Fall 9-1-2006

Renewal Letter Generation System for Continental Western Insurance Company

Tom Shide
Dakota State University

RENEWAL LETTER GENERATION SYSTEM FOR CONTINENTAL WESTERN INSURANCE COMPANY

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

September, 2006

By
Tom Shide

Project Committee:

Ronghua Shan
Omar El-Gayar

PROJECT APPROVAL FORM

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Student Name: Tom Shide

Master's Project Title: Renewal Letter Generation System For Continental Western Insurance Company

Faculty supervisor: _____ Date: 12/2/2006

Committee member: _____ Date: Dec. 2, 2006

Committee member: _____ Date: _____

ACKNOWLEDGMENT

- I would like to acknowledge my wife Jennifer Shide. Without her support and understanding I would have been able to complete this project
- The following personnel at Continental Western Insurance deserve acknowledge for giving me authorization to move forward with the project: Tammie Heeren, IS Manager for the Business Systems Analyst team; Bob Andersen IS Manager for the Transaction Processing Programming Team; and Deb Thie, IS Manager for the Data Warehouse Team.
- The following personnel at Continental Western Insurance deserve acknowledgement for input given on the design of this project: Ryan Roberts, Business Systems Analyst; Dan Mueggenberg, Senior Programmer/Analyst; Gloria Gary, Business System Analyst; Don Ballew, Database Administrator, Brian Christy, Lead Programmer/Analyst; Leann Battreal, Senior Programmer/Analyst and Anne Pittard, Business Systems Analyst.

ABSTRACT

The idea for this project was born in a meeting of supervisors in my department at Continental Western Group (CWG) Insurance Company. This meeting took place about nine months ago in January of 2006. We have a system in place to generate letters that are sent to policyholders and agents based on specific criteria for each letter. The system was put in place three years ago and although it serves its purpose much better than its predecessor, it has caused us many problems.

During our meeting my immediate supervisor and the person that supervises the other team of Business Systems Analysis (BSA) were expressing frustrations with the issues revolving around this letter generation system. The most concerning of the problems was related to our renewal maintenance system. Our policy processing system is a COBOL based system with fourth-generation tool built around it. Renewal maintenance is one of these has several roles, one of which is to create the data file that the renewal letters get their data from. It had gotten to the point that we were using too many of what we call "L" specs and it was taxing the COBOL resources on the entire system.

Multiple times over the three years we had been using this system the BSA's would be testing a new letter, and while running a COBOL utility program required for testing would lock up the system. We would then have to open a service ticket with our sister company, Berkley Technology Services (BTS). The answer we continually got back was "you have exceeded limits in renewal maintenance". This meant waiting sometimes several weeks for a resolution. As we discussed the issue more problems were brought out, those will be detailed later in this paper.

One of the ways we verified that our letters were attaching to the correct policies was to write a query against our premium data warehouse (know as DSS). The BSA group had grown knowledgeable using the SQL editor from Informix and it had become very easy to write a query that parallels the pre-condition set up in renewal maintenance. My suggestion was to forget out dealing with the renewal maintenance system and replace it with a query

against DSS that would check all of our renewal letters. Both my supervisor and the other team lead liked the idea so I started to do some research and see if it was feasible.

By the spring I had determined that this project was a go, but we would need input from our data warehouse team, the environment support team, the transaction programming team and the BSA group. Also we had to make sure the work flow of the mail room staff and print operations was not adversely affected. They also had to be in the loop on any changes pertaining to their duties. I starting to go forward with this project but because we already had a system in place, it was giving a low priority.

What the project became was a group of several deliverables. First there where the letters, they are broken into two types, policyholder notices which are mailed to policyholders and are for information purposes only, but required by law and renewal letters which are sent to the agents and sometimes policyholders, but they require some sort of response from the addressee. The system also included certificate of mail reports, which are required by the post office on legal mailing of this sort, they use them to verify the proper number of letters were sent. Since the renewal letters didn't need a certificate of mail we decided to create a verification list for them also, this is called a policy list.

We decided to utilize a tool called Hyperion Intelligence Explorer and the Hyperion Performance Suite to produce the certificate of mail, policy list and some of our letters for the collector car line. We phased out not only the renewal "L" specs but also a scheduled job using MS Access.

As the planning grew the main goals of this project centered around eliminating the renewal "L" specs, replacing the MS Access Database with Hyperion Intelligence and cleaning up several lingering issues with the existing system that were manipulating the time of most of the BSA staff. A final goal was improve the quality and efficient of our testing procedures.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Tom Shide

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

Background of the Problem

An insurance company is required to send written notification to policyholders when there is any change in coverage to a renewal policy or if premium is expected to increase over a threshold of three percent, these are called policyholder notices. In addition, the company being addressed with this project Continental Western Group (CWG) will send letters to agents or policyholders requesting information about an upcoming renewal to help with underwriting, these are called renewal letters. So CWG is sending letters to policyholders and agents on a daily basis.

CWG uses a COBOL based system for its policy processing called A+PLUS. The core system is maintained by programmers at a sister company called Berkley Technology Services (BTS). The custom designing and modification is done by CWG in-house by a group of Business Systems Analysts (BSA's). The BSA group uses fourth-generation COBOL tools to make the custom changes to the system. The changes are migrated to the production environment on a semi-monthly basis.

Prior to May of 2002 CWG was not fully automated in sending letters to policyholders and agents. At that time a decision was made to have Research & Development make a filing for each letter so it gets a unique ID and name. Maintenance of an automated process to create these letters was to be done by the BSA group. The advantage to this was to having it done in-house rather than relying on the COBOL programmers at BTS. This process allowed for a larger volume of letters to be maintained.

To create each letter a specification was designed by one of the BSA's in a fourth-generation COBOL tool called "Renewal Maintenance". This specification contained the conditions on which the letter will attach and pulled the necessary data out of the policy files on the A+PLUS system. All these specifications executed in the A+PLUS daily batch cycle to create a single file that was put onto the CWG network using file transfer protocol (FTP). A Microsoft Access macro accessed this file to create data sources used my Microsoft Word to create the letters. Access then created Certificate of Mail reports which are required by the

post office to verify bulk mailings. Lastly, the Certificate of Mail reports were moved to the company intranet for archiving.

Statement of the problem

As the number of letters filed by R&D grew so did the number renewal specifications required. This presented the first problem, a lack of COBOL resources. Multiple times since this system was put into place the limits for renewal specifications were exceeded on the A+PLUS system. This created a need for the core programming team at BTS to make some changes. It also means the BSA group has to constantly make sure that unused renewal specifications get deleted off the system as the renewal specifications are constantly as their maximum allowable size. Deleting these specifications goes against the business practice of CWG not to delete specifications due to the need to keep an audit trail on development. But it has become a necessity.

A second problem is the difficultly in programming the criteria for some letters. Because of the way the files are designed in A+PLUS some letters require multiple renewal specifications to program the attachment criteria. We have one letter that must attach to all liability coverage, to accomplish this ten renewal specifications had to be created.

The renewal specs do not always test out to be accurate in identifying the policies that require a letter. At various times since the system was introduced 17 different letters have had to be created by a manual process to insure they get attached to the proper policies. The are two members of the BSA staff that spend eight to ten hours weekly monitoring these letters that do not attach properly. On top of this the BSA group has a length process it must go through to test the renewal specs.

The role MS Access plays in the system is another problem because the company has made a decision to eliminate the use of it with any scheduled jobs. So we need to replace Access with a more powerful database tool.

There are ten letters created for CWG's collector car policyholders that have to be created manually because they print data specific to a vehicle. This works fine if a policy has one vehicle, but on policies with multiple vehicles the letter has to be directed to the word processing staff to be created manually.

Finally, the limitations of scheduling changes to the renewal specs have been an issue. CWG is allowed at most two custom releases per month, so if a problem is identified you might have to wait two weeks to get the change migrated to the test environment and another two weeks for it to get released to production. If the problem is urgent enough this schedule can be bypassed with exception processing but that requires special procedures and incurs extra cost and resources to the company.

Objectives of the project

The objectives of the project are designed to address the problems listed above plus add some functionality to the process. Specifically the project objectives are:

- Eliminate the use of the renewal specs
- Make the testing process more efficient, taking the responsibility for creating a test file away from the BSA group. Give them a better set of data to test against.
- Find a tool other than MS Access to generate the Certificate of Mail
- Find a tool other than MS Access to create the data sources for the letters.
- Shorten the window to time involved in getting a letter into production.
- Create a mechanism for verification of renewal letters. Only policyholder notices require a certificate of mail so the mail staff does not have a report to use for verification of the renewal letters.
- Train the BSA staff so they can continue with maintenance of the system.
- Eliminate the need for the manual corrections being done on the current system. There should be no need to create letters manually outside the system.
- Automate the collector car letters that print vehicle data so the word processing staff now longer has to create them manually.

CHAPTER 2 LITERATURE REVIEW

What are the corporate partners doing?

Continental Western Group is part of WR Berkley Corporation (WRBC). WRBC owns insurance companies all over the world in various segments of the market. The market that CWG competes in is regional market for property and casualty insurance. There are three other companies within WRBC in this regional market and they are referred to as the "Regional Segment" with WRBC.

The other three regional companies handle this issue in a variety of ways. One approach is to send mass mailings to all policyholders when there are filings for new coverage or change in coverage. This is an approach CWG used prior to implementing the current system. Another approach is to rely on the reporting features built into the A+PLUS system, CWG used this approach for a time also. Going forward CWG has made a commitment to staying current with filings from the Insurance Services Office (ISO) and wants to be conscientious about notifying policyholders so the A+PLUS reporting feature and a more generalized approach using mass mailings was rejected. A decision was make to improve our current system and continue to identify policies on an individual basis through a daily job.

Continental Western has decided to take the lead within the corporation in developing this system. The other three companies will have a chance to review CWG's system when it is complete.

With the A+PLUS system being phased out in the coming years there is also a need utilize an external data warehouse instead of the A+PLUS files.

Policyholder Notices and Renewal Letters

There are two types of letters sent by CWG. Policyholder notices are sent directly to policyholders and require no response. Renewal Letters are sent either to the agent to the policyholder and require a response back to the company. Policyholder notices and renewal letters are both referred to as "letters" in practice and in this document. Table 1 outlines the differences between these two letters.

Table 2.1: Differences between policyholder notices and renewal letters

Policyholder Notice	**Renewal Letter**
Mailed directly to the policyholder	Can be mailed to the policyholder or the agent
Do not require a response from the addressee	Require a response to the company in the form of underwriting data, approvals for optional coverage, etc.
Are not sent with a return envelope	Are accompanied by a return envelope
Require a Certificate of Mailing	Do not require a Certificate of Mailing
For agent reference a prototype copy is posted to the company intranet	For proof to the underwriter that it has been sent a copy is scanned into the imaging file for the policy.
R&D files the form with ISO and assigns a unique name beginning with the letters 'PN'	R&D does not file the letter but it does give it a unique name beginning with the letters 'UW'
In force for a finite period of time, usually one year.	Generally are on-going but in some cases they will print for a finite period of time.

Definitions

The **System Under Discussion** (SUD) is the policyholder notice and renewal letter generation system being developed with this project.

Certificate of Mail is a report generated by the systems at Continental Western Group. Its format is approved by the post office and used to verify bulk mailing of policyholder notices. For a sample see Figure 1 in Appendix B.

A **Policy List** is the equivalent to the Certificate of Mail for Renewal Letters. It is a list identifying every policy for every renewal letter. It is created daily and posted to the intranet. Unlike the Certificate of mail, the policy list is not used by the post office and is for internal records only.

A **Business Systems Analyst** (BSA) is an IT professional at CWG that specializes in making custom changes to the A+PLUS policy system. The BSA team is responsible for programming and testing the policyholder notices and renewal letters.

Berkley Technology Services (BTS) is a corporate partner of CWG. Also part of WR Berkley Corp, it is an IT firm that serves the technology needs of all WR Berkley companies.

The policy processing system used by CWG is known as **A+PLUS**, it is a COBOL based system whose core programs are maintained by programmers at BTS.

The files that maintain the history of the A+PLUS system are called **PREPMSTR**. There are 33 PREPMSTR files that are updated in the daily batch processing, these 33 files hold the full history of all policies processed on the A+PLUS system.

The **Network Operating Center** (NOC) is the IT center for WR Berkley Corporation. Production Servers for all WRBC companies are located on this site. The NOC is located in Wilmington, Delaware.

The **Renewal Maintenance System** is a fourth-generation tool that is part of the A+PLUS tool set. It is programmed by the BSA staff at CWG to perform several functions. One of these functions is creating a file with all the data necessary for printing policyholder notices and renewal letters. This file is called **RENPRT**. Other functions of renewal maintenance include deleting records from a policy file at renewal time, pre-filling data items at renewal time, adding forms and deleting forms.

