRS Section 5 Par Ref
4.2.1.3	File Transfer USDA	T	1	5.1.1.1
4.2.1.4	Data Upload USDA	T	1	5.1.1.2
4.2.2	RecordTransfer AVI	T	1	5.1.1.3
4.2.3	Data Upload AVI	T	1	5.1.1.4
4.2.6	Forms	I	1	5.1.1.5
4.2.6	Web Sever (Unit Test)	I	1	5.1.1.6
3.2	Dynamic Web Page Generator	I	1	5.1.1.7
3.2	Graphics Engine via Web Page	I	1	5.1.1.8
3.2	Report Writer via Web Page	I	1	5.1.1.9
4.2.1.5	Normalization USDA	T	2	5.1.2.1
4.2.1.5	USDA Data Correction	I	2	5.1.2.2
4.2.4	Monitor/Record Change Flag	T	2	5.1.2.3
4.2.5	Integrator/DWH Update	T	2	5.1.2.4
4.2.6.2.24	Production Defect Analysis	T	3	5.1.3.1
4.2.6.2.25	Production Variable Data Analysis	T	3	5.1.3.2
4.2.6.2.26	Rework Analysis	T	3	5.1.3.3
4.2.6.2.27	Defective Lot Analysis	T	3	5.1.3.4
4.2.6.2.28	Defect Analysis	T	3	5.1.3.5
4.2.6.2.29	Trend Analysis	T	3	5.1.3.6
3.3	Intranet Access	I	4	5.1.4.1
3.3	Internet Access	I	4	5.1.4.2
3.2	Web Server (System Test)	I	4	5.1.4.3
4.29	Dynamic Web Access	I	4	5.1.4.4
3.4	User Access/Privelage	I	4	5.1.4.5
3.4	Security	I	4	5.1.4.6
4.21	Access to USDA database	I	4	5.1.4.7
4.21	Access to AVI database	I	4	5.1.4.8

Qualification Method:
- I: Inspection
- A: Analysis
- D: Detection
- T: Review of Test Data
- O: Other

Qualification Test Level:
- 1 Unit Test
- 2. Integration
- 3. Software DT&E
- 4. System DT&E
- 5. Other

5.1.1   Unit Tests

5.1.1.1   File Transfer USDA

A test will be conducted spread over a three day period that will login to the USDA database on each consecutive day and retrieve any files that are not existing locally. The test is successful if after each day, the files stored locally are identical in size and date to the files stored on the USDA bulletin board.

5.1.1.2   Data Upload USDA

A test will be conducted that will run over a three day period that will check for new files and upload the records to designated tables in the Oracle database. The test is successful if after each upload the records in the Oracle tables are identical to the records in the file that were uploaded.

5.1.1.3   Record Transfer AVI

A test will be conducted spread over a three day period that will login to the AVI database on each consecutive day and retrieve any records that are not existing locally. The test is successful if after each day, the records stored locally are identical to the records stored on the AVI Lotus Notes database.

5.1.1.4   Data Upload AVI

A test will be conducted that will run over a three day period that will check for new records and upload the records to designated tables in the Oracle database. The test is successful if after each upload the records in the Oracle tables are identical to the records in the AVI Lotus Notes database.

5.1.1.5   Forms

Each data entry form will be tested to assure that the entered data is checked for consistency and entered in the appropriate database table. The test is successful if the application detects erroneous or missing information and the entered data can be identified in the underlying database table.

Web page data entry by producers and assemblers will be tested.
Data warehouse functionality will be tested as new data records are added to the system
Data correction functionality will be tested by adding erroneous data sets to the database

5.1.1.6   Web Server

As soon as the required software is installed a test will be conducted to assure that the web server can generate a dynamic web page, using example supplied by the vendor

5.1.1.7   Dynamic Web Page Generator

As soon as the required software is installed a test will be conducted to assure that the web server can generate a dynamic web page, using forms from the QDMS application

5.1.1.8   Graphics Engine via Web Page

As soon as the required software is installed a test will be conducted to assure that the graphics engine works over the web server and can be displayed as a dynamic web page, using graphs from the QDMS application

5.1.1.9   Report Writer via Web Page

As soon as the required software is installed a test will be conducted to assure that the web server supports the report writer to send reports as "pdf" files using reports from the QDMS application

5.1.2 Integration Tests

5.1.2.1 Data Normalization USDA

After the design of the database is completed and the applications to import the USDA data, a test will be conducted to verify the functionality of the Data Normalization procedures for USDA data. For this purpose, some records will be generated as USDA data. The test will be successful if these records are automatically uploaded into the normalized production tables.

5.1.2.2 USDA Data Correction

The data correction module will be tested after erroneous USDA records were entered in the database. The test will be successful after the records were correctly identified, displayed to the user where the error is and successfully imports the record after the correction has been made. A similar test should be done on a record with an incorrect correction. This record should be not be committed to the database but re-identified as erroneous.

5.1.2.3 Monitor

The Monitor module will be tested after new records have been added to one or more normalized production tables. The Monitor will generate a record that will flag these records as New/Changed/Deleted. The test will be successful if the Monitor correctly changes the flag "Record_Update" to either "Y" or notes in the "Deletion_Log" table which record was deleted.

5.1.2.4 Integrator

The Integrator will be tested after the Monitor has identified that records were added/changed/deleted in the normalized production tables. The test will be successful if the new or edited or deleted records propagate up into the data warehouse tables

5.1.3 Software System DT&E Tests

5.1.3.1 Production Analysis

The production analysis tool will be tested after the previous test have successfully been completed and data resides in the various database tables from various sources. The parameters of the analysis will be set such that a know set of data is being analyzed. The results of the analysis will be verified by comparing the results against a manual calculation

5.1.3.2 Defective Lot Analysis

The defective lot analysis tool will be tested after the previous test have successfully been completed and data resides in the various database tables from various sources. The parameters of the analysis will be set such that a know set of data is being analyzed. The results of the analysis will be verified by comparing the results against a manual calculation

5.1.3.3 Defect Analysis

The defect analysis tool will be tested after the previous test have successfully been completed and data resides in the various database tables from various sources. The parameters of the analysis will be set such that a know set of data is being analyzed. The results of the analysis will be verified by comparing the results against a manual calculation

5.1.3.4 Trend Analysis

The trend analysis tool will be tested after the previous test have successfully been completed and data resides in the various database tables from various sources. The parameters of the analysis will be set such that a know set of data is being analyzed. The results of the analysis will be verified by comparing the results against a manual calculation

5.1.4 System DT&E

The system DT&E of the hardware is not anticipated as the hardware procurement falls outside the scope of this project.

5.1.4.1 Intranet Access

Once the hardware and software is installed at DPSC a general test will be performed to confirm that the web server hardware is part of the Intranet

5.1.4.2 Internet Access

Once the hardware and software is installed at DPSC a general test will be performed to confirm that the web server hardware is part of the Internet

5.1.4.3 Web Server

Once the hardware and software is installed at DPSC a general test will be performed to confirm that the web server hardware is operational and has access to the database

5.1.4.4 Dynamic Web Page Generator

After completion of the Web Server test, all forms will accessed from a client network station using either Netscape or Explorer as the web browser. This test will be successful if all forms show as designed

5.1.4.5 User Access/Privilege

After confirming the functionality of all forms, graphs and reports, specific queries will be run under different user names and passwords. This test will be successfully completed after it is demonstrated that the data displayed is limited based on the access privileges.

5.1.4.6 Security

A test will be made to assure that the database can not be accessed via a common ODBC interface using TCP/IP protocol and one of the know user login/password

5.1.4.7 Access to the USDA database

A system test will be performed that the hardware installed supports the requirements of the software to gain access to the USDA bulletin board to download new files

5.1.4.8 Access to the AVI database

A system test will be performed to assure that the AVI application has access to the QDMS database and is capable to download and upload new records

5.1.5 Test Requirements

Data from the USDA, AVI, Producers and Assemblers should be available
All hardware and support software needs to be available, including any network access as specified in this document.

6 NOTES

Appendix B
Quality Data Management System: User Manual for Administration and Maintenance

Quality Data Management System

User Manual for:

QDMS Administration

And

Maintenance

February 1999

Index

1. Introduction

The Quality Data Management System (QDMS) is a data base application that stores pertinent inspection information from producer, assemblers as well as government inspection agencies. The application includes analysis tools that allow the user to analyze for defect trends as function of time, product or producer.

2. Users

The QDMS has various users groups such as producer data entry clerks, producer analysts, DPSC analyst, DPSC maintenance, and Database Administrator. Each user of the system will be assigned an unique login name and password and will be assigned to one of these user groups and a company location codes. The data available to each user is restricted based on a user privileges. For example, a user working for company A can only see data that either belongs to lots produced by company A or was entered by company location code A. Users assigned to one of the DPSC user groups have access to all producer data

3. System Requirements

The QDMS application can either be deployed as a client server application or as a web browser application. A client server deployment requires that Oracle Developer 2000 runtime files are installed on the client machine. Also the pointers in the registry should point to the application files on the file server. A web browser deployment requires the installation of a web browser on the client machine that is 100% compliant with Java JDK 1.1 standard. Acrobat reader, version 3.0 or higher is required to print reports with a web browser deployment. Figures in this manual were generated from a client server machine.

At this time neither the Microsoft nor the Netscape browser JVM is 100% compliant with the JDK 1.1 standard. Oracle has released a Java browser add-in (JINIT1.1.5.3) that can be used instead of the browsers default JVM. This add-in can be downloaded automatically from the QDMS site when the appropriate URL is selected for the first time, or can be manually downloaded from the following Oracle site at this particular time:
> http://www.oracle.com/products/tools/dev2k/dn.html

Note: Oracle might upgrade the version of their JINIT from time to time. Newer version might require a different html page for login.

Hardware Requirements:
> Computer: Pentium Processor 100 MHz or higher
> Operating System: Windows 95 or Windows NT
> Memory: 32 MB
> Hard Disk Space: 20 MB free

4. Accessing QDMS Application

To run the application as a client server application the following property should be specified as the target: D:\ORANT\BIN\F45RUN32.EXE Q:\Fmx\Vesstart.fmx. The Q drive needs to be mapped to the file server where the application files are stored.

To login into the QDMS system via a Web Browser, the users at DSCP should use the following URL:
> http://dscp121.dscp.dla.mil/web_html/qdmsj3.html

To login into the QDMS system via a Web Browser, the external user, outside the DSCP network should use the following URL:
> http://www2.dscp121.dscp.dla.mil/web_html/qdmsj3w.html

After the server has been found, the Web Server will down load various applets to the client machine before the QDMS application can start. This time to download the applets is a function of the network speed and can between 30 seconds (internal connection) to 300 seconds (28,800 baud modem speed). Once all applets are downloaded the QDMS application will start by displaying a login screen. Oracles JVM will cache this information to the hard disk. After the initial files have been cached, subsequent logins should be significant faster.

(typical path: C:\program files\oracle\jinitiator\jcache\xxxxxxxx.jc. In case this file becomes corrupt subsequent login's might fail which would require the user to delete this file and download and download a new copy).

A login screen as shown below appears where the **username ,password** and **database** has to be entered to start the application. This will be assigned by the Database Administrator and must be maintained in strict confidence. It should be noted that the login id's are specific to the users location, therefore an organization with multiple production locations should request login ID's for each location.

Figure 1

5
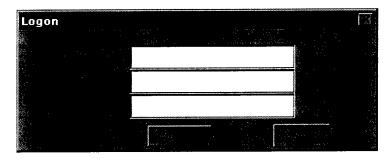
5. QDMS Menu Structure and Navigation Commands

5.1. Menu Structure

After the user has logged into the system, the main menu screen will be displayed. Each user type has been granted distinct privileges and data views in the QDMS application. Therefore the menu displayed will be different for each user group. The Startup window of the QDMS application for the administrator looks as in Figure 2
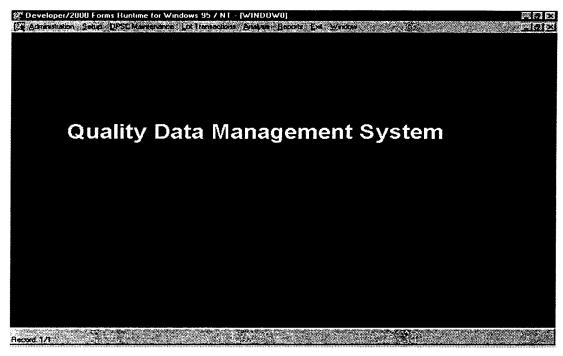
Figure 2

The Main Menu of the application has the following structure:

Administration
        Users
        Error Codes
        Lot Activity Types

Setup
        Products
        Defect Types
        Inspection Types and their Defects
        Product Classes and their Defects

Organizations
Organizations and Locations
Analysis Defects
Container Types
Family of Rations
Packaging Methods
Product Processes
Units of Measure

DPSC Maintenance
Contracts
Waivers
Geninfo Errors
Failinsp Errors
Update Data Warehouse

Lot Transactionns
Create Lots
Lot Receipts
Finished Lot/Receipt Inspections
Inprocess Inspections

Analysis
Data Analysis
Production Analysis
Inspection Defect Analysis
Variable Defect Analysis
Defective Lot Analysis
Rework Analysis
Trend Analysis
Production Trend
Inspection Defect Trend
Variable Defect Trend
Defective Lot Trend
Rework Trend
Analysis Reporting
Production Detail
Inspection Defect Detail
Variable Defect Detail
Defective Lot Detail
Rework Detail

Reports
Contracts
Products
Product Structure
Lot Activity
Product Class/Defects
USDA Inspection Header Errors
USDA Inspection Detail Errors
Organizations/Locations
Organization/Products
Production Lot
Receipts
Waivers

Exit

      Exit Form

5.2. Navigation Menu and Commands

In the following sections each of the forms in this application will be discussed in detail and its functionality given. The following menu can be seen on top of all forms.

Action	Edit	Block	Field	Record	Query	Window	Help
Clear all	Cut	Previous	Previous	Previous	Enter	Cascade	Help
Save	Copy	Next	Next	Next	Execute	Tile	Keys
Print	Paste	Clear	Clear	Scroll up	Last Criteria	Arrange Icons	List
Exit	Edit		Duplicate	Scroll Down	Cancel		Display Error
				Clear	Count Hits		
				Remove	Fetch next set		
				Insert			
				Duplicate			
				Lock			

Each of the navigation tasks can also be performed by the following short cut key sequence:

Oracle Navigation Short Cut Key Commands:

Navigation	Browser Command	Client/Server Command
clear block	F7	<shift>+F5
clear field	F5	<ctrl>+u
clear form	F8	<shift>+F7
clear record	F6	<shift>+F4
commit record	<ctrl>+S	F10
count query	F12	<shift>+F2
delete record	<ctrl>+up	<shift>+F6
display error	<shift>+<ctrl>+E	<shift>+F1
down	<down>	<down>
duplicate field	<shift>+F5	F3
duplicate record	<shift>+F6	F4
edit	<ctrl>+E	<ctrl>+E
enter query	F11	F7
exit	F4	<ctrl>+Q
insert record	<ctrl>+<down>	F6
list of values	<ctrl>+L	F9
next block	<shift>+<page down>	<ctrl>+<page down>
next field	<tab>	<tab>
next record	<down>	<shift>+<down>
next set of records	<shift>+F8	<ctrl>+>
previous block	<shift>+<page up>	<ctrl>+<page up>
previous field	<shift>+<tab>	<shift>+<tab>
previous record	<up>	<shift>+<up>
print	<ctrl>+	<shift>+F8
return	<Return>	<Return>
scroll down	<page down>	<page down>
scroll up	<page up>	<page up>
show keys	<ctrl>+K	<ctrl>+F1
toggle query mode	<ctrl>+B	F5
up	<up>	<up>
update record	<ctrl>+U	F10

5.3. Definitions and Basic Database Functionality

Throughout this document a table has been used to illustrate the different properties of the fields in each of the forms. The different columns in the table are Field, Datatype, Required. LOV, Navigable, Insert, Query and Update. Barring the first two columns all other columns have a Y or no entry for the different items.

FIELD   This specifies the name of the **item** in the form.

DATATYPE   This specifies the **datatype** of the items (Character, Number, Date being the different entries).

REQUIRED   This column will either have an Y or no entry. If it is Y, it means that the column under question **has to be entered** while inserting a record The user will not be allowed to leave the field without entering a value.

LOV   This column will either have a Y or no entry. If it is Y , it means a list of allowable list of values (LOV) **exists** which can be used while inserting a record.

NAVIGABLE   As in the previous columns this column too will either have an Y or no entry. If there is no entry it means the field is not navigable. This means that the cursor does not move to that field for the user to do **anything**.

INSERT   If this column has entry Y then while inserting a record this field **can be entered** by the user as opposed to being populated by the system under certain conditions.

QUERY   If this column has entry Y then one can a enter a value in the field and **query**.

UPDATE   If this column has an entry Y then the values in that column **can be changed** by the user at any time if need be. Otherwise the user will not be allowed to change the value in that field once the records are committed to the database.

If LOV exists for a field as seen from the table, click on the field and click on List under Help in the menu. Then a list is displayed for choosing the desired value. For example in the Users form the following LOV is displayed for the field Works At. This functionality helps data entry to be faster and easier. If an LOV exists for a field and the user tries to enter a value which is not in the LOV the system will not allow the user to do that.
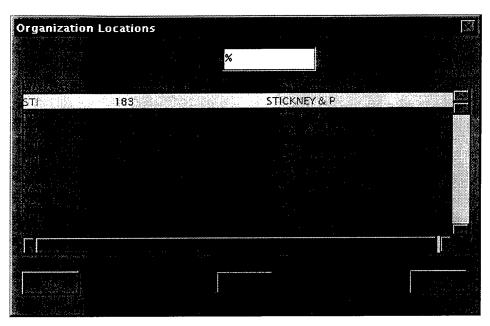
Figure 3

1. To query, select Enter under Query. Then enter the value for the query and execute the query. The query will be executed and the relevant values will be displayed.

2. To insert a record, enter the values for all the fields starting from the first. Then select Insert under Record .In the table for each form there is a column named insert. If the entry for any field under this column is Y that means a value can be entered for that field while inserting a record. If not, user will not be allowed to enter a value for that field.

3. To remove a record , select the record and select Remove under Record

4. If a user wants to use the keyboard in the place of mouse, when Keys under Help menu is clicked, a window opens up showing the different keys used for different actions like entering a query, executing a query, displaying an LOV and so on. The window looks like the following;
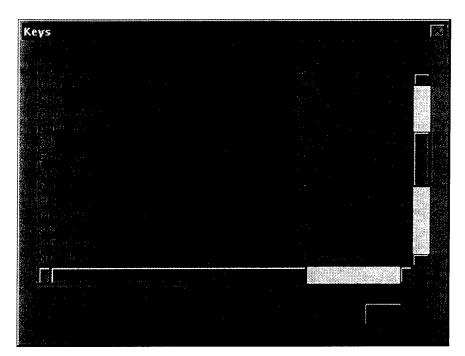
Figure 4

5. Whenever a field is mentioned to have a poplist the field is as shown in the following figure. The field Research Contract and Contract Expired are fields with poplist. When clicked on the arrow a list of allowable values pops up. As seen in the figure when clicked on the arrow in the field Research Contract both Yes and No show up.
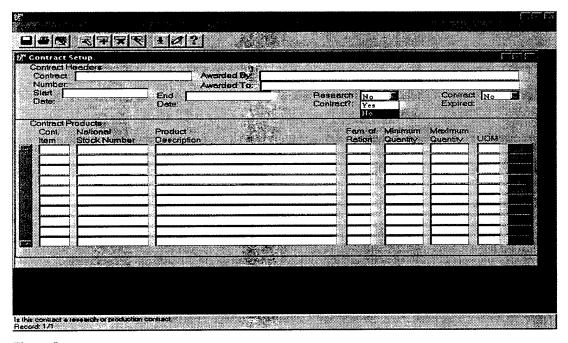
Figure 5

6. QDMS Functionality

In this section, we will review the functionality of each form. Accompanying referenced screen captures are included in the appendix.

6.1. Administration

The forms under Adminstration are only for Database/System Administration purposes.
These forms should only be used by the DBA.

Users

The purpose of this form is to relate a *User* with the company he/she works for. This is important for implementing security. This is a single block form as shown in fig 6
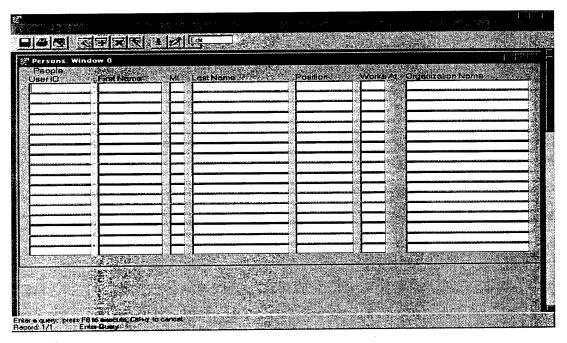
Figure 6

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
User Id	Char	Y		Y	Y	Y	
First Name	Char			Y	Y	Y	Y
MI	Char			Y	Y	Y	Y
Last Name	Char			Y	Y	Y	Y
Position	Char			Y	Y	Y	Y
Works At	Char	Y	Y	Y	Y	Y	Y
Organization Name	Char						

It can be noticed that when the "Works At" field is populated from the LOV, the corresponding Organization Name gets populated.

6.1.1. Error Code

The purpose of this form is to relate error codes used in validating USDA records with their associated text descriptions. The descriptions help in understanding errors that need to be fixed. This is a single block form as shown in fig 7.
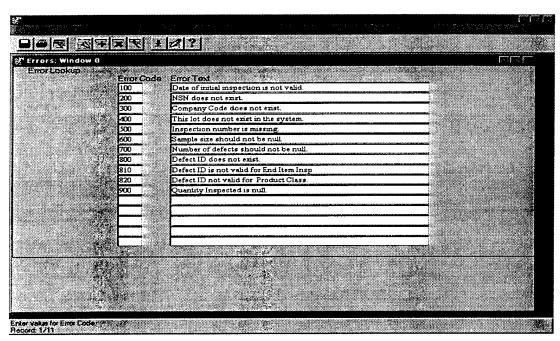
Figure 7

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
ErrorCode	Number	Y		Y	Y	Y	
Error Text	Char			Y	Y	Y	Y

6.1.2. Lot Activity Types

This is a single block form as shown in fig 8. This form is mainly for defining different Lot Activities. This in turn helps to relate codes in lot_histories table to their corresponding text description. The lot_activity_id field cannot be updated.

13

Figure 8

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Lot Activity Id	Number	Y		Y	Y	Y	
Activity Description	Char			Y	Y	Y	Y

6.2. Setup

Setup forms are used to define data that will be used in validating all production data entered or imported into the application. Setup data needs to be periodically updated, as they are like the building blocks of the application. The QDMS Administrator and DPSC Maintenance have access to these forms.

6.2.1. Products

Products form is a setup form to enter product description into the system. The key field in this form is the National Stock Number (NSN). This form consists of two blocks, one for the products and the other for the product structure. The form is as shown in the fig.9.

Figure 9

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
3letter_code	Char	Y		Y	Y	Y	
NSN	Char	Y		Y	Y	Y	
Product Description	Char	Y		Y	Y	Y	Y
Component Product	Char	Y		Y	Y	Y	Y
Product Specification No	Char			Y	Y	Y	Y
Number Per Case	Number			Y	Y	Y	Y
Label Weight	Number			Y	Y	Y	Y
Weight UOM	Char		Y	Y	Y	Y	Y
Unit Of Contract	Char		Y	Y	Y	Y	Y
Process Id	Char			Y	Y	Y	Y
Process Description	Char						
Product Class Description	Char	Y	Y	Y	Y	Y	Y
Container Type Desc	Char	Y	Y	Y	Y	Y	Y

15

3letter_code: The field is used to reference a product description to a three letter code. The three letter code does not have to be unique

NSN: Each product has to be uniquely identified by it's National Stock Number.

Product Description: Each product should be described in less than 60 characters. Preferable the same description should be used as stated in the NSN catalog.

Component Product: This field is used to identify if this product is used as an component in an assembly

Product Specification No: This field is used to record the specification number that pertains to this product

Number per case: This field is used to record how may units of this product are packed in a case

Label weight: This field is used to identify the label weight of this product

Weight UOM: This field identifies the unit of measure for the label weight. Only unit of measures can be used that exist in the UOM table.

Unit of Contract: This field identifies the unit of measure for this product. All lot, receipt and inspection data references this unit of measure to assure that data is recorded in a consistent manner.

Process Id and Process Description: the process id field is used to cross-reference the product to a specific sub category. This field is used during data analysis based on the sub category. Only process Id's can be used that exist in the process table.

Product Class Description: The product class description is used to assign the product to a specific class of product, which have distinct defect characteristics. Only product classes can be used that exist in the product class table. This field is also used during data analysis where products can be analyzed based on this category.