Renewal L-Specs are the specifications in the renewal maintenance system that interrogate the PREPMSTR files on A+PLUS to determine if a record should be created on the RENPRT file.

DSS is the data warehouse for the A+PLUS policy system, it was developed and is maintained by BTS. DSS gets its data directly from the A+PLUS PREPMSTR files, through a process run in the daily batch cycle. The data for the SUD is created by a query against DSS.

The **Loss Data Repository** (LDR) is the corporate claims warehouse created and maintained by BTS from the claims system, known as Claims Workstation (CWS). It is referenced here only because it is part of the definition for STATLITE, not other mention of this system is needed as related to the SUD.

STATLITE is the CWG data warehouse, its core is created out of DSS and the LDR to provide financial data to the company. However, plans are under way to expand it so the policyholder notice and renewal letter data can be generated from it. STATLITE also contains data pulled from other systems such as claims, imaging, accounting and application tracking. It runs on a SQL Server 2000 platform and will hold the tables used in our SUD.

The **Data Warehouse Team** is a group of IT professionals at CWG responsible for maintaining and developing the corporate data warehouse and the scheduled processes that use it as well as the LDR and DSS. The Data Warehouse Team is responsible for the scheduled jobs used by the SUD.

The cw-dsm-datamart server, referred to as "**datamart**", is the development server used by the Data Warehouse team. On this server all the extract, transform and load (ETL) jobs are created.

The BTSDESQL27 server, referred to as "**SQL27**", is the production server on which the CWG data warehouse resides. Scheduled jobs are migrated here from datamart after they pass testing.

The **Master Query** is a query written against DSS that identifies all the policy data needed to print the policyholder notice and renewal letters.

The **cwg_work** database is located on the datamart and SQL27 servers and contains the tables used in this SUD. It is considered part of STATLITE.

The table **phn_policy_data** contains policy specific data used to populate the variable fields on the policyholder notices, renewal letters, and certificate of mail and policy list. It is part of the cwg_work database.

The table phn_letter_data contains information about all the policyholder notices and renewal letters. This information is used by the Master Query to find policies that need to be have letters sent and also to calculate the status of each letter. It is part of the cwg_work database.

The **Environment Support Group** (ESG) is a team if IT professionals at CWG responsible for the migration of changes from development to production for all systems used by CWG. For policyholder notices and renewal letters they will move changes to the daily processes and to the master query and letter spreadsheet.

The **Policyholder Notice Coordinators** are two members of the BSA group that are trained in more detail on the policyholder notice and renewal letter process. Their responsibility in this area expands beyond the rest of the BSA group to include troubleshooting issues on the production system; working with the ESG team to get changes migrated to the production environment and providing information on each letter to the mail center and print operations when a new letter is created.

The **Administrative Services Team** is a group of word processing and documentation professional that create the specimen copies of all the policyholder notices and renewal letters.

The **Distribution Services Team** is the group of mail service and the print operations professionals. The print operations technician will receive an email notification every morning after the data is ready, he or she will then start a job that prints all the letters, certificate of mail and policy list reports. The print is delivered to the mail room where all the letters are stuffed and crossed checked against the certificate of mail and policy list.

The **Report Management System** (RMS) is a reporting system developed in house by CWG. It is an area on the CWG Intranet where reports of various origins are placed here. Reports are created daily, weekly and monthly. The categories of reports fall into one of the following groups: Accounting, Agency, Claims, DSS, Miscellaneous, Operations, Reinsurance and Underwriting. The certificate of mail and policy list will be posted here on a daily basis.

Tools

IBM Informix SQL Editor is an SQL editing tool that is used to build the Master Query. The query was written against the DSS database, which is on the Informix platform.

Microsoft SQL Query Analyzer is an SQL editing tool used with SQL Server 2000. The query tool was used to format data created by the DSS query after it was stored on the data warehouse.

Microsoft SQL Server Enterprise Manager is the administrative tool for SQL Server. We used it to create the table phn_policy_data and phn_letter_data.

Microsoft Word is the word processor used to create the policyholder notices and renewal letters.

Microsoft Query is a query tool internal to MS Word, it was used to link each policyholder notice and renewal letter to the cwg_work database on datamart and SQL27 and create the data source.

Hyperion Intelligence Explorer is the corporate stand database reporting tool. It was used to create the Certificate of Mail, the Policy Lists and ten collector car letters that print vehicle data. It was necessary to create the collector car letters here because MS Word could not handle policies that have multiple vehicles insured. Under the previous system the Administrative Services team had to create these manually.

Hyperion Performance Suite is also called the "I - Server", it is a web-based tool used to create scheduled jobs from documents created on Hyperion Tools. The I - Server was used to schedule daily jobs for the Certificate of Mail, Policy List and the some collector care renewal letters. Another job was created to run daily to move the output to the Intelligence FTP server.

Microsoft XML Notepad is an XML editing tool used to create two XML files. One file was used on the I - Server job that moved output from the development staging area to the CWG network. The second file moves the output from the production area to the CWG network.

Pervasive Process Designer is referred to by CWG IT personnel as Data Junction because that was the vender name when we started using it. This is an ETL tool used to run the daily scheduled job that creates the phn_policy_data and phn_letter_data tables on

cwg_work. It also emails the print operations staff signaling it is OK to start the job that prints the letters.

Pervasive Map Designer is the map designing tool used by the Pervasive Process Designer. Our process to create the phn_policy_data and phn_letter_data tables uses three maps.

The **Embarcadero Job Scheduler** is a tool used by CWG to schedule jobs. This project will have two jobs scheduled with Embarcadero, one to run the process to create the data and another to print everything.

The Current System

As of today (11/17/2006) there are 14 policyholder notices active on the system with two more in a pending status, meaning they are implemented, but they have not yet started to print. There are 17 underwriting letters currently active. See table # in Appendix B for a complete list.

One of the issues from the problem statement above was the difficult in programming criteria for some forms. Using data from 11/17/2006 this will be illustrated. Of the 16 policyholder notices, two of them PN6010 and PN6113 are being created manually on a process outside of the current system. Of the 17 renewal letters, five are created manually. This is due to limitations of the renewal maintenance system in A+PLUS. Table 2 shows the results of the printing from 11/17/2006.

Table 2.2: Data Verification results from 11/17/06.

Forms That Printed Automatically	Number Printed	Forms that had to be created manually	Number Printed
PN6118	56	PN6010	1
PN6119	10	PN6113	17
PN6120	1	UW0020	0
PN6121	45	UW0021	1
PN6128	6	UW0022	1
PN6130	2	UW0037	2
PN6131	6	UW0040	3
PN6137	1		
UW0013CC	2		
UW0026CC	1		
UW0033	2		
Total Letters Printed	183	Total Letters Printed	25

If a policyholder notice or renewal letter does not show in Table 2 it means there was no print generated for it that day. So of the 16 policyholder notices, eight printed off the system, two had to be generated manually, two more are pending and the other four did not hit on any policies this day. Of the 17 renewal letters, five have to be created manually, three printed automatically as they should and the other nine did not hit on any policies. It is important to note that UW0013CC and UW0026CC are collector car letters that print vehicle data. It was mentioned above that if these letters hit a policies with multiple vehicles the system kick them out in error, forcing the Administrative Services group to create them manually, this was not the case on this day.

Table 2 shows that 208 pieces of mail went out the door on 11/17/2006, but 25 of them (12%) had to be created manually. The new system had addressed the problems with all these letters and had as an object to get the ratio down to zero percent. The two members

of the BSA staff that serve as Policyholder Notice coordinators now spend eight to ten hours weekly on issues related to the letters that are created manually.

Here is a summary of how the system functions today. The driver behind the creating of the RENPRT file is a program called RNWLP001. This program executes all the renewal specifications. As mentioned above the renewal maintenance system uses the L-specs to create the RENPRT file, however there are several other types of renewal specs in renewal maintenance, see table 3 for a detailed list.

Table 2.3: Type of A+PLUS Renewal Specifications

Type of renewal spec	**Number of specs**	**Description**
L-Specs	145	Used to create reports, build the RENPRT file
Diary Specs	375	Used to create messages in the A+PLUS diary system
Form Specs	400	Used to add and delete forms from a policy
Data Spes	499	Used to pre-fill field on the A+PLUS screen at renewal time
R-Specs	103	Used to delete records from the A+PLUS policy files, re-sequence items on A+PLUS screens and sort.
S-Specs	1	Used to stop processing according to a pre-condition
Total	1,523	

The data specs outnumber the rest but they are very simple in structure and only few of them (less than 10) call COBOL subroutines so they take very little processing time. The form specs are also very simple and none of them use COBOL routines. The R-specs take even less significant when it comes to processing time. The diary specs each call COBOL subroutines but they have caused little or no trouble, which leaves the L-specs taking up the most resources when RNWLP001 is run.

At 6:00 P.M. central time Monday through Friday the A+PLUS system is taken down so the daily batch processing can be done. One of the programs run in the batch cycle is RNWLP001. During the execution of RNWLP001 each L-Spec will check the expiration dates for each policy on the PREPMSTR files and then compare the rest of the criteria on the pre-condition. If the precondition is satisfied a record is created in the RENPRT file. After RNWLP001 completes the RENPRT file is moved to the CWG network using file transfer protocol (FTP). Between 7:00 and 7:30 A.M. the technician in the print operations center at CWG will run a file called filechk.bat. This file starts a macro in MS Access that reads the RENPRT file, builds tables for every letter to be used as a data source and creates a Certificate of Mail for all the policyholder notices. After Access closes a macro is started in MS Word to send all the letters to the printer. After filechk.bat completes the print technician starts a file called RENPCOM.bat. This file transfers all the Certificate of Mail reports to a staging area so they can be posted to the CWG intranet.

The print technician does not get a notification to run these jobs, it is built into the job requirements. See Figure 1 on the next page for an illustration of the current system.

Current Letter Generation System

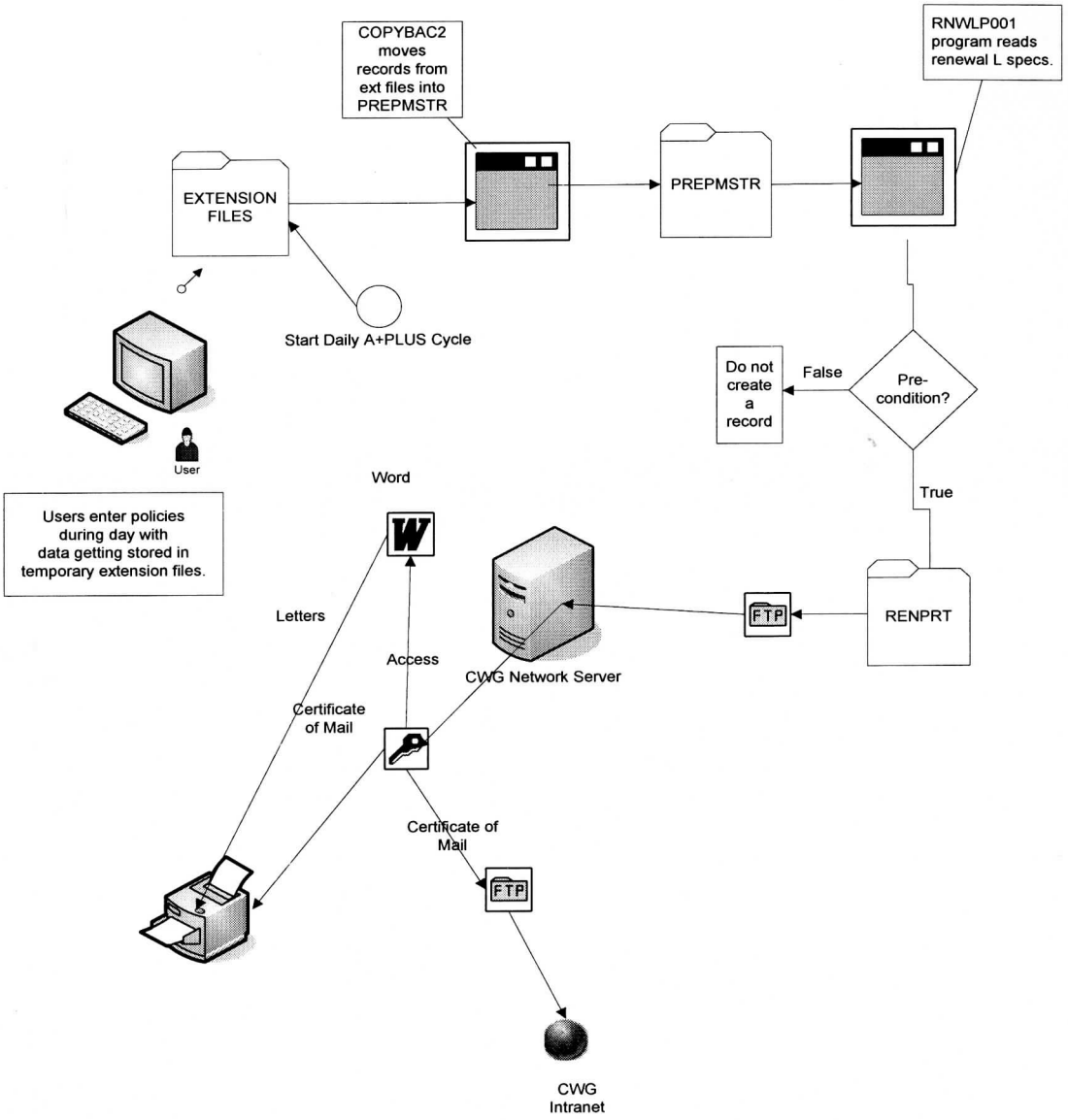

Figure 2.1: Illustration of the current letter generation system used by CWG

CHAPTER 3 SYSTEM DESIGN

Analysis Strategy

The analysis strategy will be addressed according to the components of the project, these include the data used to create the letters, certificate of mail and policy list; the letters created in MS Word; the certificate of mail created for the policyholder notices; the policy list created for the renewal letters; the collector car letters that print vehicle data and the letter status report.