Container Type Description: The Container Type Description is used to identify a specific container to this product. Only container sizes can be used that exist in the container type table. This field is also used during data analysis where products can be analyzed based on a specific container.

The Product Structure block will be used to identify the association between components and assemblies. This data will be used during the product lot data entry to validate component products being made for assembly Contract Items.

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
NSN	Char	Y	Y	Y	Y	Y	
Code	Char					Y	
Product Description	Char					Y	
Label Wgt	Number					Y	
UOM	Char					Y	
Component Quantity	Number	Y		Y	Y	Y	Y

Records can be entered in this block only for records existing in the products block. In the LOV for NSN all the 4 fields that follow the NSN are also included. Once the NSN is selected all the fields except

Component Quantity are populated. In the Component quantity field one can identify how many of these components are used in the assembly of the main product.

6.2.2. Defect Types

The form Defect Types is a one-block form as shown in fig 10 and manages the master list of all the various defects used by the QDMS system. Each defect description is assigned a defect id and a result type. The system recognized four result types: FP_Attribute, FP_Variable, IP_Attribute and IP_Variable, where FP stands for finished product and IP stands for In-Process.

Also in this form each inspection defect can be associated to a higher level defect: "analysis defect". This feature is not yet enabled but would allow one to combine a group of defects and perform a higher level analysis. The analysis defects are assigned in the "analysis defect" form. For example all product defects could be assigned to the analysis defect: "product" which would include drain weight, net weight, fat, salt, etc kind of defects.

When a request is made to add a defect to the system for a certain product at a specific inspection location the following procedure should be followed:
- Determine the result type of the defect and investigate if the defect already exists in the master defect list.
- If the defect does not exist, assign a unique defect id number to the defect description using the following numbering convention:

FP_Attribute:	1000 – 3999
FP_Variable:	4000 – 4999
IP_Variable:	5000 – 5999
IP_Attribute:	7000 – 7999

- Once the defect is part of the master defect list, the next step is to assign the defect to the product class and inspection type using the instructions listed in following two sections.

Defect Id	Defect Description	Result Type	Analysis Defect	Description
1001	ABBERRATION	FP Attribute	210	OTHER
1010	ABRASION	FP Attribute	110	POUCH
1020	ANALYTICAL TESTING	FP Attribute	100	PRODUCT
1021	ASH ANALYSIS	FP Attribute	100	PRODUCT
1022	FAT ANALYSIS	FP Attribute	100	PRODUCT
1023	MOISTURE ANALYSIS	FP Attribute	100	PRODUCT
1024	PROTEIN ANALYSIS	FP Attribute	100	PRODUCT
1025	SALT ANALYSIS	FP Attribute	100	PRODUCT
1030	BONE	FP Attribute	100	PRODUCT
1040	BROKEN ITEM	FP Attribute	100	PRODUCT
1050	CAN CLOSURE	FP Attribute	120	CAN
1053	CAN CONDITION EXTERIOR	FP Attribute	120	CAN
1055	CANDY OVERWRAP	FP Attribute	130	COMPONENT
1056	CAN CONDITION INTERIOR	FP Attribute	120	CAN
1060	CASE PACKING	FP Attribute	140	CASE
1070	COAGULATION	FP Attribute	100	PRODUCT
1080	COATING	FP Attribute	100	PRODUCT

A unique identifier for the procedure.
Record 1/?

Figure 10

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Defect Id	Char	Y		Y	Y	Y	
Defect Description	Char	Y		Y	Y	Y	Y
Result Type	Char			Y	Y	Y	Y
Analysis Defect	Char		Y	Y	Y	Y	Y
Defect Description	Char					Y	

When the entry for the field Analysis Defect is selected from the LOV the Defect Description also gets populated. The user cannot enter a value in this field since the description appears when a cross-reference is made to the list of defects associated with DPSC.

6.2.3. Inspection Types and their Defects

The form Inspection Type and their Defects consist of 2 blocks, and is used to identify the various Inspections that can be performed and that are distinct different from other inspections. While new inspection types can be assigned it is not recommended that this is performed without consultation with the data base administrator. The reason for this is that the data from various inspections flow to different tables and this integrity, which is hard coded, needs to be maintained. By default each new added inspection type will store it's data as In-Process Inspection data. The first block Inspection Stages gives the Inspection Type and the description. The second block Inspection_Defect_Xref is used to assign inspection defects to each inspection type. The defects that are identified in this list are all defects that can be identified at this inspection type regardless of the product or product class. This information is used to restrict the list of values when defects are recorded. Only defect can be assigned that are listed in the inspection defect table. Care should be taken during this process. Sometimes the same defect description is listed but each has a different result type. Defects with the result type FP_Attribute and FP_Variable should only be used for end item or receipt inspections. IP_Attribute and IP_Variable should only be used for Inspections such as pre retort, post retort, rework and assembly inspection. The form looks like fig 11.

Figure 11

The formats of the fields in the first block are :
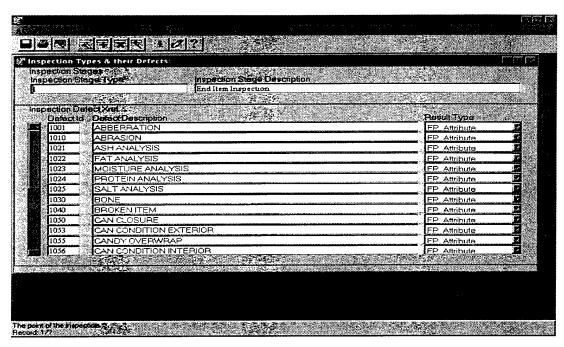
FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Inspection Type	Char	Y		Y	Y	Y	
Inspection Type Desc	Char			Y	Y	Y	Y

The user can query on either field and can enter any inspection type and the corresponding description into this block.

The formats of the fields in the second block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Defect Id	Char	Y	Y	Y	Y	Y	
Defect Description	Char					Y	
Result Type	Char					Y	

The records are always inserted in this block only in the context of the first block. All 3 fields are included in the LOV. When a selection is made for the Defect_Id all the fields are populated. Users are not allowed to enter any data in these fields.

6.2.4. Product Classes and their Defects

The form Product Classes and their defects consists of 2 blocks and is used to associate a list of defects to each product class. This list should include all defects that can be found in this product regardless of the inspection type. The form looks like fig 12.

Figure 12

The formats of the fields in this block are :
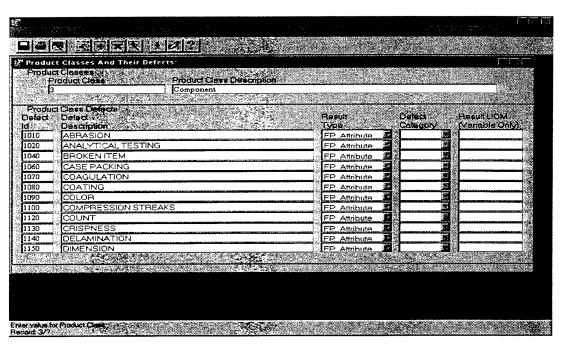
FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Product Class	Char	Y		Y	Y	Y	
Product Class Desc	Char			Y	Y	Y	Y

The next block lists the defects for each Product Class. This block is used to assign defects from the master defect list to each product class and also to assign a defect category (critical, major, minor) and unit of measure (only for variable data) to each of these defect descriptions.

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Defect Id	Char	Y	Y	Y	Y	Y	
Defect Description	Char					Y	
Result Type	Char					Y	
Defect Category	Char			Y	Y	Y	Y
UOM	Char		Y	Y	Y	Y	Y

These records can be queried or entered only in the context of the product class. Once the Defect_Id is selected from the LOV the next 2 fields are also selected as they are part of the LOV. Defect Category is a poplist giving the different categories.

20

6.2.5. Organizations

The form Organizations is a single block form to enter the organization name and generate their Id's for future use in this application. Organizations are required as Contracts are awarded to Organizations. The form looks like fig 13
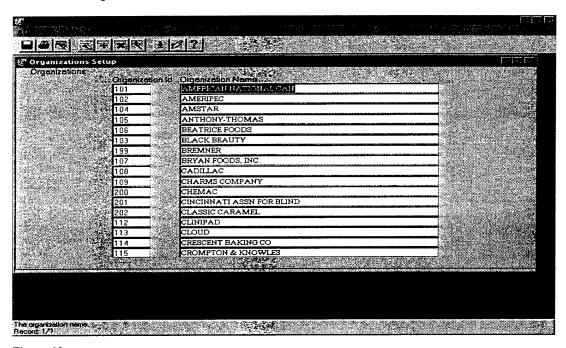
Figure 13

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Organization Id	Char					Y	
Organization Name	Char	Y		Y	Y	Y	Y

When the Organization Names are entered the system generates the Organization Id. When queried the complete list of Organization Names and their Ids can be displayed.

6.2.6. Organizations and Locations

Organizations and Locations is a single block form as in fig 14. This form is used to identify the various locations that can be identified for this organization. Lot information and inspection information will be associated to the location of the organization that generated the data. The form is also used to define if a location is "online". The "online" describes if a location is required to submit data to the QDMS system. This feature is used while importing USDA inspection records. If an organization location is "on-line" no inspection data from the USDA will be accepted unless a producer or assembler enters the lot information. If the location is off-line then the submission of the USDA inspection record will result in the creation of the required lot information and subsequently import of their inspection data. Each location needs to be assigned a unique three-letter location code. This code is used to uniquely identify lot and inspection information. The use of this code across all organizations is essential to the functionality of the system.

Figure 14

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Location 3letter Code	Char	Y		Y	Y	Y	Y
Location Description	Char			Y	Y	Y	Y
Organization Name	Char	Y	Y	Y	Y	Y	Y
Address1	Char			Y	Y	Y	Y
Address2	Char			Y	Y	Y	Y
Address3	Char			Y	Y	Y	Y
City	Char			Y	Y	Y	Y
State	Char			Y	Y	Y	Y
Online User	Char			Y	Y	Y	Y

6.2.7. Analysis Defects

Analysis Defects is again a single block form as in fig: 15. This form will be used to identify various higher level defects that can be used by this application during the data analysis. Analysis Defects are meant to perform higher-level defect analysis in which various defects are grouped together. This form is used to enter the Analysis defects and their description. Once these higher level defects have been established then an association can be made using the defect type form.

22

Figure 15

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Analysis Defect	Char	Y		Y	Y	Y	
Defect Description	Char	Y		Y	Y	Y	Y

6.2.8. Container Types

Container Types is also a single block form as shown in fig. 16. This is a setup form to enter the Container Types and their descriptions to be referenced in other parts of the application. Each Product will be referenced to one of the identified container types.

Figure 16

The formats of the fields in this block are :
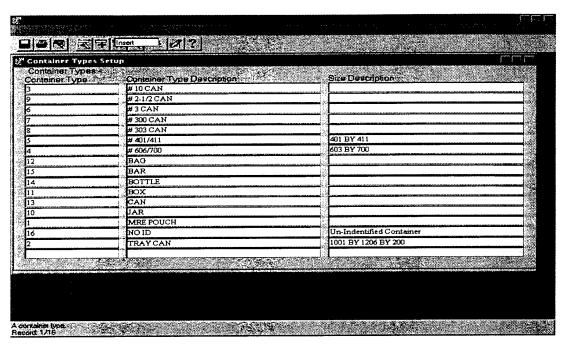
FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Container Type	Char	Y		Y	Y	Y	
Container Type Desc	Char			Y	Y	Y	Y
Size Description	Char			Y	Y	Y	Y

The user can query on any of these fields. Records can also be entered to add new Container Types.

6.2.9. Family of Rations

Family of Rations is a single block form as shown in fig.17. This is a setup form where Family of Ration Codes and their descriptions can be entered into the application. Each line item in a contract will refer to a family of ration description.

Figure 17

The formats of the fields in this block are :
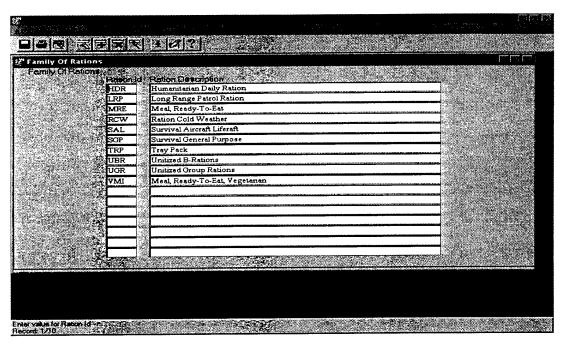
FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Ration Id	Char	Y		Y	Y	Y	
Ration Description	Char			Y	Y	Y	Y

The user can query on any of these fields.

6.2.10.   Packaging Methods

Packaging Method is also a one block setup form to enter the different packaging methods and their descriptions into the system as shown in fig: 18. Each lot that is generated in the system will reference one of the identified packaging methods.

Figure 18

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Packaging Method	Char	Y		Y	Y	Y	
Packaging Description	Char			Y	Y	Y	Y

The user can query on any of these fields.

6.2.11. Product Processes

Product Processes is also a single block setup form as shown in fig 19. This form is used to define processes used during the manufacturing of a product. Each product can be referenced to one of these process descriptions.

Figure 19

The formats of the fields in this block are :
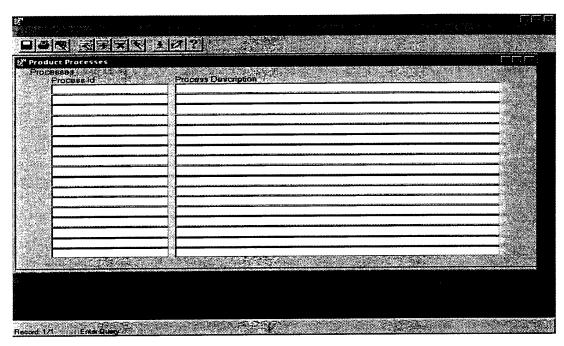
FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Process Id	Char	Y		Y	Y	Y	
Process Description	Char			Y	Y	Y	Y

6.2.12.   Units Of Measure

Units Of Measure is a single block setup form as shown in fig: 20. This form gives the different units of measure and their descriptions referenced in this application. The unit of measure is referenced by the product and used in various forms to quantify quantities.

Figure 20

The formats of the fields in this block are :
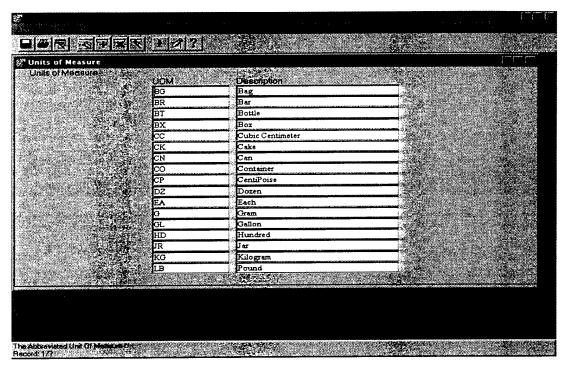
FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
UOM	Char	Y		Y	Y	Y	
UOM Description	Char			Y	Y	Y	Y

6.3. DPSC Maintenance

These forms are basically data entry forms for the DPSC maintenance. Unlike the setup forms mentioned earlier, these forms are more frequently updated. Whenever a contract is issued, the contract form must be completed. Waiver form is updated whenever a waiver is granted to a lot that is in the system. The error forms must be checked every day for USDA errors and the data must be corrected to fix the errors.

6.3.1. Contracts

The contract form is used to enter contracts into the system and looks like fig.21. This form has three blocks. The Contract Headers block is the master block, which is used to enter the contract numbers, the contract grantor, contract grantee, contract start and end dates.

28

Figure 21

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Contract Number	Char	Y		Y	Y	Y	
Awarded By	Char	Y		Y	Y	Y	Y
Awarded To	Char	Y		Y	Y	Y	Y
Start Date	Date			Y	Y	Y	Y
End Date	Date			Y	Y	Y	Y
Research Contract	Char	Y		Y	Y	Y	Y
Contract Expired	Char	Y		Y	Y	Y	Y

The last two fields are poplists with Y and N options.

The Contract Products block lists the details for the records in the master block. It will be used to enter line item information for each product that is contracted for including the minimum and maximum quantity.

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Contract Item	Char	Y		Y	Y	Y	
NSN	Char	Y	Y	Y	Y	Y	Y
Product	Char				Y		

29

Field	Datatype	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Description							
Family Of Rations	Char	Y	Y	Y	Y	Y	Y
Minimum Quantity	Number			Y	Y	Y	Y
Maximum Quantity	Number			Y	Y	Y	Y
UOM	Char				Y		

The user can query on any of the fields. But records can be entered only in the context of Contract Headers. This precisely describes the products, their description etc for which the contracts are issued. For each of the products their delivery schedules are listed in the Contract Delivery Schedules block.

Next to each record in the Contract Products block a small push button is placed. When that button is pressed the delivery schedule details are listed. This block will be used to enter delivery order information for each contract line item.

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Del.Order .No	Char	Y		Y	Y	Y	
Del.Item	Char	Y		Y	Y	Y	
Qty Required	Number	Y		Y	Y	Y	Y
First.Del Date	Date			Y	Y	Y	Y
LastDel Date	Date			Y	Y	Y	Y
Destination	Char			Y	Y	Y	Y

6.3.2. Waivers

The Waivers form is a single block form to record any waiver that will be granted by DPSC. This form is used to enter the details of the waiver process namely the person granting waiver, date of waiver, lots to which the waiver was granted and so on. The structure of the form is as fig 22.

Figure 22

The formats of the fields in this block are :
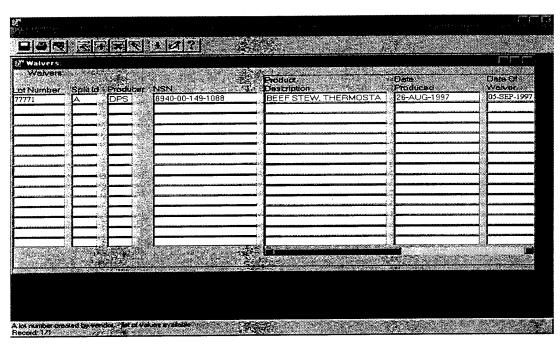
FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Lot Number	Char	Y	Y	Y	Y	Y	
Split Lot Id	Char				Y		
Producer Code	Char				Y		
NSN	Char				Y		
Product Description	Char						
Date Produced	Date				Y		
Date Of Waiver	Date	Y		Y	Y	Y	Y
Person Granting Waiver	Char	Y		Y	Y	Y	Y
Comments	Char			Y	Y	Y	Y

When the Lot Number is selected from the LOV the following 5 fields (till Date Produced) are selected based on the details of the lots already existing in Product Lots. There are two parts. The lots (first 4 fields) are in a row and the rest of them are in a frame where you have horizontal scrollbar to see all the other fields. Date of Waiver is a mandatory field and if not entered will default to sysdate.

The next two error forms are for fixing USDA bad data records. These forms are not meant for inserting records. They are mainly for query and update. Whenever the Data Normalization procedures find errors in records stored in the Data Source Storage, they will be marked as bad data. The Geninfo Errors (USDA Header)and the Failinsp Errors (USDA Details)forms list these bad records with the error code and the error

31

desription . It is the responsibility of the user to fix the bad records . Once the records are free of errors they will be normalized and entered in the operational database when the job is performed the next time.

6.3.3. Geninfo Errors

The Geninfo Errors form is a single block form as shown in the fig 23
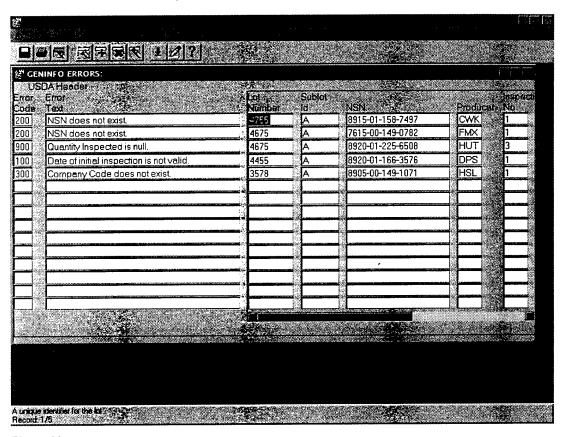
Figure 23

The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Error Code	Number	Y				Y	Y
Error Text	Char					Y	
Lot Number	Number	Y		Y	Y		Y
Sublot Id	Char	Y		Y	Y		Y
NSN	Char	Y		Y	Y		Y
Company Code	Char	Y		Y	Y		Y
Inspection No	Char	Y		Y	Y		Y
Quantity Inspected	Number			Y	Y		Y
Date Initial Inspection	Date			Y	Y		Y

Error Code	Number	Y				Y	Y
Lot Passed	Char			Y	Y		Y

It can be noticed from the previous sections in this manual that the first two columns are set in the Error Codes forms under Administration section

6.3.4. Failinsp Errors

The Failinsp Errors form is a single block form as shown in fig 24
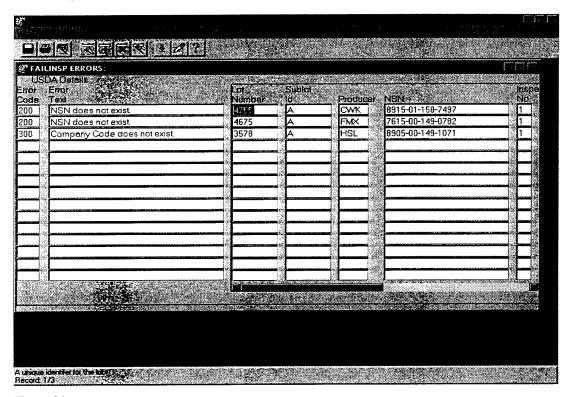
Figure 24
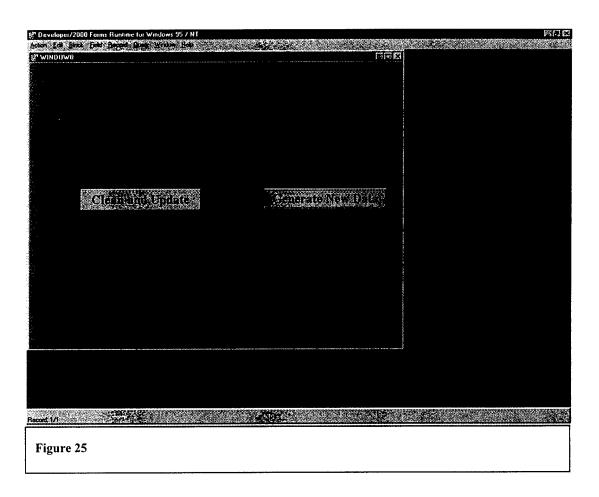
The formats of the fields in this block are :

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Error Code	Number	Y			Y		
Error Text	Char				Y		
Lot Number	Number	Y		Y	Y		Y
Sublot Id	Char	Y		Y	Y		Y
Company Code	Char	Y		Y	Y		Y
NSN	Char	Y		Y	Y		Y
InspectionNo	Char	Y		Y	Y		Y
Defect Id	Char	Y		Y	Y		Y
USDA Defect	Char			Y	Y	Y	Y
Sample Size	Number			Y	Y		Y
Number	Number			Y	Y		Y

Defects							
Comments	Char			Y	Y		Y

6.3.5. Update Data Warehouse

Under this function the user can update the data warehouse tables manually. In normal operation these tables are updated on a daily basis. However, these updates are incremental based on changes and additions made to the under lying tables. This form enables the user to manually perform the same operation by clicking on "Generate New Data". However, if the data in the data warehouse appears to be corrupt, a complete new generation of the data warehouse should take place. This is accomplished by clicking on "Clean and Update". This will clean out all data from the data warehouse and generate a new version.

Figure 25
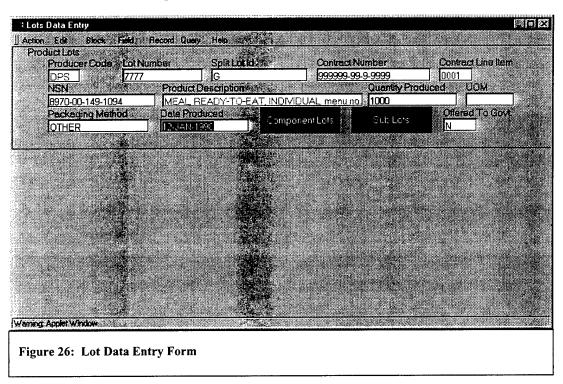
6.4. Lot Transactions

Lot transaction data will be recorded by the producing, receiving or inspecting organization. After the record has been committed the user has 30 days period in which the record can be edited via this form. Deletion of the record is also permitted during the same time period if there are no associated data records linked to the record. For example, once an inspection record has been associated to a production lot, the lot information can not longer be deleted unless the inspection record is deleted first.