Data

Where to store the data and how to create it was the first issue tackled with this project. The method currently used to collect the data was the main reason for changing the system, so continuing to use the PREPMSTR files as the source for the data was not an acceptable option. This left two possible sources to get the data, the CWG data warehouse and DSS. The CWG data warehouse does not have the detail of data required to program the attachment criteria for all the letters. The DSS system does have this data because in addition to the premium transaction information loaded into DSS daily, there is also a copy of the PREPMSTR files loaded into DSS daily. So any data accessible on A+PLUS is also accessible on DSS.

There was a third option to creating the data, to run a query against DSS using Hyperion Intelligence Explorer and scheduling a daily job. The attraction to the Hyperion tool was the fact that the Hyperion server is located at the NOC, as is the DSS server (cwgpn02) so the data would not have to be pulled across the Wide Area Network (WAN). Research on the query capabilities of Intelligence showed it did not have the flexibility to pull in outside text files or create temporary tables like we do with SQL Editor. Hyperion also ran much slower. Since plan to run the DSS query on off hours the problem with pulling data over the WAN will end up not being an issue.

So the decision was made to go ahead and use an Informix query against DSS to get the data needed to generate the letters.

Looking to the future CWG is in the process of expanding its data warehouse to include more detail. When this project is complete the process will be changed so the process is run against the data warehouse instead of DSS. However it was decided not to wait for the data warehouse project as it is too far in the future to hold up the system we are discussing here.

In addition to the query against DSS, an ELT tool was needed because the data must be moved from the server on which DSS resides to the server on which the data warehouse exists. CWG currently uses the Pervasive Process Designer (formerly Data Junction) as the ELT tool when data is moved from Informix (the DSS database vender) to the warehouse. See table 4 for a description on tasks involved with the ELT job.

Letters

When it was first decided to move forward with this project the desire was to create the letters using Hyperion Intelligence Explorer. The reason was the Hyperion Server is located at the NOC along with the DSS server (cwgpn02), so we would not have to be pulling the data across the Wide Area Network (WAN). After researching the reporting capabilities of Intelligence it was decided that it did not provide the formatting capability we need for the policyholder notices and for any of the renewal letters that were more than one page. It was found to be an excellent reporting tool but not a sufficient word processing tool.

So all the letters with the exception of ten collector car letters will be created using MS Word. Another benefit to using Word is that the letters are already created. The only changes needed are to rename them, move them to a new location and reset the data source.

Hyperion Intelligence Explorer did prove out to be the best tool to create our collector car letters that return vehicle data. Because these letters more resemble a report than a letter, using a reporting tool like Explorer made the most sense.

Certificate of Mail

The current system used MS Access to create the certificate of mail reports for each letter. Because one of the objectives of this project is to eliminate the use of Access a change was necessary. Hyperion Intelligence Explorer is clearly the best tool we have for this job as it is our corporate standard for generating reports. Using Explorer will also reduce the

development time for future projects. On the current system the BSA has to create a new certificate of mail report for each new letter. We have gained several advantages with Explorer as our tool to create the certificate of mail, they include:

- We no longer have to create a new certificate of mail report for every new policyholder notice. Once the initial job is set up each new letter will automatically get added to the certificate of mail job.
- We were able to set up the report in Explorer so seven policies could print on one page. The most we could get on the reports from Access were three. This will reduce the amount of pages the mail room has to deal with by more than 50%.

Policy Lists

The policy lists are a new component that the existing system does not have. Because they will be formatted similar to the certificate of mail it was an easy decision to use Hyperion Intelligence Explorer to create these.

Letter Status Report

The letter status report is another component the existing does not have. The distribution services staff does keep an excel spreadsheet with information they need from all the letters, however it is a spreadsheet that is manually updating and sometimes the updates do not get made in a timely manor. So the decision came down to a choice between continuing as we are now are upgrading this part of the process.

We are going ahead with creating a report in Explorer using the Excel spreadsheet the BSA group uses to submit their jobs. The advantages to using Explorer include

- We will create a job that updates this report daily, the report does not need to be delivered daily but since the certificate of mail, policy lists and other Hyperion jobs are done daily it made sense to do this daily. Also, the status of a letter can change from one day to the next.
- We will be able to give the distribution staff more data. We will now have a 'status' fields that says 'Active', 'Pending' or 'Expired' so the mail and print staff can quickly see the status of a letter. There will also be a 'Start Print'

and 'End Print' date so the mail and print staff can know exactly when a letter will stop printing.

- The report will be more accurate because its source, the letter spreadsheet updated by the BSA staff is quality checked and has to be accurate. There is no quality checking in place with the existing spreadsheet.

Requirements Analysis

Table 3.1: In/out list

Task	In	Out
Policyholder notices and renewal letters print automatically from a daily scheduled job.	X	
Create an electronic, scanned image of each renewal letter		X
Create a certificate of mail report for each policyholder notice	X	
On each letter print a sequencing number that can be cross checked against the same number on the certificate of mail or policy list	X	
Modify the look of the certificate of mail to match the form on file with the post office, PS Form 3877	X	
Create a report summarizing the status of all policyholder notices and renewal letters	X	
Create a list, similar to the certificate of mail that can be used for verification of renewal letters	X	
Post certificate of mail and policy lists to the CWG Intranet on a daily basis	X	
Email notifications specific to errors in the master query		X

Table 3.2: Primary Actors

System Under Design (SUD)	The system will start all the jobs automatically so it will be the primary actor for all use cases.

Table 3.3 – Secondary Actors

DSS	As mentioned above this is the corporate data warehouse for the A+PLUS policy system, it uses Informix as its database platform.
CWG data warehouse	This is a SQL server data base that CWG populates daily from DSS and other systems. The policy data and letter data generated by the master query are stored here.
Hyperion Explorer	There are Hyperion Explorer files created for the certificate of mail, the policy list, the letter status report and the collector car letters.
I – Server	The I –Server is the tools used to schedule job to run on Hyperion Intelligence.

Table 3.4 – Stakeholders

Business System Analysts (BSA)	Group of IT professionals at CWG that will be responsible updating and maintaining the system. The two BSA's that serve as policyholder notice coordinators take on a more active role.
Print operations staff	Part of the distribution services team, the print operations staff monitors and run the central printer in CWG's home office, they also deliver the print to the mail staff.
Mail staff	Part of the distribution services team, the mail staff is responsible for all mail that goes out the door. They run the machines that stuff the envelopes, assign the postage and they also very that all pieces of mail are accounted for.
Data warehouse team	This project was implemented by the data warehouse team so going forward they will address any issues with the scheduled jobs in Embarcadero or the I - Server.
Environment support team	The ESG team maintains the jobs run on Embarcadero and will be responsible for merging changes to the master query, letter spreadsheet from development to production.
Research & Development	The Research & Development team is at the very beginning of this process. When they file for a new coverage they also file the number for the form and work with the BSA staff to get the project scheduled.
Home office underwriting staff	The home office underwriting staff oversees underwriting over all four regions of CWG. They are responsible for the content of the letters and act as a contact person for the BSA that is developing a new policyholder notice or underwriting letter
Regional Underwriting staff	The regional underwriting staff is the ones that work the policies, they are assigned to a specific state or product line and consist of a rater, underwriting assistant and underwriter. They have to field any questions that come from the policyholder or agent and they collect the data sent to them via a renewal letter.
Administrative Services	The word processing team falls into this group. They work with the home office underwriter to design a specimen copy of the letter, then send it to the BSA to get the variable data loaded into it.

Policyholder	The policyholder is the owner of an insurance policy with CWG. They will receive a policyholder notice in the mail if they policy satisfies the criteria.
Agent / Agency	The agent or agency is the person or persons that broker the insurance contract between CWG and the policyholder. Renewal letters normally are sent to the agent with a request for information to be returned to CWG.

Table 3.5: Actor/Goal List

System Under Design	Automatically print policyholder notices and renewal letters as part of a scheduled job.
System Under Design	Create a certificate of mail report for each policyholder notice
System Under Design	Print a sequencing number for each letter on the letter and on the certificate of mail or policy list so the mail team can verify the print.
System Under Design	Modify the look of the certificate of mail to match the form on file with the post office, PS Form 3877
System Under Design	Create a report summarizing the status of all policyholder notices and renewal letters
System Under Design	Create a list, similar to the certificate of mail that can be used for verification of renewal letters
System Under Design	Post certificate of mail and policy lists to the CWG Intranet on a daily basis

Use case 1 – create the system data

Table 3.6: Use Case 1 – Characteristic Information

Description	This use case will create the data needed to print the letters, certificate of mail, policy lists and letter status report. Two tables will be created on the CWG data warehouse, one for policy data that will generate the letters, certificate of mail and policy list. The other table will have data specific to the status of each letter, it will be used to generate the letter status report.
Primary Actor	System under design, Embarcadero Scheduler
Secondary Actor	DSS
Stakeholder(s)	Data warehouse team, BSA team, Environment Support Group
Trigger	The Embarcadero scheduler starts the Pervasive ELT job.
Precondition	The daily A+PLUS cycle has completed successfully
Guarantees	The policies identified will be in force and match the precondition the associated letter
Success End Condition	The table tables on the CWG data warehouse created by this system will get deleted and replaced by a new version.
Failed End Condition	Noting happens

Table 3.7: Use Case 1 – *MAIN SUCCESS SCENARIO*

1.	The system sends an email notification to the data warehouse team and the policyholder notice coordinators indicating the process have begun.
2.	The system maps the status information on all the letters to a text file called letter.txt.
3.	The system uses the letter.txt file to update the status of each letter and update the letter status table on the CWG data warehouse.
4.	The system uses the letter.txt file to determine the date criteria for each letter.

5. The system compares the policy expiration date in DSS to the date criteria found in #4
6a. The system saves the policy data in a temporary file. [repeat at 5]
7. The system compares the policy attributes in DSS to the specific criteria for a letter.
8a. The system creates a record in the policy data table on the CWG data warehouse. [repeat at 7]
9. The system sends an email the data warehouse team and the policyholder notice coordinators indicating that the process has completed.

Table 3.8: Use Case 1 – *EXTENSIONS*

*a. An error is encountered during the ELT job.
*a1. The system sends an email to the data warehouse team and the policyholder notice coordinators indicating the process failed. [fail]
6b. The system bypasses the policy. [repeat at 5]
8b. The system does not create a record in the policy data table. [repeat at 7]

Table 3.9 – Create the system data rules

Number	**Rule Description – The system shall …**
1	Run the ELT process until 7:00 a.m., Monday through Friday
2	Process data one day behind because DSS is updated one time daily
3	Group the data for Saturday, Sunday and Monday into Monday's job.