6.4.1. Create Lots

This form is used to create a lot in the system. The form consists of a main form and two sub forms: (sub lot and component lot). In the main form one enters pertinent production data for a lot. This form will be frequently used by producers and assemblers to enter product lot information into the system. This information includes d ta to uniquely identify lots, production date, quantity produced and so on. There are 2 buttons on this form named Component Lots and Sub Lots. When these buttons are pressed the corresponding detail records can be entered. The sub lot form can be used to identify which sub lots can be identified under the main lot. The component lot form can be used to identify the component lots that were used during the production/assembly of the main lot.

Create Lots is a form consisting of three blocks as shown in fig 26.

Figure 26: Lot Data Entry Form

The format of each field is as follow:

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Producer Code	Char				Y		
Lot Number	Number	Y		Y	Y	Y	
Split Lot Id	Char	Y		Y	Y	Y	
Contract Number	Char	Y	Y	Y	Y	Y	Y
Contract Line Item	Char				Y		
NSN	Char	Y	Y	Y	Y	Y	
Product Description	Char				Y		
Quantity	Number	Y		Y	Y	Y	Y

Produced							
UOM	Char				Y		
Packaging Method	Char	Y	Y	Y	Y	Y	Y
Date Produced	Date	Y		Y	Y	Y	Y
Offered To Govt	Char	Y	Y		Y		

The **Producer Code** will be automatically populated based on the user id that was used to login to the system.

The **Lot Number and the Split Lot Identifier** is the Number that is assigned and marked on the lot. This can either be the Julian Date or a proprietary character string. Due to the possibility that a lot number might be split into two or more lots and offered to the government as such, the system requires that a split lot identifier is used for each lot. The system defaults to the split lot identifier "A", is no split lot identifier is marked with a inspection record.

Contract Number and Contract Line Item: Each lot needs to be cross referenced to a contract number and line item number in that contract. A list of values can be requested for these fields. If the materail lot is a contractor supplied material then the lot should be cross referenced to the assembly contract number/line item for which it is intended.

NSN: All product lots need to be cross-referenced to a National Stock Number. A list of values is also available for this field. The list field is limited to the contract item that was identified above and the component products that are part of an assembled product. This list is maintained by DPSC. Based on the NSN and use of look up tables the system will automatically populate the following fields: **Product Description** and **UOM**. The UOM is standardized so that all contract quantities, production quantities and inspection quantities refer to the same unit of measure.

Quantity Produced: The producer is requested to fill in the size of the lot that has been produced. In most cases this number should be the number of units processed or packaged.

Packaging Method: The field "Packaging Method" is to be used when thermal stabilized MRE lots are entered into the system. The purpose of the field is to differentiate between Vertical Form Fill Seal MRE Pouches and Horizontal Form Fill Seal MRE Pouches. All other products should use the packaging designation: "Other". A list of values is available to select from the three alternatives.

Date Produced: This date is the production date of the lot and should be entered in the format of dd-mmm-yyyy. It should be noted that the year is to be entered as a four digit number due to the year 2000 problems that other wise occur.
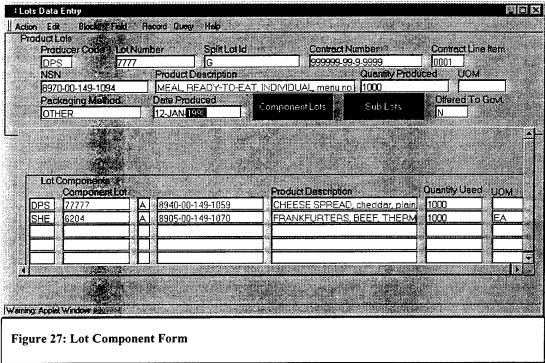
Offered To Govt. : This field is a system maintained field and used to identify if a government inspection agency has recorded an inspection record against this lot. This field will be automatically populated by the system. As the lot is created in the system, this field will be set to "NO". Once an USDA or AVI record has been posted against this lot the field will be set to "YES".

Component Lots: When the Component Lots button is pressed the block lot components pops up. This block can be used to enter the lots and quantities used in making an assembled lot/product.

Sub Lots: When the Sub Lot button is pressed. the sub lot block pops up. This block can be used when it is necessary to record the various sub lots that were used to assemble the main lot.

6.4.1.1. Component Form

When the Component Lots button is pressed the block lot components pops up. The items of this block are shown in the next page. This block is used to enter the lots and quantities used in making an assembled

Figure 27: Lot Component Form

lot/product. All fields that describe a lot can be selected from a single LOV, therefore the lot needs to exist in the system before it can be referenced. Users can only update or insert the Quantity Used.

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Component 3letter Code	Char	Y	Y	Y	Y	Y	
Component Lot Number	Number				Y		
Component Split Lot Id	Char				Y		
Component NSN	Char				Y		
Product Description	Char						
Quantity Used	Number			Y	Y	Y	Y
UOM	Char						

6.4.1.2. Sub Lot Form

To enter the details of the Sub Lots, the button Sub Lots is to be pressed.. Then a window pops up with the following fields.

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Sublot Id	Char	Y		Y	Y	Y	
Sublot Quantity	Number			Y	Y	Y	Y
UOM	Char			Y	Y	Y	Y

Sub lot ID: In this field any character string can be typed to identify the code for the sub lot. The sub lot does not have to exist in the QDMS system

Sub lot Quantity: This field is used to record the size of the sub lot. Note: The sum of all the sub lot quantities will not be cross-referenced to the total quantity produced for the main lot.

UOM: This field is used to identify the unit of measure that pertains to the sub lot quantity. The unit of measure can be selected from a LOV and has to be in compliance with the units used by the system. The unit of measure does not have to be the same as the unit of measure used for recording the quantity produced

6.4.2. Lot Receipts

Lot Receipts Form tracks the receipt of a particular lot. Lots, which do not exist in the system at the time of receipt, will be created on receiver's request. This form consists of 2 blocks namely the Receiving Header and the Receiving Details.

Figure 28: Lot Receiving Form

The Receiving Header block has the following items:

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Document Number	Char	Y		Y	Y	Y	
Document Type	Char			Y	Y	Y	Y
Document Date	Date			Y	Y	Y	Y
Date Shipped	Date			Y	Y	Y	Y
Date Received	Date			Y.	Y	Y	Y
Received By	Char				Y		

Document Number: The only mandatory field in the document header is the document number. This need to be a unique number. In the remote case that the system refuses the document number because it already existed in the system, it is suggested that the document number is appended with an additional letter (ex: -B)

Document Type: Two document types have been identified in the system: either a DD250 or a BL (bill of Loading). A pop list is available for this field.

Document Date: This is the date that the shipping document was generated (dd-mmm-yyyy)

Date Shipped: This is the date that the product was shipped (dd-mmm-yyyy)

Date Received: This is the date that the product was received (dd-mmm-yyyy). This field is autoimatically populated by the system with the system date, but can be edited by the user.

Received By: This field is automatically populated by the system, based on user id.

The user can query on any of these fields. It is worth noting that the Document Type field is a poplist. If the Date Received is not entered then it is populated with a default date of System date and the default value for Received By will be the receiving location.

For a given header record, details can be entered in the Receiving Details block. The Receiving Details items are:

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Contract Number	Char	Y	Y	Y	Y	Y	Y
Contract Item	Char			Y	Y	Y	Y
Delivery Order No	Char			Y	Y	Y	Y
Delivery Line Item	Char			Y	Y	Y	Y
Lot Number	Number	Y		Y	Y	Y	
Split Lot Id	Char	Y		Y	Y	Y	
Producer Code	Char	Y	Y	Y	Y	Y	
NSN	Char	Y	Y	Y	Y	Y	
Product	Char				Y		

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Description							
Received Quantity	Number	Y		Y	Y	Y	Y
UOM	Char				Y		
Comments	Char			Y	Y	Y	Y

Contract Number and Contract Item: When inserting detail records the Contract Number is selected from a LOV and the Contract item is automatically populated. Only contract numbers can be used that exist in the system. Maintenance of contract numbers is the responsibility of DPSC.

Delivery Order and Delivery Line Item: These fields can be populated if the delivery was made based on contractually requirements. The information for these fields can be obtained from the delivery orders.

Lot Number and Split Lot Id: No list of values is available for these fields. The information should be obtained from the shipping papers. If the shipping papers do not identify a split lot id than the user should default to the split lot identifier "A"

Producer Code: A three letter code should be entered in this field which identifies the producers location code. A LOV is available for this field, or can be found in the appendix

NSN, Product Description and UOM: The NSN should be identified for the product received. A LOV is available for this field. The selection of the NSN will also populate the Product Description and the UOM field.

Received Quantity: This field should be used to record the quantity received in the unit of measure identified.

Comments: This field is optional an can be used to record any comments regarding the delivery of each line item

When the user commits the data to the database, an internal check is made to confirm if the lot already exists in the system. If the lot is not listed then the user is asked if he wants to create the lot. The user needs to confirm this, unless an error was made in the data entry. The receipt detail can not be recorded unless the lot information is existing in the system. A list of values for available lots is not available to protect the vendors from letting every one else know how many and what quantity they have produced.

6.4.3. Finished Lot/Receipt Inspections

Finished Lot/Receipt Inspection form is used to record lot inspection data based on a lot-sampling plan. This form should be used to record either lot inspection or receipt inspection. The form consists of a header in which details are given about the lot that is being inspected. The form also contains two detail blocks as shown in fig. 4.3. The header block is used to record details such as: who performed the inspection, when it was inspected, the end result of the inspection, quantity inspected, etc.

The formats of the header fields are:

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Inspected By	Char						
Inspection Type	Char	Y	Y	Y	Y	Y	
Lot Number	Char	Y	Y	Y	Y	Y	
Split Lot Id	Char				Y		
Producer	Char				Y		

Code							
NSN	Char				Y		
Product Description	Char						
Lot Quantity Inspected	Number	Y		Y	Y	Y	Y
UOM	Char						
Date Inspected	Date	Y		Y	Y	Y	Y
Lot Accepted	Char	Y			Y	Y	Y

Inspected By: This field is automatically populated by the system based on the user login

Inspection Type: using the LOV should populate this field

Lot Number, Split Lot Id, NSN, Product Description and UOM: selecting the appropriate lot number from the LOV should populate these fields

Lot Quantity Inspected: In this field the total quantity offered for inspected needs to be recorded consistent with the UOM

Date Inspected: In this field the user should record the data that the inspection of the lot started (dd-mm-yyyy)

Lot Accepted: In this field the user should identify if the lot passed or failed the inspection. . Lot Accepted is a poplist with allowable values being Y or N

Variable Data: When this button is pressed, a form for recording variable data will pop up. In this form one can record detailed information regarding variable inspection data such as average, standard deviation, and sample size.

Attribute Data: When this button is pressed, a form for recording attribute (defect) data will pop up. In this form one can record detailed information regarding the defects that were found during the inspection

Close Defect Block: When this button is pressed, either detailed inspection form will be closed

6.4.3.1. Variable Inspection Data

The button Variable Data when pressed gives variable test results of the lots listed in the header. This block captures the variables like average, standard deviation, sample size and so on.

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	QUERY	INSERT	UPDATE
Id	Char	Y	Y	Y	Y	Y	
Description	Char				Y		
Sample Taken	Number	Y		Y	Y	Y	Y
Average	Number			Y	Y	Y	Y
Std Dev	Number	Y		Y	Y	Y	Y
UOM	Char				Y		
Comments	Char			Y	Y	Y	Y

ID, Description and UOM: The user should select the most appropriate variable description from the list of values. The appropriate list of values for the particular product inspected is maintained by DPSC. To ensure compatibility of data among various users the UOM for each variable will be specified.

Sample Taken: In this field the user should record the total number of samples that were used to calculate the variable data.

Average: In this field the user should record the average value that was calculated for the variable

Std Dev: In this field the user should record the sample standard deviation that was calculated for the variable.

Comments: This field can be used to record any worthwhile inspection information for the variable

6.4.3.2.   Attribute Data

The button Attribute Data pops up the attribute defect data entry block. This block lists the actual number of defective lots ,sample size taken, defects percentage and so on. The fields are listed in the next page. When the Defect Id is picked Defect Description gets populated. When the Sample Taken and Number Defects are entered system populates the field Pct Defective. As in other blocks user can query on any of the fields.

FIELD	DATATYPE	REQUIRED	LOV	NAVIGABLE	INSERT	QUERY	UPDATE
Defect Id	Char	Y	Y	Y	Y	Y	
Defect Description	Char					Y	
Sample Taken	Number	Y		Y	Y	Y	Y
Number Defects	Number	Y		Y	Y	Y	Y
Percent Defective	Number					Y	
Comments	Char			Y	Y	Y	Y

Defect Id and Defect Description: The user should select the most appropriate defect description from the list of values. The appropriate list of values for the particular product inspected is maintained by DPSC. It is assumed that the defect UOM is identical to the UOM of the inspected quantity.

Sample Taken: The user should record the total sample size that was evaluated for this particular defect

No of defects: The user should record the total number of defects that were found in the above sample size

Pct Defective: This field is calculated by the system and based on No of defects found and total sample taken.

Comments: This field can be used to record any worthwhile inspection information for the defect
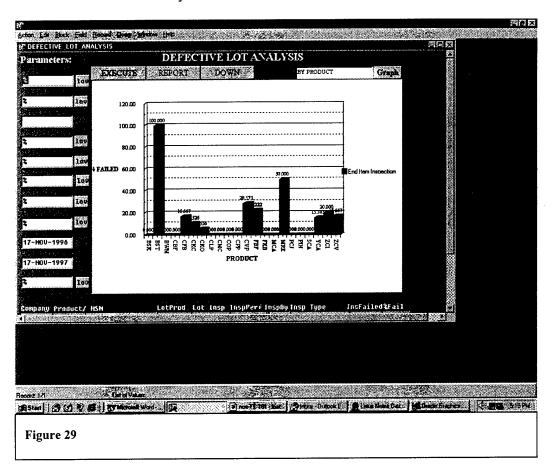
In-Process Inspections

In-process Inspections form is used to record inspection data that was collected during the manufacturing process.   The system has been setup to record inspection data for the following inspection types: pre-retort inspection, post-retort inspection, rework inspection, and inspection during the assembly process. As in the Lot Inspection, inspection data can either be recorded as variable data or as attribute (defect) data. The form

looks like fig 4.4. The inspection header for in-process inspection is identical to the inspection header discussed before with one exception: a lot can not pass or fail the inspection. Also the variable data form is identical to the form discussed previously under the lot inspection. The attribute form differs from the previous discussed form as it assumes that the inspections are based on a 100% inspection, with other words the sample size for the defect recorded is identical to the total quantity inspected as identified in the inspection header.

6.5. Analysis

6.5.1. Data Analysis

Figure 29

6.5.1.1. Production Analysis

Input Variables:
> Producer
> Product
> Product Family
> Product Class
> Container
> Process

43

Inspected by:
Start Date
End Date
Inspection Type
Output Variables:
Producer
Product
Number of Lots Produced
Total Quantity Produced
Inspection Location
Number of defects found
Ratio Defects over quantity produced

6.5.1.2. Inspection Defect Analysis

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type
Output Variables:
Producer
Product
Number of lots produced
Number of lots inspected by specific organization location
Quantity inspected by specific organization location
Number of estimated defects found
Ratio of estimated defects found over quantity inspected

6.5.1.3. Variable Defect Analysis

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type
Output Variables:
Producer
Product
Number of lots produced
Quantity produced
Inspection Location

Inspection Oraganization
Number of lots inspected by inspection organization
Average value for variable
Average coefficient of variation for variable

6.5.1.4. Defective Lot Analysis

The Defective Lot Analysis analyzes the data in the database in the number of failed inspections in comparison to the number of lot inspections performed.

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type
Output Variables:
Producer
Product
Lot Produced
Lot Inspected
Inspected Performed
Inspected By
Inspection Type
Inspection Failed
Percent Failed Inspections

6.5.1.5. Rework Analysis

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type
Output Variables:
Producer
Product
Total Lots produced
Total lots offered to the government
Total units offered to government
Number of lots reworked that were offered to government
Total lots not offered to the government
Total units not offered to government

Number of lots reworked that were not offered to government

6.5.2.   Trend Analysis

6.5.2.1.   Production Trend

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type
Output Variables per time period:
Producer
Product
Number of Lots Produced
Total Quantity Produced
Inspection Location
Number of defects found
Ratio Defects over quantity produced

6.5.2.2.   Inspection Defect Trend

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type
Output Variables per time period:
Producer
Product
Number of lots produced
Number of lots inspected by specific organization location
Quantity inspected by specific organization location
Number of estimated defects found
Ratio of estimated defects found over quantity inspected

6.5.2.3.   Variable Defect Trend

Input Variables:
Producer
Product
Product Family

Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type

Output Variables per time period:
Producer
Product
Number of lots produced
Quantity produced
Inspection Location
Inspection Organization
Number of lots inspected by inspection organization
Average value for variable
Average coefficient of variation for variable

6.5.2.4. Defective Lot Trend

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type

Output Variables:
Producer
Product
Lot Produced
Lot Inspected
Inspected Performed
Inspected By
Inspection Type
Inspection Failed
Percent Failed Inspections

6.5.2.5. Rework Trend

Input Variables:
Producer
Product
Product Family
Product Class
Container
Process
Inspected by:
Start Date
End Date
Inspection Type

Output Variables:

47
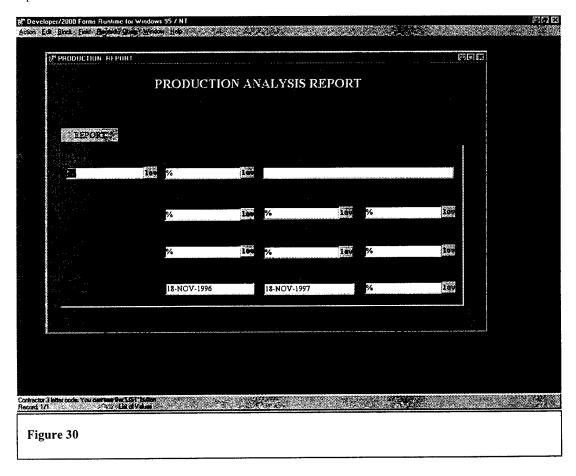
Producer
Product
Total Lots produced
Total lots offered to the government
Total units offered to government
Number of lots reworked that were offered to government
Total lots not offered to the government
Total units not offered to government
Number of lots reworked that were not offered to government

6.5.3. Analysis Reporting

The Analysis Reports will show in a report from the detailed inspection information that is stored in the database. When a specific report is selected, a form will be displayed in which the user can set various parameters that will restrict the output of the query to the report. Once all parameters have been set, the user can click on the "REPORT" button to generate the report. In the client server configuration, a report writter will display the report. In the web browser application, Acrobat Reader will be used to display the report.

Figure 30

6.5.3.1. Production Detail

This report will display the detailed information regarding the units that were removed from the production lot during one of in-process inspection points:

Input parameters for report:

Producer
Product
Product Class
Container Type
Process Type
Product family
Defect
Inspection Type
Start Date Production
End Date Production
Inspected By
Output parameters
Date production
Producer
Product
Lot Number
Quantity Produced
Inspection Date
Inspection Location
Inspection Organization
Defect Description
Number of Defects Removed

6.5.3.2. Inspection Defect Detail

This report will display the detailed information regarding defective units that were found during the inspection process of lots:
Input parameters for report:
Producer
Product
Product Class
Container Type
Process Type
Product family
Defect
Inspection Type
Start Date Production
End Date Production
Inspected By
Output parameters
Date production
Producer
Product
Lot Number
Inspection Date
Inspection Location
Inspected quantity
Inspection Organization
Defect Description
Sample size
Number of Defects Removed
Estimated number of defects in quantity inspected

6.5.3.3. Variable Defect Detail

This report will display the detailed information regarding the variable data that was recorded during the inspection process of lots:

> Input parameters for report:
>> Producer
>> Product
>> Product Class
>> Container Type
>> Process Type
>> Product family
>> Variable
>> Inspection Type
>> Start Date Production
>> End Date Production
>> Inspected By
> Output parameters
>> Date production
>> Producer
>> Product
>> Lot Number
>> Inspection Date
>> Inspection Location
>> Inspection Organization
>> Vraibale Description
>> Sample Size
>> Average value
>> Standard Deviation
>> Unit of measure for variable

6.5.3.4. Defective Lot Detail

This report will display the detailed information regarding defective lots that were found during the inspection process of lots:

> Input parameters for report:
>> Producer
>> Product
>> Product Class
>> Container Type
>> Process Type
>> Product family
>> Lot Acceptance (Y/N)
>> Inspection Type
>> Start Date Production
>> End Date Production
>> Inspected By
> Output parameters
>> Date production
>> Producer
>> Product
>> Lot Number
>> Quantity Produced
>> Inspection Date
>> Inspection Location
>> Inspection Organization

Quantity Inspected
Inspection Status

6.5.3.5. Rework Detail

This report will display the detailed information regarding inspections that were performed. Even though the intend of the report is to show only rework details, the query parameters can be set such that also details are given regarding the other inspection locations.

Input parameters for report:
Producer
Product
Product Class
Container Type
Process Type
Product family
Inspection Type
Start Date Production
End Date Production
Inspected By
Output parameters
Date production
Producer
Product
Lot Number
Inspection Date
Inspection Location
Inspection Organization
Quantity Inspected

6.6. Reports

The next menu item is the Reports. This consists of the name of the different reports. When any of these reports are selected a parameter form appears on the screen. The user can set the parameters in that form and then click on REPORT so that the report will run for those parameters. Most of the parameter fields have the list of values from which the values of the parameters can be picked. The report form will look like this:
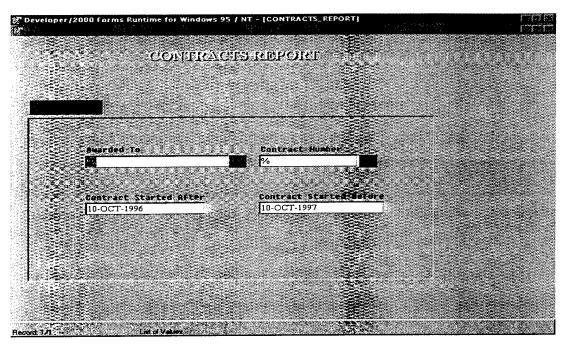
Figure 31

6.6.1. Contracts

This report will show in a report form the contracts header and details as listed in the database.

> ***Input parameters for this report are:***
> > Awarded To:
> > Contract Number:
> > Contact Start Date Period: (Begin/End Date)

If the parameters Awarded To ,Contract Number are not mentioned , the default value % is used to run the report for all the values available in the database.
The report will contain information on the Contracts. It will be a master detail report. For every master record all the detail records are displayed.

> ***Information displayed in this report:***
> > **Header:**
> > > Contract Number
> > > Awarded To
> > > Research Contract(Y/N)
> > **Detail:**
> > > Contract Item
> > > NSN
> > > Product Description
> > > Family Of Rations
> > > Min. Quantity
> > > Max. Quantity.

6.6.2. Products

This report will show in a report form the product information as listed in the database.

Input parameters for this report are:
    NSN
    Product Class
    Process Type
    Container Type
Information displayed in this report
    NSN
    Product Description
    Three Letter Product Code
    Label Weight
    Unit Of Measure
    Process Type
    Container
    Specification
    Number in Case

If the input parameters are not mentioned the report will be run for a default value of % meaning for all existing values in the database.

6.6.3. Product Structure

This report will show in a report form the various assembled products and the components that could be used in the assembly.

Input parameters for this report are:
    NSN

Information displayed in this report:
    Header:
      NSN
      Product Description
    Detail
      Component NSN
      Component Description
      Component Quantity

6.6.4. Lot Activity

This report will show in a report form the history of a lot as listed in the database.

Input parameters for this report are
    Production Date (Time Frame)
    NSN
    Product Class
    Process Type
    Container Type
Information displayed in this report:
    Product Lot
    Producer Code
    NSN
    Activity Date
    Activity
    Quantity

Performed By

6.6.5. Product Class/Defect

This report will show in a report form the defects that are assigned to a product class as listed in the database.

Input parameters for this report are:
Product Class
Information displayed in this report:
Product Class
Defect Id
Defect Description
Result Type
Defect Category
Unit Of Measure
Analysis Defect ID
Analysis Defect Description

6.6.6. USDA Inspection Header Errors

This report will show in a report form USDA geninfo_errs table that lists all the errors that were incurred during the import process of the geninfo information.

Input parameters for this report are:
Producer Code
NSN
Information displayed in this report:
Company Code
Error Code
Error Description
Lot Code
NSN
Date Initial Inspection
Inspection No
Quantity Inspected
Lot Passed

6.6.7. USDA Inspection Detail Errors

This report will show in a report form USDA fail_errs table that lists all the errors that were incurred during the import process of the failinsp information

Input parameters for this report are:
Producer Code
NSN
Information displayed in this report:
Company Code
Error Code
Error Description
Lot Code
NSN
Inspection No

54

Defect Id
Number Defects
Sample Size

6.6.8. Organizations/Locations

This report will show in a report form the organization and its location information as listed in the database.