Use case 2 – Create the Certificate of Mail

Table 3.10: Use Case 2 – Characteristic Information

Description	This use case will create the certificate of mail from the system data tables created in use case 1.
Primary Actor	System under design
Secondary Actor	CWG data warehouse, Hyperion Explorer, I - Server.
Stakeholder(s)	Data warehouse team, BSA team, ESG Team
Trigger	The I - Server starts the certificate of mail scheduled job.
Precondition	The policy and letter tables on the CWG data warehouse created in use case 1 have been updated.
Guarantees	The policies listed on the certificate of mail will have a corresponding letter created and can be verified by the sequence number.
Success End Condition	The certificate of mail report will be placed in PDF format in the 'PHN' folder on the Hyperion Intelligence Server.
Failed End Condition	Noting happens

Table 3.11: Use Case 2 – *MAIN SUCCESS SCENARIO*

1.	The system opens the certificate of mail query file in Hyperion Explorer.
2.	The certificate of mail query file processes a query against the system data files created on the CWG Warehouse.
3.	The certificate of mail query file uses the query results to create the certificate of mail report.
4.	The certificate of mail query file runs JavaScript code to export the certificate of mail as a PDF file to the 'PHN' folder on the Hyperion Intelligence Server located at the NOC.
5.	The I – Server closes the certificate of mail query file and sends email notification to the data warehouse team and policyholder notice coordinators.

Table 3.12: Use Case 2 – *EXTENSIONS*

*a.	An error is encountered during either I - Server Job.
	*a1. The system sends an email to the data warehouse team and the policyholder notice coordinators indicating the process failed. [fail]

Table 3.13 – Create the certificate of mail rules

Number	**Rule Description – The system shall …**
1	Not start use case 2 until use case 1 is completed

Use case 3 – Create the Policy List

Table 3.14: Use Case 3 – Characteristic Information

Description	This use case will create the policy list from the system data tables created in use case 1.
Primary Actor	System under design
Secondary Actor	CWG data warehouse, Hyperion Explorer, I - Server.
Stakeholder(s)	Data warehouse team, BSA team, ESG Team
Trigger	The I - Server starts the policy list scheduled job.
Precondition	The policy and letter tables on the CWG data warehouse created in use case 1 have been updated.
Guarantees	The policies listed on the policy list will have a corresponding letter created and can be verified by the sequence number.
Success End Condition	The policy list report will be placed in PDF format in the 'PHN' folder on the Hyperion Intelligence Server.
Failed End Condition	Noting happens

Table 3.15: Use Case 3 – *MAIN SUCCESS SCENARIO*

1.	The system opens the policy list query file in Hyperion Explorer.
2.	The policy list query file processes a query against the system data files created on the CWG Warehouse.
3.	The policy list query file uses the query results to create the policy list report.
4.	The policy list query file runs JavaScript code to export the policy list as a PDF file to the 'PHN' folder on the Hyperion Intelligence Server located at the NOC.
5.	The I - Server closes the policy list query file, ends the job and sends email notification to the data warehouse team and policyholder notice coordinators.

Table 3.16: Use Case 3 – *EXTENSIONS*

*a.	An error is encountered during either I - Server Job.
	*a1. The system sends an email to the data warehouse team and the policyholder notice coordinators indicating the process failed. [fail]

Table 3.17 – Create the policy list rules

Number	**Rule Description – The system shall …**
1	Not start use case 3 until use cases 1 and 2 are completed

Use case 4 – Create the collector car vehicle letters

Table 3.18: Use Case 4 – Characteristic Information

Description	The collector car vehicle letters are renewal letters that print the year, make, model and serial number of every eligible vehicle on a policy. Because policies can have multiple vehicles these letters are treated as reports. This use case is repeated for ten different letters
Primary Actor	System under design
Secondary Actor	CWG data warehouse, Hyperion Explorer, I - Server.
Stakeholder(s)	Data warehouse team, BSA team, ESG team
Trigger	The I - Server starts the collector car scheduled job.
Precondition	The policy and letter tables on the CWG data warehouse created in use case 1 have been updated.
Guarantees	The sequencing number on each letter will match a sequencing number of the policy list.
Success End Condition	The collector car letters will be placed in PDF format in the 'PHN' folder on the Hyperion Intelligence Server.
Failed End Condition	Noting happens

Table 3.19: Use Case 4 – *MAIN SUCCESS SCENARIO*

1.	The system opens the collector car query file in Hyperion Explorer.
2.	The collector car query file processes a query the checks the name of the letter against the specific collector car letter name.
3a.	The collector car query file adds a record to the results. [repeat at 2]
4	The collector car query file uses the result set to create a report that is actually a set of letters.
5.	The collector car query file runs JavaScript code to export the letters as a PDF file to the 'PHN' folder on the Hyperion Intelligence Server located at the NOC.

6.	The I - Server closes the collector car query file, ends the job and sends email notification to the data warehouse team and policyholder notice coordinators.

Table 3.20: Use Case 4 – *EXTENSIONS*

*a.	An error is encountered during either I - Server Job.
	*a1. The system sends an email to the data warehouse team and the policyholder notice coordinators indicating the process failed. [fail]
3b.	The collector car query file does not add a record to the result set. [repeat at 2]

Table 3.21 – Create the certificate of mail rules

Number	**Rule Description – The system shall …**
1	Not start use case 4 until use cases 1, 2 and 3 are completed

Use case 5 – Create the letter status report

Table 3.22: Use Case 5 – Characteristic Information

Description	This use case will create the letter status report from the system data tables that were created in use case 1.
Primary Actor	System under design
Secondary Actor	CWG data warehouse, Hyperion Explorer, I - Server.
Stakeholder(s)	Data warehouse team, BSA team, ESG Team
Trigger	The I - Server starts the policy list scheduled job.
Precondition	The policy and letter tables on the CWG data warehouse created in use case 1 have been updated.
Guarantees	All active and pending letters will be accounted for on this list.
Success End Condition	The letter status report will be placed in PDF format in the 'PHN' folder on the Hyperion Intelligence Server.
Failed End Condition	Noting happens

Table 3.23: Use Case 5 – *MAIN SUCCESS SCENARIO*

1.	The system opens the letter status query file in Hyperion Explorer.
2.	The letter status query file processes a query against the system data files created on the CWG Warehouse.
3.	The letter status query file uses the query results to create the letter status report.
4.	The letter status query file runs JavaScript code to export the letter status report as a PDF file to the 'PHN' folder on the Hyperion Intelligence Server located at the NOC.
5.	The I - Server closes the query file, ends the job and sends email notification to the data warehouse team and policyholder notice coordinators.

Table 3.24: Use Case 5 – *EXTENSIONS*

*a.	An error is encountered during either I - Server Job.
	*a1. The system sends an email to the data warehouse team and the policyholder notice coordinators indicating the process failed. [fail]

Table 3.25 – Create the letter status report rules

Number	**Rule Description – The system shall …**
1	Not start use case 5 until use cases 1, 2, 3 and 4 are completed

Use case 6 – Print all documents

Table 3.26: Use Case 6– Characteristic Information

Description	All the documents created on Hyperion Intelligence Explore get exported to the 'PHN' folder on the Hyperion Intelligence server. This use case moves these files to the CWG network so they can be printed locally.
Primary Actor	System under design
Secondary Actors	I-Server, Embarcadero, CWG Data Warehouse, CWG Intranet
Stakeholder(s)	Data warehouse team, BSA team, ESG Team, Home Office Underwriting staff, Regional Underwriting Staff, Print Staff, Mail Staff, Research & Development, Policyholder, Agent,
Trigger	The I - Server starts the FTP scheduled job.
Precondition	Use cases 2, 3, 4 and 5 have completed successfully
Guarantees	
Success End Condition	The certificate of mail, policy list, letter status report and all collector car vehicle letters are move to the appropriate folder on the CWG network with current date's timestamp.
Failed End Condition	Noting happens

Table 3.27: Use Case 6 – *MAIN SUCCESS SCENARIO*

1.	The system opens the XML file set up to move the documents located in the 'PHN' file
2.	The system moves the files in the 'PHN' folder on the Intelligence Server to the Hyperion Intelligence FTP server.
3.	The I – Server closes the XML file, ends the FTP job and sends email notification to the CWG data warehouse team and the policyholder notice coordinators
4.	Embarcadero starts a Pervasive ETL Process and sends email notification to the CWG data warehouse team and the policyholder notice coordinators indicating that the job has started.

5.	The system performs two FTP processes, one to a location on the CWG network, the other to the Report Management System staging area on the CWG Intranet.
6.	The system prints the PDF files located in the 'PHN' folder.
7.	The system starts a macro that opens the first MS Word Document
8.	A .dns file connects the MS Word document to its data source on CWG Warehouse (server SQL 27).
9.	The macro pulls the data into the document, sends the document to the printer, then closes the document. [repeat at 7]
10.	The ETL job sends an email notification to the CWG data warehouse team and the policyholder notice coordinators indicating that the job completed.

Table 3.28: Use Case 6 – *EXTENSIONS*

*a.	An error is encountered during either I - Server Job.
	*a1. The system sends an email to the data warehouse team and the policyholder notice coordinators indicating the process failed. [fail]
*b	An error is encountered during the ELT process
	*b1. The system sends an email to the data warehouse team and the policyholder notice coordinators indicating the process failed. [fail]

Table 3.29 – Print all documents rules

Number	**Rule Description – The system shall …**
1	Not start use case 6 until use cases 1, 2, 3, 4and 5 are completed
2	Have the timing worked out on the I – Server and Embarcadero so the files are moved to the FTP Server on Hyperion Intelligence before Embarcadero starts the job to print the documents.
3	The I – Server will start the FTP job at 7:15 a.m.
4	Embarcadero will start the ELT process at 7:30 a.m.

Illustration of the system under discussion

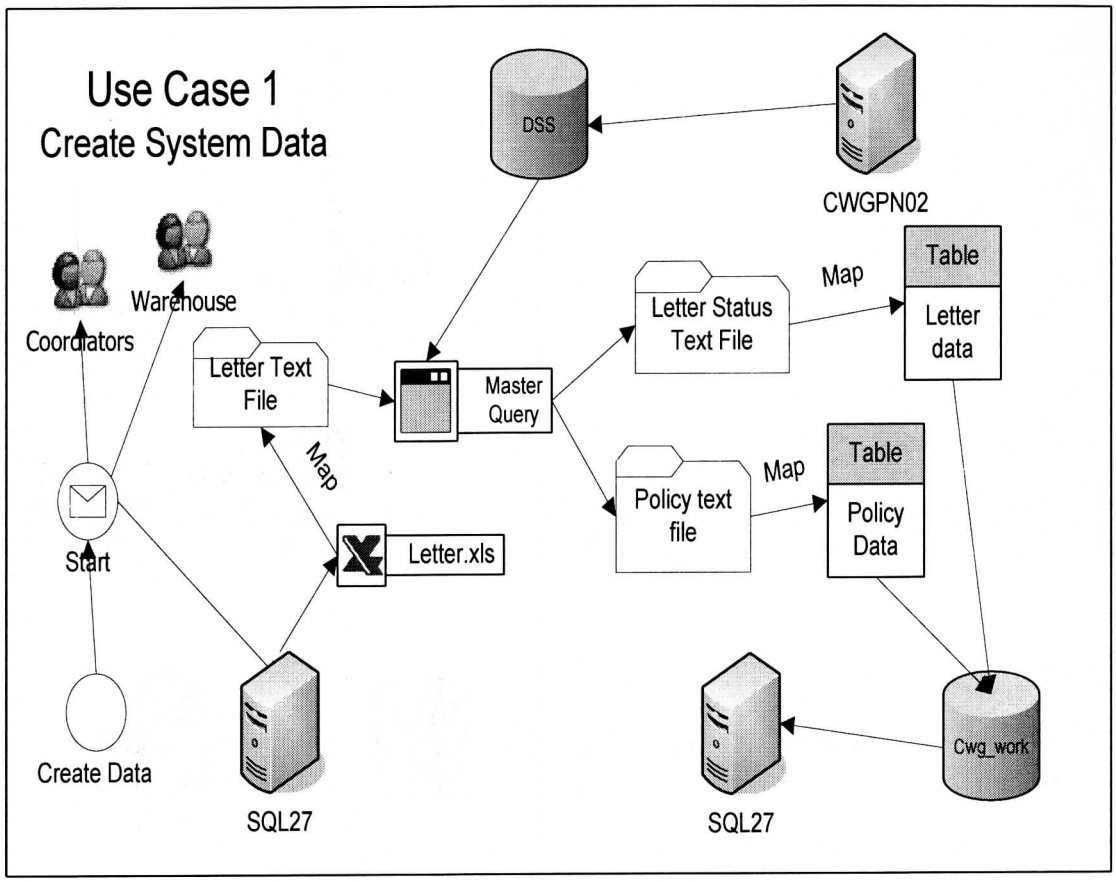

Figure 3.1: Use Case 1 – Create the system.