Input parameters of this form are:
Organization Name
Location 3 Letter Code

Information displayed in this report:
Organization Name
Organization Id
Location 3 Letter Code
Location Description
Address
Cage Number

6.6.9. Organizations/Products

This report will show in a report form the products that were produced by an organization as listed in the database table Organization_Products. This report can also be used to show which companies have produced a product in the past.

Input parameters of this report are:
Organization Name
NSN

Information displayed in this report :
Header:
Organization Name
Detail:
NSN
Product Description

6.6.10.   Production Lot

This report will show in a report form the product lot information as listed in the database.

Input parameters of this report are:
Producer Code
Production Date(Bfore/After)
NSN

Product Class
Product Type
Container Type
Contract Number
Contract Line Item
Family Of Rations
Information displayed in this report:
Producer Code
Lot Number

NSN
Product Description
Packaging Method
Contract Number
Contract Line Item
Lot Quantity
 Offered To Government

6.6.11.   Receipts

This report will show in a report form the product receipt information as listed in the database.

Input parameters for this report are:
Received By
Receipt Time(Before/After)
 Contract Number
NSN
 Product Class
Process Type
Container Type
Information displayed in this report:
Date Received
 Received By
 Document Number
 Document Type
 Document Date
 Contract Number
 Contract Item
 Delivery Order Number
 Delivery Line Item
 Received Quantity

6.6.12.   Waivers

This report will show in a report form the waiver information as listed in the database.

Input parameters for this report are:
 Lot Produced Date(Start/End)
 Producer Code
 NSN
Information displayed in this report:
 Lot Number
 Producer Code
 NSN
 Date Of Waiver
 Waiver Granted By
 Comments

Appendix C
QDMS File Transfer Utility User Manual

QDMS File Transfer Utility

User Instructions

March 1999

1 Introduction

This is a utility to facility the transfer of vendor data files up to the QDMS application server.
The file specifications are defined in the Appendix of this document. This web utility will send files up to the server where they will be processed nightly in batch mode.
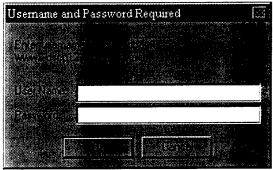
The application can be run from any browser supporting SSL. SSL is used in order to encrypt any communication between the user's browser and the server.

2 Connecting to QDMS

The URL for the application is as follows

https://www2.dscp121.dscp.dla.mil/wa/owa_dba/owa/imgload.show_form
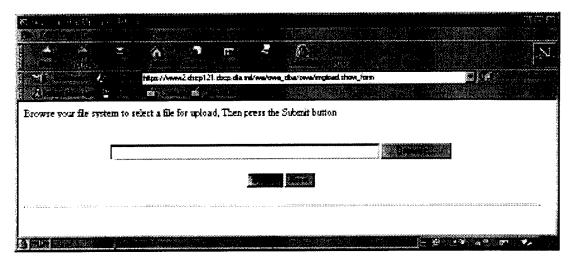
The following login dialog box will appear after the URL is entered.
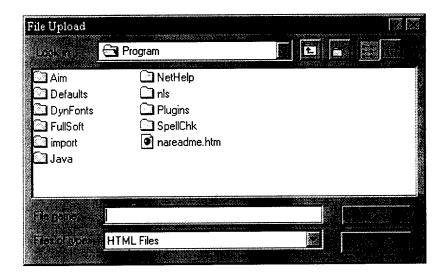
Enter your QDMS (Oracle) UserName and Password and Click on the **OK** button.

3 Submitting a File
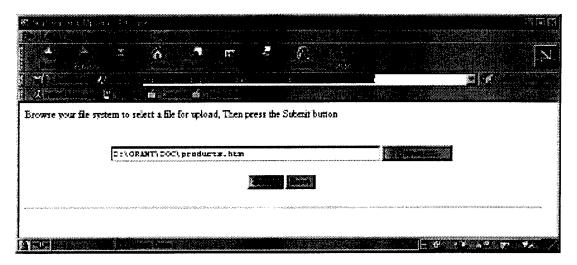
The following will appear in your browser after successful login.
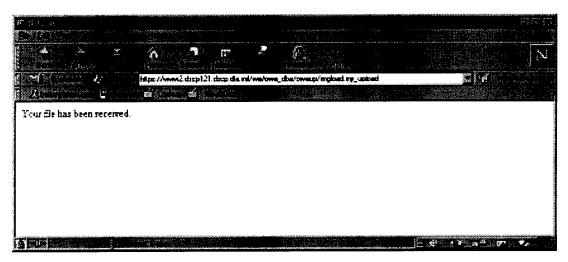
Click on the Browse button to select a file from your local file system for upload. Only one file may be sent at any one time. When the browse button is clicked you will see the following window. You may then navigate through your file system to select a file.

After selecting a file it will appear in the field on your screen.

Browse your file system to select a file for upload. Then press the Submit button

C:\GRANT\DOC\products.htm

Clicking on the **submit** button will send the file up to the server and you will see the following message.

Your file has been received.

To submit another file click on the Back button in the browser toolbar to go back to the previous page and select a different file.

4 Record Resubmission

If an error report is received, or if a previous submitted record is considered to be erroneous, these records need to be corrected and resubmitted. If a previous accepted record is re-submitted for correction, this record must meet certain conditions. A record is considered re-submitted if the bolded fields identified in the appendix match an existing record in the database. When this happens, all non-bolded fields in the record are updated from the submitted data. A new record is created if no match is found for the bolded fields.

A record can also be updated by going on-line and changing the record.
Record deletions must be done on-line.

Appendix

1   Introduction

The Quality Data Management System recently deployed at DSCP is used to collect and analyze quality data related to the production of combat rations. This data comes from vendors, the USDA, and the AVI. The USDA and AVI data is currently loaded into the Oracle database via electronic file transmission between their respective system and QDMS. The vendors enter data into the system by keying it into forms that run over the Web. Vendors that currently store their quality control records electronically, may find this inconvenient. This document provides a specification for a file based data transfer of QDMS type data between vendors (Producers & Assemblers) and the QDMS application.

This specification will allow vendors to export data from their existing applications into a predefined file format. These files will then be sent to the QDMS system. The QDMS system will validate all incoming records, and import them in the appropriate tables. If an error is found during the validation step, an error report will be e-mailed to the sender of those particular records. The sender must then resubmit corrected records.

2   Specification

The following section will specify the format of the data transmission and the format of the files to be submitted.

2.1   Specification for Data Transmission

- A Web Cartridge on the QDMS server side and a SSL (Secured Socket Layer) enabled Web Browser on the client side (producer/assembler) will be used to send the data files to QDMS. The cartridge will facilitate user authentication and data encryption.
- The data files send must be Text (ASCII) files
- All data files need to contain records that meet the specified format (section 2.2). Each file can contain a single record type or multiple record types. However, each record needs to be identified by a record prefix.
- Vendors must provide an E-mail address to receive error reports.
- All resubmitted inspection records must have the same key fields as the original record.

2.2 File Format Specification

In this section the format for the Electronic File Transfer Structure of QDMS data is specified. Each record is to be identified by a record prefix to indicate the type of record. The fields of a record are separated by a "comma" (,). Text fields should be enclosed in quotes ("text1", "text2"). Certain fields have a defined list of a values (LOV). Section 3 lists the LOV's for "inspection type" , "defect id" , "packaging method" and "unit of measure". LOV's for "National Stock Number", and "Location 3Letter code" can be obtained from DSCP. **Bold** fields are key fields which make a record unique. Records that have identical key will result in the previous record being over written by the new record.

2.2.1 Product Lots

Product Lots will have the following record formats.
"PL", "PL10" ,"PL20" ,"PL30" ,"PL40" ,"PL50", "PL60", "PL70", PL80,"PL90"

PL	Record prefix. PL denotes a product lot record.	Char(2)	Required
PL10	Location 3Letter code. This is the code of the production site. (lov)	Char(3)	Required
PL20	Lot Number	Char(10)	Required
PL30	Split Lot ID (Default is A)	Char(1)	Required
PL40	National Stock Number (XXXX-XX-XXX-XXXX) (lov)	Char(16)	Required
PL50	Packaging Method (HFFS, VFFS, OTHER) (lov)	Char(15)	Required
PL60	Contract Number (Contract that lot is made for)	Char(18)	Required
PL70	Contract Item	Char(5)	Required
PL80	Lot Quantity	Number	Required
PL90	Date produced (dd-mon-yyyy)	Char(11)	Required

2.2.2 Sub Lots

Sub Lots are smaller lots that a lot is divided into. They usually represent a portion of a day's production, whereas a lot usually represents a full day's production. Sub Lots will have the following format.
"SL", "SL10", "SL20", "SL30", "SL40", "SL50", SL60, "SL70"

SL	Record prefix. SL denotes a Sub lot record.	Char(2)	Required
SL10	Location Three Letter code. (Code of the production site) (lov)	Char(3)	Required
SL20	Lot Number	Char(10)	Required
SL30	Split Lot ID (Default is A)	Char(1)	Required
SL40	National Stock Number (XXXX-XX-XXX-XXXX) (lov)	Char(16)	Required
SL50	SUB Lot ID	Char(9)	Required
SL60	Sub Lot Quantity	Number	Required
SL70	Unit of Measure	Char(10)	Required

2.2.3 Component Lots

Component Lots are lots that go into making a lot. This information is especially important when making Assembly lots. Component Lots will have the following format.
"CL", "CL10", "CL20", "CL30", "CL40", "CL50", "CL60", "CL70", "CL80", CL90

CL	Record prefix. CL denotes a component lot record.	Char(2)	Required
CL10	Location Three Letter Code. (Lot being produced) (lov)	Char(3)	Required
CL20	Lot Number (Lot being produced)	Char(10)	Required

6

CL30	Split Lot ID (Lot being produced)	Char(1)	Required
CL40	NSN (XXXX-XX-XXX-XXXX) (lot being prod.) (lov)	Char(16)	Required
CL50	Component Location Three Letter Code. (Lot being used) (lov)	Char(3)	Required
CL60	Component Lot Number (Lot being used in production)	Char(10)	Required
CL70	Component Split Lot ID (Lot being used in production)	Char(1)	Required
CL80	Component NSN (XXXX-XX-XXX-XXXX) (Lot being used)(lov)	Char(16)	Required
CL90	Component Lot Quantity (Qty. of component lot used)	Number	Required

2.2.4 Inspection Headers

Inspection Headers are records that indicate that an inspection was performed. e.g. Pre-retort, End item, etc. Inspection Headers have the following format:
"IH", "IH10" ,"IH20" ,"IH30" ,"IH40",IH50, IH60, "IH70", "IH80",IH90, "IH100"

IH	Record prefix. IH denotes an inspection header record.	Char(2)	Required
IH10	Location 3Letter code. (Code of location producing lot) (lov)	Char(3)	Required
IH20	Lot Number (Part of lot identifier)	Char(10)	Required
IH30	Split Lot ID (Part of lot identifier)	Char(1)	Required
IH40	National Stock Number (Part of lot identifier) (lov)	Char(16)	Required
IH50	Inspection ID (Part of link to inspection detail records)	Number	Required
IH60	Inspection Type (e.g. End Item, Post Retort)(Part of Link) (lov)	Number	Required
IH70	Date inspected (dd-mon-yyyy) (Part of link to Insp. Detail records)	Char(11)	Required
IH80	Inspected By. (Code of location inspecting lot) (Part of Link) (lov)	Char(3)	Required
IH90	Lot Quantity Inspected	Number	Required
IH100	Lot Accepted (Y/N)	Char(1)	Required

Inspection Ids must be unique for all inspections on the same day.