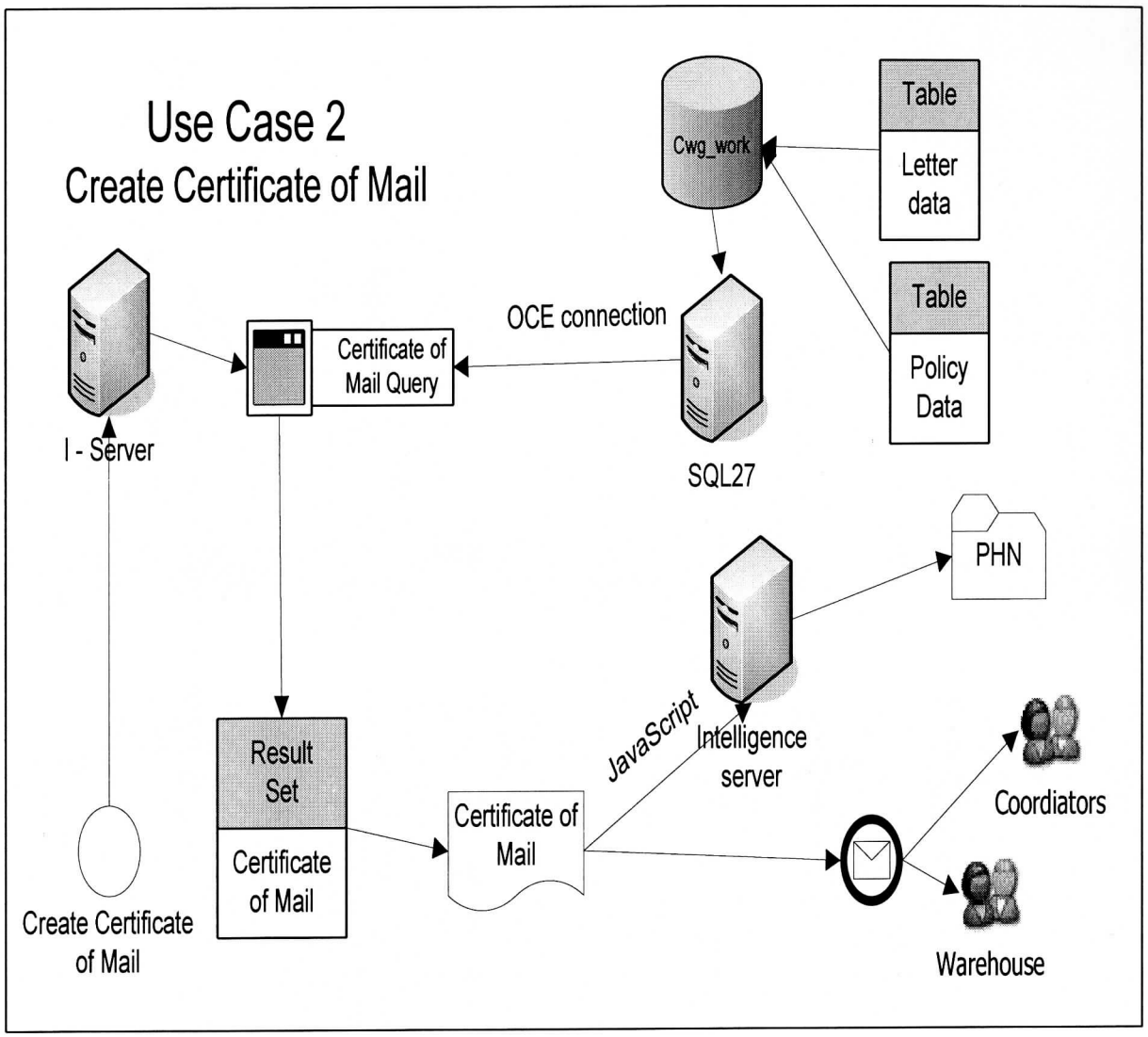

Figure 3.2: Use Case 2 – Create the Certificate of Mail

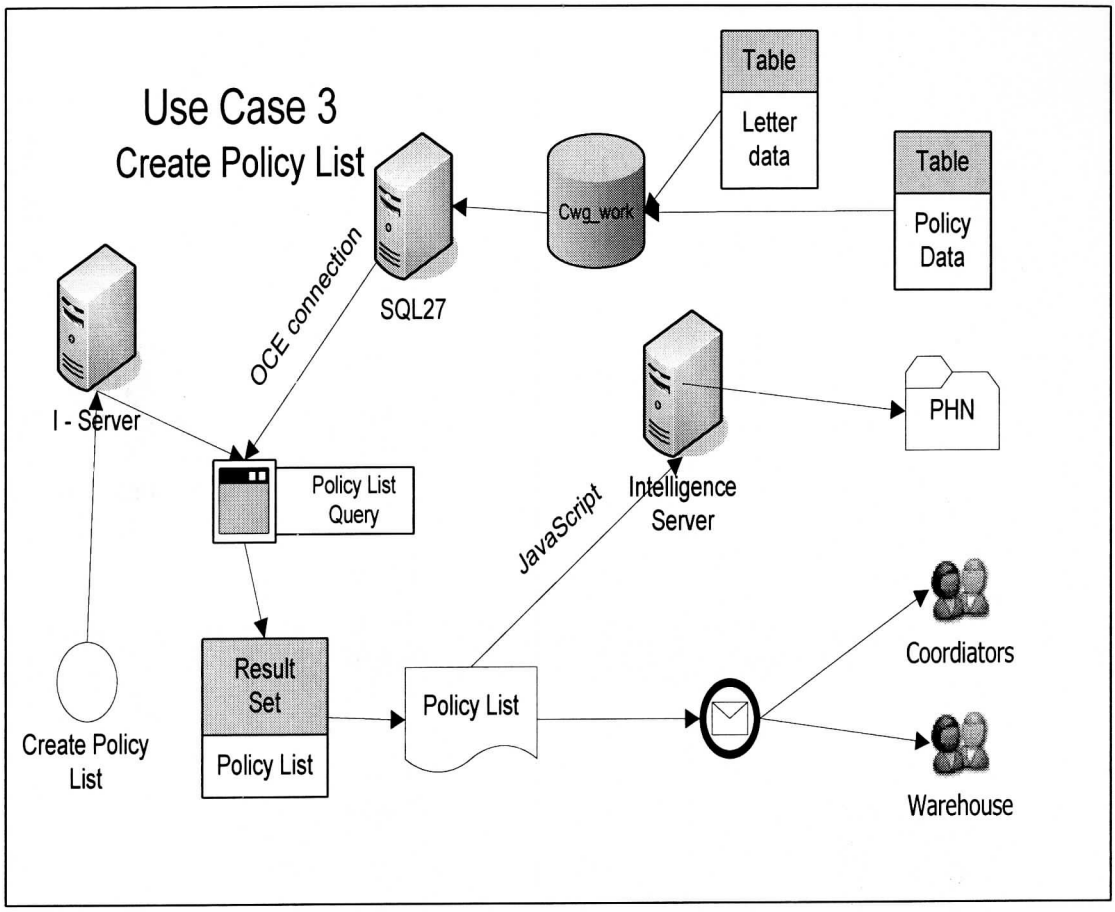

Figure 3.3: Use Case 3: Create the Policy List

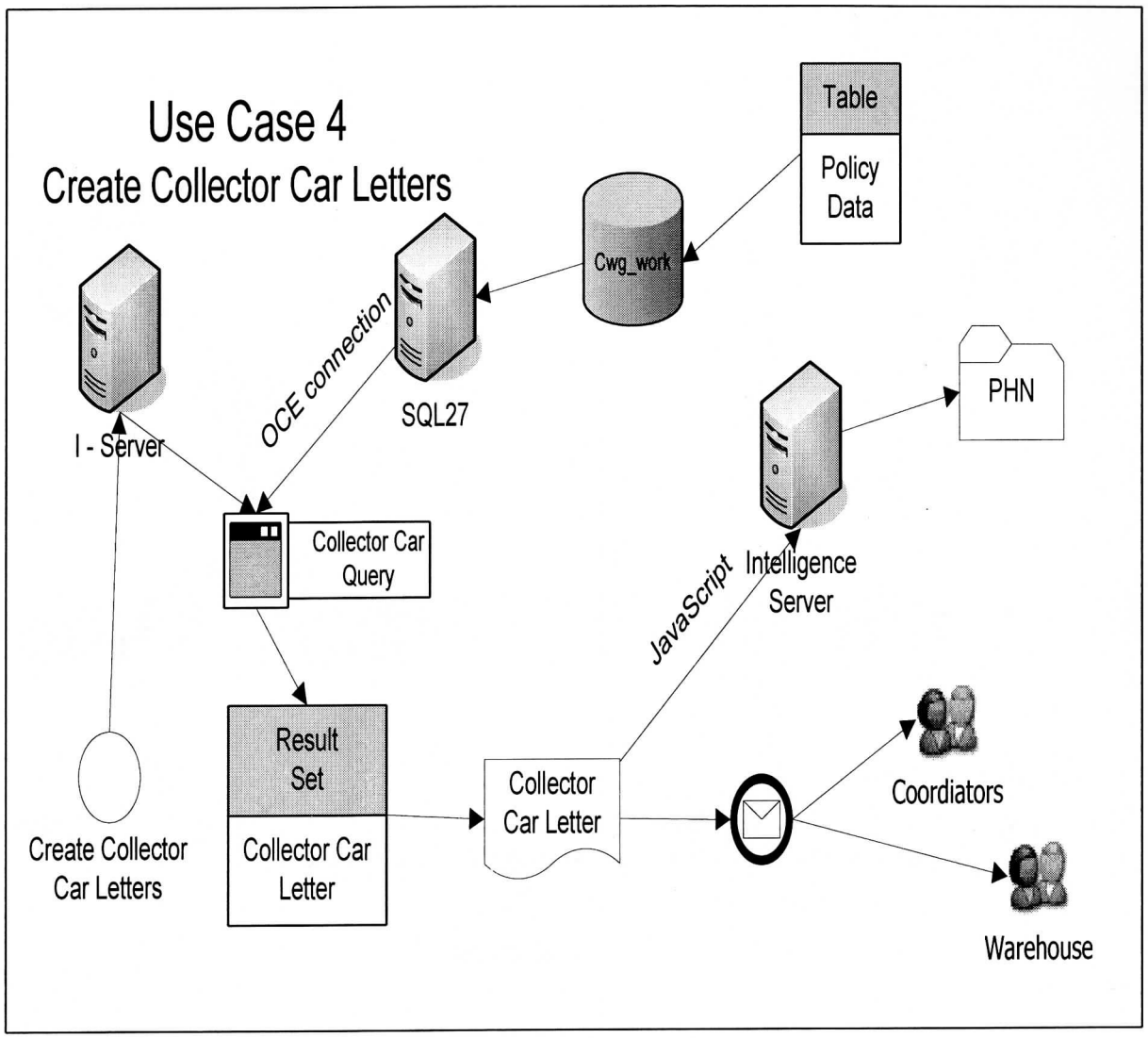

Figure 3.4: Use Case 4 – Create Collector Car Letters

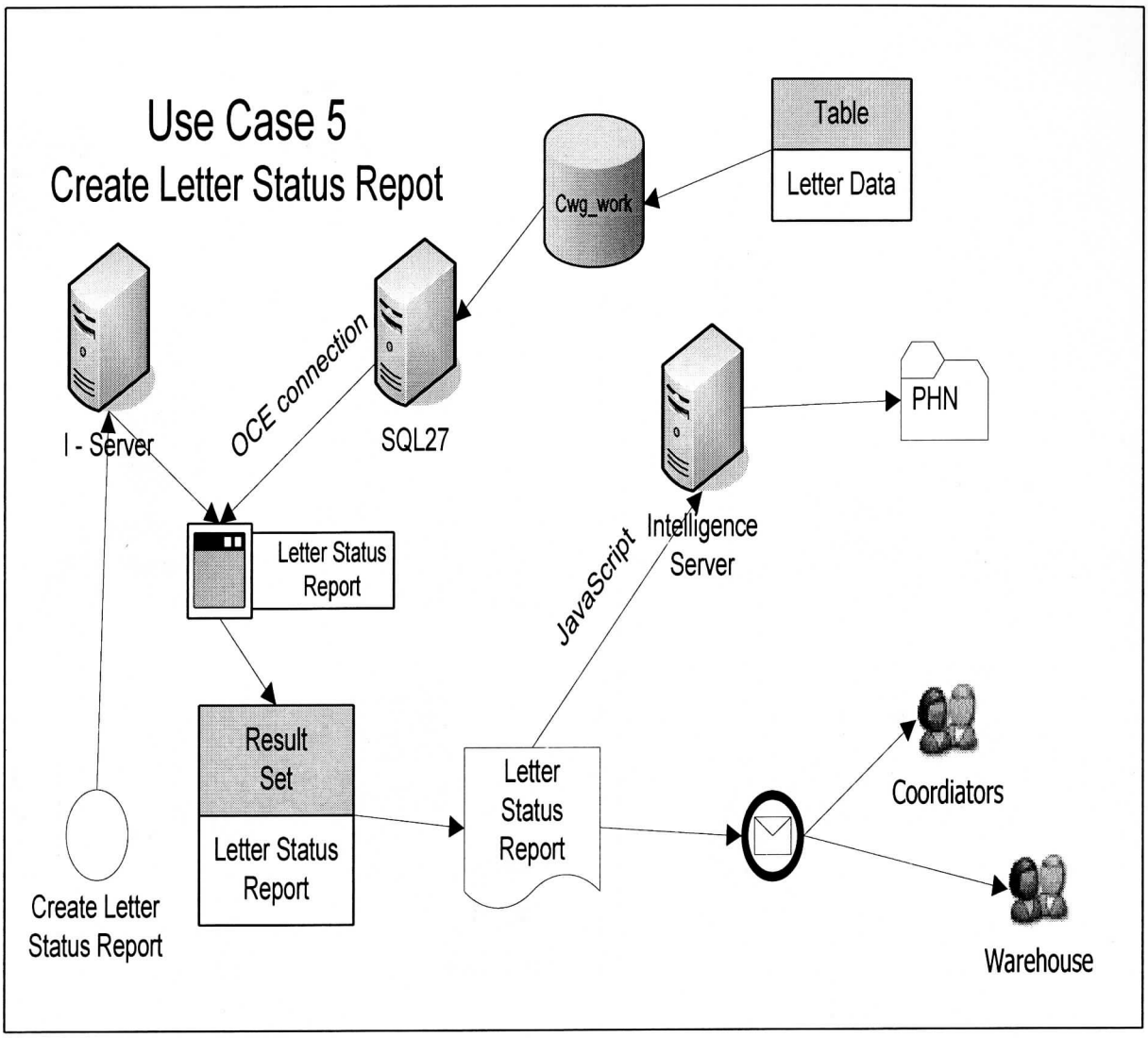

Figure 3.5: Use Case 5 – Create the Letter Status Report

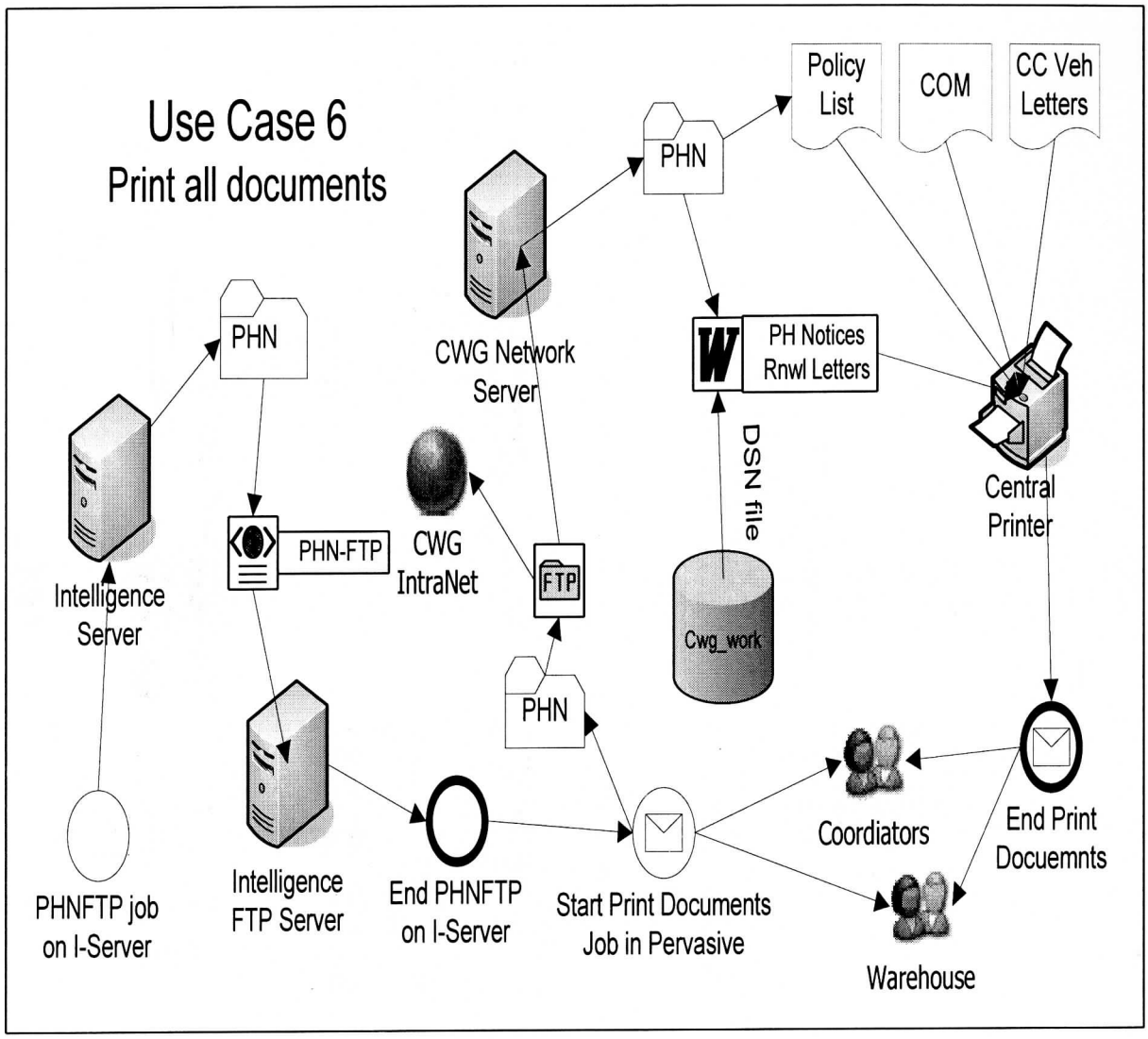

Figure 3.6 – Use Case 6 – Print all the documents

CHAPTER 4 RESULTS AND DISCUSSION

Development Process

The master query was the first component developed for this program. The tool used is the IBM Informix SQL Editor and the query was written against the DSS system on CWG's production server cwgpn02. The query uses as input an Excel spreadsheet that contains specific information about each letter. See Appendix B for a copy of the actual spreadsheet, Table B.2 in Appendix C shows the spreadsheet columns and a description. After the master query completes we have two text files, one contains data about the letters and the other contains data about the policies attached to each letter. Both of these text files are mapped to a table on the CWG Data Warehouse. See Table B.3 in Appendix C for the layout of the letter table. See Table B.4 in Appendix C for the layout of the policy table.

Some of the data needed further formatting so a second query was designed, this time on SQL Server Query Analyzer. We needed to get the underwriter name from the data warehouse, format the rating company and branch office so they were a name and not a number and formatted the date fields to be mm/dd/yyy format.

The SQL Server tables are accessed by the word documents for each letter as a data source. A .dsn file was created to link the word docs to the datamart and SQL27 servers. They are also the source files for the jobs run in Hyperion Intelligence Explorer that create the certificate of mail, policy list, letter status report and the collector car vehicle letters. See Table B.1 in Appendix C for locations of system components.

The jobs created in Hyperion Explorer are scheduled through the Hyperion Performance Suite or what CWG refers to as the I – Server. The ETL job to create the policy and letter data tables is scheduled through Embarcadero. Because we do not run Hyperion jobs through Embarcadero we have to organize the times at which these jobs kick off so the timing is correct. See table 4.1.

Table 4.1: Sequence of scheduling jobs on Embarcadero and the I – Server

Seq#	Job Description	Start Time
1	Daily A+PLUS cycle completes between 4:00 and 6:00 a.m. Tuesday through Saturday	6:00 p.m. Central
2	Embarcadero stars the Pervasive ETL process to create the policy and letter data tables	7:00 a.m. Central
3	The I-Server starts a group of jobs that will all run in parallel, they create the Certificate of Mail, Policy List, Letter Status Report and ten different collector car vehicle Letters	7:15 a.m. Central
4	The I-Server starts another job that cannot start until Seq#3 is completed. An xml file is run to move all the PDF documents created in Seq#3 to the Intelligence FTP server	7:30 a.m. Central
5	Now Embarcadero begins its next job, Seq#4 must finish before this starts. The PDF files on the Intelligence FTP server are moved to the CWG Network, then the letters, Certificate of Mail and Policy List are sent to the printer and the certificate of mail and policy lists are sent to the Intranet.	7:45 a.m. Central

The files created in Hyperion Explorer have JavaScript built into them to export a PDF file to a common location. This location acts as the source for the job that uses the xml file to get the PDF's to the FTP server.

Verification of Data

At this time the process is programmed with a sampling of the policyholder notices and renewal letters. The renewal letters programmed are UW0013CC, UW0014C2, UW0015CC, UW0016C2, UW0017CC, UW0018C2, UW0026CC and UW0027C2. The policyholder notices programmed are PN6010, PN6113, PN6115, PN6118, PN6119 and PN6120. The first step in testing will be to compare the results we are currently getting on the production system with what we are getting on test.

We have an ELT process set up on the datamart server that queries against the same database the production system does (cwgpn02). We will reset the process date on the test

query to 11/17/2006 and reset the last run date to 11/16/2006. Friday's production run used previous day (11/16) because the DSS is updated daily, not in real time. See table 4.1 for a summary of testing. A more detailed summary of testing can be found on the Excel spreadsheet in Appendix B under 'Testing Results.'

Table 4.2: Summary of testing results from 11/16 & 11/17

Processing Date	Previous Run Date	Form Name	Existing System	Proposed System	Comments
11/17/2006	11/16/2006	PN6010	3	3	All policies matched
11/17/2006	11/16/2006	PN6113	4	4	All policies matched
11/17/2006	11/16/2006	PN6115	0	0	
11/17/2006	11/16/2006	PN6118	39	40	It was determined that the extra policy should have hit on the existing system also.
11/17/2006	11/16/2006	PN6119	10	10	All policies matched
11/17/2006	11/16/2006	PN6120	1	1	Policy matched
11/17/2006	11/16/2006	UW0011CC	0	0	
11/17/2006	11/16/2006	UW0012C2	0	0	
11/17/2006	11/16/2006	UW0013CC	2	2	Policies matched
11/17/2006	11/16/2006	UW0014C2	0	0	
11/17/2006	11/16/2006	UW0015CC	0	0	
11/17/2006	11/16/2006	UW0016C2	0	0	
11/17/2006	11/16/2006	UW0017CC	0	0	
11/17/2006	11/16/2006	UW0018C2	0	0	
11/17/2006	11/16/2006	UW0026CC	1	1	Policy Matched
11/17/2006	11/16/2006	UW0027C2	0	0	
11/16/2006	11/15/2006	PN6010	0	0	
11/16/2006	11/15/2006	PN6113	2	2	Policies match up
11/16/2006	11/15/2006	PN6115	0	0	
11/16/2006	11/15/2006	PN6118	34	34	Policies match up

Processing Date	Previous Run Date	Form Name	Existing System	SUD	Comments
11/16/2006	11/15/2006	PN6119	10	10	Policies match up
11/16/2006	11/15/2006	PN6120	1	1	Policy matches up
11/16/2006	11/15/2006	UW0011CC	0	0	
11/16/2006	11/15/2006	UW0012C2	0	0	
11/16/2006	11/15/2006	UW0013CC	3	3	Policies matched
11/16/2006	11/15/2006	UW0014C2	0	0	
11/16/2006	11/15/2006	UW0015CC	1	1	Policy matched
11/16/2006	11/15/2006	UW0016C2	0	0	
11/16/2006	11/15/2006	UW0017CC	1	1	Policy matched
11/16/2006	11/15/2006	UW0018C2	0	0	
11/16/2006	11/15/2006	UW0026CC	2	2	Policies matched
11/16/2006	11/15/2006	UW0027C2	1	1	Policy matched

All issues found in testing the data from 11/16/2006 and 11/17/2006 were addressed and corrected. The only issue outstanding is with the form PN6118 where an extra policy printed on 11/17, the testing checked out on 11/16. After analysis of the policy in question it was determined that the policy did satisfy the conditions of the form so we will approach this as a problem with the existing system.

Now that we have proved that a sampling of the forms will attach the same policies as the existing system we are safe to go ahead and program the attachment for the rest of the forms. This task will be performed by the BSA staff after training is completed for them. The general concept of using DSS as our attachment database, the BSA will do individual testing on each letter to make sure the query is programmed properly.