2.2.5 In-process Defect Records

In-process Defect Records are the details of an In-process inspection were product units are removed from the lot due to a defect. Examples of in-process inspections are: pre retort inspection, post retort inspection, assembly inspection and rework. Each inspection type will be identified with an integer. In-process Defect Records will have the following record formats:
"IA", IA10, IA20, "IA30", "IA40", IA50, "IA60", IA70, "IA80"

IA	Record prefix. IA denotes an in-process attribute record.	Char(2)	Required
IA10	Inspection ID (Part of link to inspection header records)	Number	Required
IA20	Inspection Type (e.g. End Item , Post Retort) (Part of link) (lov)	Number	Required
IA30	Date inspected(dd-mon-yyyy) (Part of link to Insp. Header records)	Char(11)	Required
IA40	Inspected By. (Code of location inspecting lot) (Part of link) (lov)	Char(3)	Required
IA50	Inspection Detail ID (Part of record unique identifier)	Number	Required
IA60	Defect Id (lov)	Char(5)	Required
IA70	Number of Defects	Number	Required
IA80	Comments	Char(50)	Required

2.2.6 End Item Defect Records

End Item Defect Records are inspection defect records for lot inspections based on statistical sample taking. The identification of defects can either result in the lot passing

or failing the inspection. The pass/fail information is stored in the inspection header record. End Item Inspections can either be an end item inspection or a receipt inspection. End Item Defect Records will have the following record formats.
"EI", EI10, EI20, "EI30", "EI40", EI50, "EI60", EI70, EI80, "EI90"

EI	Record prefix. EI denotes an end item defect record .	Char(2)	Required
EI10	Inspection ID (Part of link to inspection header records)	Number	Required
EI20	Inspection Type (e.g. End Item , In Process) (Part of link) (lov)	Number	Required
EI30	Date inspected(dd-mon-yyyy) (Part of link to Insp. Header records)	Char(11)	Required
EI40	Inspected By. (Code of location inspecting lot) (Part of link) (lov)	Char(3)	Required
EI50	Inspection Detail ID (Part of record unique identifier)	Number	Required
EI60	Defect Id (lov)	Char(5)	Required
EI70	Sample Size Taken	Number	Required
EI80	Number of Defects	Number	Required
EI90	Comments	Char(50)	Required

2.2.7 Variable Data

Variable Data are inspection detail records with variable data results. Variable data is based on a statistical sampling and can either be performed during production (recorded as an in-process inspection) or performed as part of lot inspection (recorded as an end item inspection). Variable Data will have the following record formats.
"VD", VD10, VD20, "VD30", "VD40", VD50, "VD60", VD70, VD80, VD90, "VD100", "VD110"

VD	Record prefix. VD denotes a variable data record.	Char(2)	Required
VD10	Inspection ID (Part of link to inspection header records) (lov)	Number	Required
VD20	Inspection Type (e.g. End Item , In Process) (Part of link) (lov)	Number	Required
VD30	Date inspected(dd-mon-yyyy) (Part of link to Insp. Header records)	Char(11)	Required
VD40	Inspected By. (Code of location inspecting lot) (Part of link) (lov)	Char(3)	Required
VD50	Inspection Detail ID (Part of record unique identifier)	Number	Required
VD60	Variable Id (lov)	Char(5)	Required
VD70	Sample Size Taken	Number	Required
VD80	Average	Number	Required
VD90	Standard Deviation	Number	Required
VD100	UOM (lov)	Char(3)	Required
VD110	Comments	Char(50)	Required

2.2.8 Receiving Header

Receiving Header records indicate that a receipt occurred.

Receiving Header records will have the following format.
"RH", "RH10" ,"RH20" ,"RH30" , "RH40", "RH50", "RH60"

RH	Record prefix. RH denotes a receiving header record .	Char(2)	Required
RH10	Document Number(Shipping document number) (Link to detail)	Char(15)	Required
RH20	Date received(dd-mon-yyyy) (Part of link to Detail records)	Char(11)	Required
RH30	Received By. (Code of location receiving lot) (part of link to detail)	Char(3)	Required
RH40	Date shipped (dd-mon-yyyy)	Char(11)	Required
RH50	Document Type (DD250, ...)	Char(6)	Required
RH60	Document Date (dd-mon-yyyy)	Char(11)	Required

2.2.9 Receiving Detail

Receiving Detail records are line items in a receipt.

Receiving detail records will have the following record formats.
"PD", "RD10" ,"RD20" ,"RD30" ,"RD40" ,"RD50", "RD60", "RD70", "RD80","RD90", RD100, "RD110", "RD120", "RD130"

RD	Record prefix. RD denotes a receiving detail record.	Char(2)	Required
RD10	Document Number(Shipping document number) (Link to header)	Char(15)	Required
RD20	Date received (dd-mon-yyyy) (part of link to header)	Char(11)	Required
RD30	Received By. (three letter location code) (part of link to header) (lov)	Char(3)	Required
RD40	Location 3Letter code. This is the code of the production site. (lov)	Char(3)	Required
RD50	Lot Number	Char(10)	Required
RD60	Split Lot ID (Default is A)	Char(1)	Required
RD70	National Stock Number (XXXX-XX-XXX-XXXX) (lov)	Char(16)	Required
RD80	Contract Number (Contract that lot is received for)	Char(18)	Required
RD90	Contract Item	Char(5)	Required
RD100	Received Quantity	Number	Required
RD110	Delivery Order Number (If being delivered as per contract)	Char(6)	Required
RD120	Delivery Line Item Number (If delivering as per schedule)	Char(6)	Required
RD130	Comments	Char(50)	Required

3 List of Values

In this section, we are giving you the currently available List of Values (LOV) that are being used by QDMS. DSCP reserves the right to expand these LOV in the future.

3.1 Inspection Type

The following table lists the current inspection types that QDMS recognizes. The numeric value of the inspection type is to be used in the submitted record.

INSPECTION_	INSPECTION_TYPE_DESCRI
1	End Item Inspection
2	Pre-Retort
3	Post-Retort
4	Rework
5	Assembly
6	Receipt Inspection

3.2 Defect ID and Variable ID's

Defect and variable ID's are differentiated between:

- defects found and removed during in-process inspection (inspection type 2,3,4 and 5)
- defects found during inspection based on a sample from a lot (inspection type 1 and 6)
- variable data attributes measured during in-process inspection (inspection type 2,3,4 and 5)
- variable data attributes measured during lot inspection (inspection type 1 and 6)

Please contact DSCP regarding the defects/variables that can be used for specific product groups and for specific inspection types. Also contact DSCP regarding the cross reference between QDMS defect ID's and defects identified in the product specifications.

3.2.1 In-Process Defect ID's

The following table lists the defects that QDMS recognizes. The number listed in the first column is to be used in the record. Please contact DSCP regarding the defects that can be used for specific product groups and for specific inspection types.

DEFECT_ID	DEFECT_DESCRIPTION
7000	DELAMINATION
7010	HOLE/LINE CUT
7020	ABRASION
7030	SEAL WRINKLE
7040	SEAL NOT AS SPECIFIED
7050	SEAL CONTAMINATED
7060	LABEL
7070	NET WEIGHT
7080	OTHER
7045	SEAL - OPEN

3.2.2 Finished Product Defect ID's

The following table lists the defects that QDMS recognizes. The number listed in the first column is to be used in the record. Please contact DSCP regarding the defects that can be used for specific product groups and for specific inspection types.

DEFECT_ID	DEFECT DESCRIPTION
1001	ABBERRATION
1010	ABRASION
1020	ANALYTICAL TESTING
1030	BONE
1040	BROKEN ITEM
1050	CAN CLOSURE
1053	CAN CONDITION EXTERIOR
1055	CANDY OVERWRAP
1056	CAN CONDITION INTERIOR
1060	CASE PACKING
1070	COAGULATION
1080	COATING
1090	COLOR
1100	COMPRESSION STREAKS
1110	CONSISTENCY
1120	COUNT
1130	CRISPNESS
1140	DELAMINATION
1150	DIMENSION
1160	DISCERNIBLE UNITS
1170	DRAINED WEIGHT
1190	EMULSION STABILITY
1200	EVIDENCE OF SCORCHING
1210	EVIDENCE OF THAWING
1220	EXCESS BAG MATERIAL
1230	FAT / CONNECTIVE TISSUE
1240	FIRST ARTICLE COMPARISON
1250	FLAVOR
1260	FLEX FRACTURES
1270	FOREIGN MATERIAL
1280	FRAGILE
1290	FREE LIQUID
1300	GUMMY CENTER
1310	HARD
1320	HOLE/LINE CUT
1326	HOLE/TEAR MRE CARTON
1328	HOLE/TEAR MRE BAG
1330	DEHYDRATION
1340	POUCH INNER SURFACE
1350	INTERNAL PRESSURE
1360	LABELING
1370	MARKINGS (MAJOR)

1372	MARKINGS (MINOR)
1380	MIL-I/SPC
1390	MOLDY
1400	NET WEIGHT
1410	NOT FREE OF LUMPS
1415	FLOWABILITY
1420	APPEARANCE
1430	NOT SPREADABLE
1450	FOREIGN ODOR
1460	OILINESS
1465	OILING OFF
1470	OTHER
1480	OXYGEN CONTENT
1490	OXYGEN SCAVENGER
1500	PACKAGING
1503	PALLET COMPONENTS
1505	PALLET LOAD
1508	PEEL INDICATOR
1510	PROCESS DEVIATION
1520	SOLUBILITY
1530	REHYDRATION
1540	RESIDUAL GAS
1560	SEAL - DIMENSIONS
1570	SEAL - ENTRAPPED MATTER
1580	SEAL - TEAR NOTCH
1590	SEAL - WRINKLE
1600	SEAL - IMPRESSION
1610	SEAL - OPEN
1620	SEAL STRENGTH
1630	SEDIMENT PRESENT
1640	SIEVE ANALYSIS
1650	SIFTER
1660	SOGGY AREA
1670	STERILITY
1680	STRESS CRACKS
1690	SURFACE OILY
1700	SWOLLEN BAG
1710	TEAR NOTCH
1720	TEXTURE
1725	TRACEABILITY (MAJOR)
1726	TRACEABILITY (MINOR)
1730	US GRADE
1740	UNCLEAN
1750	UNCOOKD AREAS
1760	GRAIN STRUCTURE
1780	DISTRIBUTION OF COMPONENTS
1790	DISTRIBUTION OF INGEDIENTS

1795	UNIT PACKAGING
1800	UNITS SIZE AND SHAPE
1810	VACUUM
1820	WEEPING
1830	WHOLE UNITS
1327	HOLE/TEAR ACCESSORY BAG
1021	ASH ANALYSIS
1022	FAT ANALYSIS
1023	MOISTURE ANALYSIS
1024	PROTEIN ANALYSIS
1025	SALT ANALYSIS

3.2.3   In-Process Variable ID's

The following table lists the variables that QDMS recognizes. The number listed in the first column is to be used in the record. Please contact DSCP regarding the variables that can be used for specific product groups and for specific inspection types.

DEFECT_ID	DEFECT_DESCRIPTION
5000	NET WEIGHT
5010	DRAIN WEIGHT
5020	RESIDUAL GAS
5030	VISCOSITY
5040	FAT ANALYSIS
5050	SALT ANALYSIS

3.2.4   Finished Product Variable ID's

The following table lists the variables that QDMS recognizes. The number listed in the first column is to be used in the record. Please contact DSCP regarding the variables that can be used for specific product groups and for specific inspection types.

DEFECT_ID	DEFECT_DESCRIPTION
4010	NET WEIGHT
4020	RESIDUAL GAS
4030	VISCOSITY
4040	SEAL STRENGTH

3.3 Packaging Method

The following table lists the current packaging methods that QDMS recognizes. The text listed in the first column is to be used in the record.

PACKAGING_	PACKAGING_METHOD_DESCRIPTION
HFFS	Horizontal Form Filled Sealer
OTHER	Packaging Method other than those listed
VFFS	Vertical Form Filled Sealer, Tri Laminate
VFFS4	Vertical Form Filled Sealer, Quad Laminate

3.4 Unit of Measure

The following table lists the current Unit of Measures that QDMS recognizes. The abbreviation listed in the first column is to be used in the record. However, to make data analysis possible, QDMS determines the unit of measure that has to be used for lot quantities, inspection quantities and variable data. Please consult DSCP regarding the UOM that is to be used for these variables.

UOM_ABR	UOM_DESCRI
BG	Bag
BR	Bar
BT	Bottle
BX	Box
CC	Cubic
CK	Cake
CN	Can
CO	Container
CP	CentiPoise
DZ	Dozen
EA	Each
G	Gram
GL	Gallon
HD	Hundred
JR	Jar
KG	Kilogram
LB	Pound
ME	Meal
MX	Thousand
OZ	Ounce
PG	Package
PZ	Packet
QT	Quart
RA	Ration

Appendix D

QDMS: System Diagram

DPSC

DPSC authorized users

DPSC Notes Pump/ Notes Server

DPSC Server

Server Side Data Check

Client Side Data Check

modem

Client Side Data Check

Internet Connection

Bulletin Board System

modem

USDA

AVI Notes Server

WebForm

producers/assemblers