Extract/Merge Process

The initial merge to production

The environment support team at CWG has as its two main roles to maintain the schedules jobs in Embarcadero and to perform the extract and merge of systems from one environment to another. When the project has completed its testing on the datamart server the ESG team will move the contents of the project's process folder to the SQL27 server. The folder is named 'CWG_Daily_PHN_Process and its contents include:

- copyXMLfiles.bat – copies xml files used by the Pervasive ETL process
- CWG_Daily_PHN_Process.INI – this file contains configuration settings for the ETL process
- CWG_Daily_PHN_Process.ip.xml – xml file used for the ETL process
- CWG_Daily_PHN_ProcessFileList-IN.txt – Text file that contains three xml file names, one is the main xml file for this process, the next is for the macros used within the process and the third is for the maps used in the process.
- FileList-IN.txt – This text file contains the name of the process
- StartCWG_Daily_PHN_Process-DJCosmos.bat – This is the .bat file that starts the process
- CreateProductionFolderStatements.sql – This SQL file creates the production folder when it is time to move the process to the SQL27 server.
- CWG_Daily_PHN_Process_Param.sql – This SQL file sets off the parameters for the process these parameters are stored on a table called 'ParameterTable'.
- CWG_Daily_Process_Resourses.sql – this SQL file sets all the resources for this process. These resources are set in a table on the data warehouse called 'Process_Resources_Table'.
- Tcs_phn_main.sql – SQL File for the Informix Query against DSS. This is what we been referring to as the 'Master Query'.
- Tcs_phn_print_format_query.sql – SQL file for the print formatting query run on SQL Server.
- Letter.xls – MS Excel file that serves as the template for entering a new form.

- Letter.txt – Pipe delimited text file. The letter.xls file is mapped to this text file so it can be loaded into the master query.
- Letter_status.txt – Pipe delimited text file that is outputted by the master query, it is then mapped to the table phn_letter_data on the cwg_work database and eventually becomes the letter status report
- Policy_input.txt – Pipe delimited text file outputted by the master query, it is mapped to the table phn_policy_data on the cwg_work database and serves as the main data source for the letters, certificate of mail and policy list.
- Cntr_class_codes.csv – This is a delimited file that was created to be used in the pre-condition for one of the letters. It contains all general liability contractors' class code.

See Appendix B for two documents. The first is the procedures we follow when moving a process to the NOC, which is where our production processes are and the second is the procedure to add a scheduled job to Embarcadero.

Also after the testing is complete on datamart the BSA acting as policyholder notice coordinator will move the Word Documents and their data sources from the 'PHN' folder on the network to a folder called 'PHN_prod'.

When a new letter is added to the system

Once the system is in place on production we will need to make updates when new letters are added. When a BSA has made the necessary updates to the master query and the Excel spreadsheet a request will be made to the ESG team to move the query and spreadsheet to the process folder on datamart.

After testing is completed on datamart the ESG team will move the query and spreadsheet to the SQL27 server. Once ESG has notified the policyholder notice coordinator that the merge to SQL27 is completed the new Word Document and its data source will be copied from the 'PHN' folder to 'PHN_prod'. The new letter needs to be on SQL27 and the 'PHN_prod' folder the day before it is to begin printing. Ideally we'd like it to be there a week before in case any issues are encountered.

CHAPTER 5 CONCLUSIONS

Did we meet the project objectives?

At this time the project is not fully operational, the pieces have been put into place, we still have to last issue to address. The jobs created in Hyperion Intelligence Explorer are not sending the PDF files to the appropriate location on the Intelligence server. So we do not yet have the scheduled jobs tested on the I – Server. We expect to have this issue resolved before December 1. Training is scheduled with the BSA group for the second week of December, at that time they will be able to modify and maintain the system. So we are on track to fully implement the system on the datamart server before the end of the year and to implement on SQL27 in January.

I will address the objective individually. The first was to eliminate the use of the renewal specs. This has been accomplished. The new system is able to get the same data out of DSS that the renewal spec retrieved from the A+PLUS PREPMSTR files. After the system is implemented on the SQL27 server we will delete the renewal 'L' specs to free up COBOL resources for the other parts of renewal maintenance.

The second objective was to make the testing process more efficient by removing the responsibility of creating the RENPRT file from the BSA group and getting a better set of data to test. We have taken away the need to run the A+PLUS utility programs, the BSA will be able to utilize both the DSS test or production environment to test their query changes. That gives them the flexibility to enter their own test policies and use the set of test policies that already exist. After the query has tested successfully they will be able to test the word documents and the certificate of mail after the daily job executes. Currently it takes a BSA a total of five to ten hours to properly test a policyholder notice. We now expect to get that down to two hours or less for testing. Detail testing documentation can be found in the users' manual (Appendix A).

Another object was to replace MS Access; we have accomplished that with Hyperion Intelligence Explorer. The Access database currently being used will be disabled and removed from the CWG network after the system is operational on SQL27.

Next we wanted to shorten the window of time involved in getting a letter into production. Because we have eliminated the need for using the renewal specifications we are not bound to the A+PLUS custom release schedule. The time window now is a minimum of two weeks and a maximum of up to six weeks. With the new system we should be able to implement a change over night with proper documentation to the environment support team.

The policy list generated for renewal letters accomplishes the objective of putting a mechanism in place to verify the renewal letters. Now the mail room will have a form similar to the certificate to use, the difference is the policy list will be for in-house use only.

Training the BSA staff was another objective. The new policyholder notice coordinator began training on November 21, the entire BSA group will have a formal four hour training session during the second week in December.

The next objective was to eliminate the need to manually correct problems with specific letters outside of the system. Currently we have procedures set up to make corrections for PN6010, PN6113, UW0020, UW0021, UW0022. UW0037 and UW0040. All the issues causing problems with these letters have been corrected with the new system and they will all be automated.

The last objective was to automate the printing of the collector car letters that print vehicle information. These letters are UW0011CC, UW0012C2, UW0013CC, UW0014C2, UW0015CC, UW0016C2, UW0017CC, UW0018C2, UW0026CC and UW0027C2. Currently the word processing staff has to manually create one of these letters if they have more than one vehicle printing on the form. This issue has been corrected by using Hyperion Intelligence Explorer to create these letters.

Deliverables

The deliverables were addressed in the project description, which was turned in on October 9. The first was the set of policyholder notices and renewal letters. These are being created in MS Word as they were before, the difference being the data source is now a SQL server database table. Some of the letters are getting created in Hyperion Explorer.

The next deliverable is the certificate of mail. That is being created in Hyperion Explorer and will not require any more updating by the BSA staff. So we have delivered this item and at the same time taken away a development task.

A master list of all the policyholder notices and renewal letters was another deliverable. This was accomplished with the creating of the letter status report (See a documents link in Appendix B).

Training for the BSA staff was identified as another deliverable. This was begun for the two policyholder notice coordinators on Nov. 21; training for the entire staff will be done in December.

There was a desire to create a template from which the BSA could create a new form. To address this letter spreadsheet was created. You can see this document in Appendix B. On this spreadsheet the BSA will enter all required information to start attaching a letter. The master query pulls this data in to help set the criteria for each letter.

The final deliverable was the development manual. This is found in Appendix A. This user's manual gives directions to the BSA staff how to set up a new policyholder notice and give instructions on testing and work flow.

Things learned

This was a very education project for me. Most of the tools need to complete this project I have never used before. However the first thing I learned was how to be created with SQL code. Programming the master query was a major challenge and I had never written a query of such magnitude before. For the first time I extensively used sub-queries and updates.

I had not received the Pervasive software until September so the ETL job I put together for this project was my first. I had to become familiar with the procedures CWG already has in place for scheduled jobs and adjust this project to it. The ETL software gave me a chance to work with maps and allow me to be able to work on projects where I pulled data from different servers into a common area.

This project gave me my first exposure to working in SQL Server and Query Analyzer. I also had to educate myself on all the different servers used by CWG and BTS and understand where the data has to reside for development, testing and production.

Another new tool I had to learn for this project was Hyperion Intelligence Explorer, and with that the I – Server and its scheduling of jobs. The Hyperion tools are very useful in setting up reports but at the same time can be frustrating. I got into the details and learned

some things about this tool that helped greatly. For example, I learned how to use the "\m" character to force a carriage return, this have a huge difference in how my reports appeared. I discovered the 'cume' function to be very useful creating the sequencing number on certificate of mail reports and policy lists. Intelligence forced me to understand the file structures used by its server at the NOC and all the hoops you have to make the data jump through to get it where you need it to be. Finally, in regards to Intelligence, the problems we had getting the PDF files moved form Intelligence to the CWG network have prompted me to look for other solutions for the certificate of mail in the future. The first thing I will look at is a .net application.

In general I can now get to so much more data than before I started this project. There are serves and databases I did not even know existed before. Something else I used for the first time was a .dsn file to connect between servers. This was necessary to connect the MS Word documents to the data sources on the datamart and SQL27 servers. I also had my first experience with using an xml file.

Another issue that came up with this project is the access different employees have to servers. All our scheduled jobs run on the datamart server for development and SQL27 for production. If the BSA group is to maintain the system they have to do so without having access to datamart. No one except for the ESG team and make changes to SQL27. That led to setting up the folder on the CWG network for all the word documents and the use of the .dsn file to connect to the data in the CWG warehouse. It also meant we had to set up a procedure for having a working copy of the master query on the CWG network for the BSA's to modify, then have the ESG team merge any changes to datamart. We will have to give datamart security to the two policyholder notice coordinators because they will have to get into the master query and adjust dates for testing purposes, otherwise we would not be able to test our new letters until they day they actually become effective.

The most important thing I had to do on this project was to be creative in looking for solutions.

Future enhancements

After the project is complete I will be able to get some training on the .net applications. I will look into replacing the jobs we do in Hyperion with the reporting

facilities in .net. Another reason I'd like to research .net is because of the limitation we have of not scheduling Hyperion jobs in Embarcadero.

The daily job to create the data could use some enhancements. We will need to build in more notifications. Currently all we do is send a start and stop email. I would like to add an email notification if the master query fails. I would also like to put in a check to make sure the A+PLUS cycle has completed before we start the daily process. Some data checking could also be done on the maps. For example, if some has the letter spreadsheet open, the process with shut down. We could kick out a notification if that happens. Sometimes the mapping pulls in bad data such as nulls and bad values that fail an update query, we need to look into these issue more closely see what improvements can be made to avoid system errors.

Our company will be moving from the Pervasive ETL tool to Data Stage in 2007, at that time the daily jobs in Pervasive will have to be converted to Data Stage. We are also in the process of enhancing the company data warehouse to it has more detailed data. The motivation for this is so we can get all our scheduled jobs running against the data warehouse instead of the corporate warehouses like DSS or its equivalent on the loss side, the LDR (loss data repository). We will add all the fields used in the master query to the data warehouse so we can eventually run this against the warehouse. We expect that to happen sometime after June of 2007.

REFERENCES

IBM Informix Dynamic Server v10.0 Information Center. Retrieved from http://publib.boulder.ibm.com/infocenter/idshelp/v10/index.jsp.

Ostroff, M. How'd you do that? 6th Edition. Hyperion Intelligence Tips and Tricks Cookbook. Copyright 2005, Hyperion Solutions, Santa Anna, CA. Retrieved from http://dev.hyperion.com/downloads/code_library/intelligence/Tips_Cookbook_2005.pdf.

Ostroff, M. How did you do That? Hyperion Intelligence Tips & Tricks Part 2. Copyright 2004. Hyperion Solutions. Retrieved at http://dev.hyperion.com/download/license_agreement.cfm?file=code_library/intelligence/intelligencetips_cookbook_2004.zip.

Ostroff, M. Hyperion Intelligence Tips & Tricks Part 1 (a. k a. "How'd you do That?). Copyright 2005, Hyperion Solutions. Retrieved at http://dev.hyperion.com/download/license_agreement.cfm?file=code_library/intelligence/Tips_for_2005.zip.

The Code, AdamFranz.com. Retrieved at http://www.adamfranz.com/code.html.

APPENDICES

APPENDIX A: SYSTEM DOCUMENTS

Policyholder notice renewal letter spreadsheet

On the next page we have printed a copy of the policyholder notice and renewal letter spreadsheet. It contains information needed to determine attachment of the letters and is used as input for the master query.

To keep the page numbering separate from the page numbering for this report the letters "LS" preceed the page number in the bottom right-hand corner of the report.

Policyholder Notice and Renewal Letter Template

Form Num	Seq	Edition	Description	Developer	Type	Pages	Eff Dt	Exp Dt	Term	Days Prior	Copies	Rtn Env?	Duplex?	Imaged?	Cert Of Mail?	Mail To	I Net Copy
PN6010	1	07-05	DXS and Umbrella Notice	Ryan Roberts	PN	1	5/15/2006	5/15/2007	1 year	65	1	N	N	N	Y	I	Y
PN6051	1	07-05	Policyholder Advisory Notice - Contractors Equipment IM7000 04 04	Susan Weidemann	PN	2	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6052	1	07-05	Policyholder Advisory Notice - Contractors Equipment IM7001 06 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6054	1	07-05	Policyholder Advisory Notice Contractors Equipment IM7003 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6055	1	07-05	Policyholder Advisory Notice Installation Floater IM7100 06 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6056	1	07-05	Policyholder Advisory Notice Installation Floater IM7101 0604	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6057	1	07-05	Policyholder Advisory Notice Builders' Risk Coverage Form IM7050 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6058	1	07-05	Policyholder Advisory Notice Builders' Risk Coverage Form IM 7051 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6059	1	07-05	Policyholder Advisory Notice Builders' Risk Coverage form IM7052 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6060	1	07-05	Policyholder Advisory Notice Builders' Risk Coverage Form IM7053 04 04	Susan Weidemann	PN	6	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6061	1	07-05	Policyholder Advisory Notice Scheduled Property Floater IM7500 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6062	1	07-05	Policyholder Advisory Notice Miscellaneous Bailee - Processor Floater IM7501 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6064	1	07-05	Policyholder Advisory Notice Contractors Equipment IM7001 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6066	1	07-05	Policyholder Advisory Notice Contractors Equipment IM 7003 04 04	Susan Weidemann	PN	2	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6067	1	07-05	Policyholder Advisory Notice Installation Floater IM7100 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6073	1	07-05	Policyholder Advisory Notice Schedule Property Floater IM7500 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6075	1	07-05	Policyholder Advisory Notice Bailee Customers Floater - Dry Cleaner IM7550 06 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6081	1	07-05	Policyholder Advisory Notice Fine Arts Floater IM7400 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6085	1	07-05	Policyholder Advisory Notice Radio & Television Towers & Eq IM7600 04 04	Susan Weidemann	PN	6	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6087	1	07-05	Policyholder Advisory Notice Riggers Liability IM7150 06 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y

Policyholder Notice and Renewal Letter Template

PN6091	1	07-05	Policyholder Advisory Notice Transportation Coverage IM7250 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6092	1	07-05	Policyholder Advisory Notice owner's Cargo Coverage IM7251 04 04	Susan Weidemann	PN	4	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6098	1	08-05	Advisory Notice to Policyholders - Motor Truck Cargo Legal Liability Coverage Reporting Form IM5046 07 05	Susan Weidemann	PN	6	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6099	1	08-05	Advisory Notice to Policyholder - Motor Truck Cargo Legal Liability Coverage IM5747 07 05	Susan Weidemann	PN	6	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6100	1	08-05	Advisory Notice to Policyholder - Motor Truck Cargo Legal Liability Coverage IM5747 07 05	Susan Weidemann	PN	6	12/15/2005	12/15/2006	1 year	90	1	1	Y	N	Y	I	Y
PN6104	1	02-05	Policyholder Notice - Church Guardian Program advisory notice		PN	2	8/1/2005	8/1/2006	1 year	90	1	N	Y	N	Y	I	Y
PN6105	1	01-06	Policyholder Notice - Advisory Notice for Personal Auto Policy	Jason Prekker	PN	4	1/1/2006	1/1/2007	1 year	60	1	N	Y	N	Y	I	Y
PN6108	1	09-05	Policyholder Notice - General Liability form revision for Minnesota		PN	1	10/1/2005	10/1/2006	1 year	90	1	N	N	N	Y	I	Y
PN6113	1	05-06	Policyholder Notice - Illinois Defense Costs	Anne Pittard	PN	1	5/1/2006	5/1/2007	1 year	90	1	N	N	N	Y	I	Y
PN6114	1	11-05	Policyholder Notice - Advisory Notice for Umbrella Policy	Ryan Roberts	PN	1	12/15/2005	12/15/2006	1 year	90	1	N	N	N	Y	I	Y
PN6115	1	01-06	Policyholder Notice - Umbrella Liability	Jason Prekker	PN	1	3/1/2006	3/1/2007	1 year	65	1	N	N	N	Y	I	Y
PN6116	1	11-05	Policyholder Notice - Umbrella Email, Fax, Phone Calls Exclusion	Jeanne King	PN	1	2/15/2006	12/16/2006	1 year	90	1	N	N	N	Y	I	Y
PN6118	1	06-06	Policyholder Notice - Amendment of Insured Contract Definition	Deb Theirolf	PN	1	9/1/2006	9/1/2007	1 year	90	1	N	N	N	Y	I	Y
PN6119	1	10-06	Policyholder Notice Commercial Auto multi-state form revision - SD, WY, OR	Ryan Roberts	PN	6	10/1/2006	10/1/2007	1 year	90	1	N	Y	N	Y	I	Y
PN6119	2	10-06	Policyholder Notice Commercial Auto multi-state form revision - MO	Anne Pittard	PN	6	1/1/2007	1/1/2008	1 year	90	1	N	Y	N	Y	I	Y
PN6119	3	10-06	Policyholder Notice Commercial Auto multi-state form revision - IA	Jeanne King	PN	6	2/1/2007	2/1/2008	1 year	90	1	N	Y	N	Y	I	Y
PN6119	4	10-06	Policyholder Notice Commercial Auto multi-state form revision - IA	Kristy Jensen	PN	6	3/1/2007	3/1/2008	1 year	90	1	N	Y	N	Y	I	Y
PN6120	1	06-06	Policyholder Notice - Amendment of Insured Contract Definition	Jeanne King	PN	1	9/1/2006	9/1/2007	1 year	90	1	N	N	N	Y	I	Y
PN6121	1	06-06	Policyholder Notice - Equipment Breakdown	Deb Theirolf	PN	2	10/1/2006	10/1/2007	1 year	90	1	N	N	N	Y	I	Y
PN6128	1	05-06	Policyholder Notice - Seclusion Building Inspection	Gloria Gary	PN	1	10/15/2006	10/15/2007	1 year	90	1	N	N	N	Y	I	Y
PN6130	1	06-06	Policyholder Notice - Exclusion Injury or Damage due to earth movement	Susan Weidemann	PN	1	12/1/2006	12/1/2007	1 year	90	1	N	N	N	Y	I	Y
PN6131	1	07-06	Policyholder Notice - Punitive Damage Exclusion	Ryan Roberts	PN	1	12/1/2006	12/1/2007	1 year	90	1	N	N	N	Y	I	Y

Policyholder Notice and Renewal Letter Template

PN6133	1	07-06	Policyholder Notice - Water Districts Union to CW Conversion	Deb Theirolf	PN	1	2/1/2007	2/1/2008	1 year	90	1	N	N	N	Y	I	Y
PN6134	1	07-06	Policyholder Notice - Water Districts Union to CW Conversion	Deb Theirolf	PN	1	2/1/2007	2/1/2008	1 year	90	1	N	N	N	Y	I	Y
PN6141	1	07-06	Policyholder Notice - GL Forms Revision	Deb Theirolf	PN	1	2/1/2007	2/1/2008	1 year	90	1	N	N	N	Y	I	Y
UW0011CC	1	06/2003	Sr. Driver Discount - CC	Jason Prekker	RL	1	1/1/1901	12/31/2099	on-going	120	2	N	N	Y	N	A	N
UW0012C2	1	06/2003	Sr Driver Discount - CC2	Jason Prekker	RL	1	1/1/1901	12/31/2099	on-going	90	2	Y	N	Y	N	I	N
UW0013CC	1	06/2003	Limited Use Odometer - CC	Jason Prekker	RL	1	10/1/2003	12/31/2099	on-going	120	2	N	N	Y	N	A	N
UW0014C2	1	06/2003	Limited Use Odometer - CC2	Jason Prekker	RL	1	10/1/2003	12/31/2099	on-going	90	2	Y	N	Y	N	I	N
UW0015CC	1	06/2003	Odometer Reading for Category 9 - CC	Jason Prekker	RL	1	10/1/2003	12/31/2099	on-going	120	2	N	N	Y	N	A	N
UW0016C2	1	06/2003	Odometer Reading for Category 9 - CC2	Jason Prekker	RL	1	10/1/2003	12/31/2099	on-going	90	2	Y	N	Y	N	I	N
UW0017CC	1	06/2003	Odometer Special Use - CC	Jason Prekker	RL	1	10/1/2003	12/31/2099	on-going	120	2	N	N	Y	N	A	N
UW0018C2	1	06/2003	Odometer Special use - CC2	Jason Prekker	RL	1	10/1/2003	12/31/2099	on-going	90	2	Y	N	Y	N	I	N
UW0020	1	07/2003	Farmowner Renewal Questionnaire for Non-Blanket policies	Barb Mossholder	RL	1	10/1/2003	12/31/2099	on-going	90	2	N	Y	Y	N	A	N
UW0021	1	07/2003	Farmowner Renewal Questionnaire for Blanket policies	Barb Mossholder	RL	1	10/1/2003	12/31/2099	on-going	90	2	N	Y	Y	N	I	N
UW0022	1	07/2003	Farmowner - custom Farming or Business Endorsement	Barb Mossholder	RL	1	10/1/2003	12/31/2099	on-going	90	2	N	N	Y	N	A	N
UW0026CC	1	10/2003	Restoration Form - CC	Jason Prekker	RL	1	1/1/1901	12/31/2099	on-going	120	2	N	N	Y	N	A	N
UW0027C2	1	10/2003	Restoration Form - CC2	Jason Prekker	RL	1	1/1/1901	12/31/2099	on-going	90	2	N	N	Y	N	I	N
UW0030IA	1	03-04	Underwriting Letter - Commercial Auto Garage program	Jason Prekker	RL	1	12/1/2005	12/1/2006	1 year	120	2	N	Y	Y	N	A	N
UW0030IL	1	03-04	Underwriting Letter - Commercial Auto Garage program	Jason Prekker	RL	1	10/1/2005	10/1/2006	1 year	120	2	N	Y	Y	N	A	N
UW0030KS	1	03-04	Underwriting Letter - Commercial Auto Garage program	Jason Prekker	RL	1	12/1/2005	12/1/2006	1 year	120	2	N	N	Y	N	A	N
UW0033	1	06-05	Underwriting Letter - Renewal Application for Liquor Liability	Barb Mossholder	RL	1	1/1/1901	12/31/2099	on-going	120	2	Y	Y	Y	N	A	N
UW0033A	1	06-05	Underwriting Letter - Renewal Application for Liquor Liability	Barb Mossholder	RL	1	1/1/1901	12/31/2099	on-going	120	2	Y	Y	Y	N	A	N
UW0037	1	10-05	Underwriting Letter - Renewal Application for CWP Liability	Ryan Roberts	RL	1	5/1/2006	12/31/2099	on-going	120	2	Y	N	Y	N	A	N
UW0040	1	02-06	UW Letter - MN coinsurance changes	Ryan Roberts	RL	1	1/1/2006	12/31/2099	on-going	65	2	Y	N	Y	N	A	N
UW0040	2	02-06	UW Letter - MN coinsurance changes	Todd Koehler	RL	1	1/1/2006	12/31/2099	on-going	60	2	Y	N	Y	N	A	N
PN6137	1	07-06	Policyholder Notice - IMD renewals from UIC to CWIC	Susan Weidemann	PN	1	3/1/2007	3/1/2008	1 year	90	1	N	N	N	Y	I	Y
PN6138	1	07-06	Policyholder Notice - IMD renewals from UIC to CWIC	Susan Weidemann	PN	1	3/1/2007	3/1/2008	1 year	90	1	N	N	N	Y	I	Y

Parallel testing spreadsheet

The spreadsheet printed on the next set of pages illustrates parallel testing that was done on the new system in comparison with the existing system. Sixteen forms were tested, you will see two columns for each form, the first column lists the policy numbers generated on the current production system and the second lists policy numbers generated by the system under discussion.

The testing was done over 11/16/2006 and 11/17/2006. Three of the forms were also tested on 11/20/2006. The addresses some questions that came up in the testing specific to some forms and how these questions are being addressed.

Parallel Testing

Testing Notes

11/19/2006 PN6118	I discovered that we cannot check for GL class codes in the 90000 range The table PPGL0191 does not have all the class codes in this range We will have to re-work the pre-condition.
11/20/2006 PN6118	I created a text file with all the values on logical table PPGL0191 and ran the process again. After adding the table the results match with one exception. Policy 2429942 appeared on the new system but did not appear on the existing system. When I looked at this policy on the A+PLUS system nothing indicated that it should not attaching this form. The class code 98305 is on the table PPGL0191 and the policy is in-force and has is set to computer renew. I'm moving on and marking this an an issue with the existing system.
11/20/2006 PN6118	With is issue of policy 2429942 noted as a problem with the exising system, I'm going to classify the testing for 11/17/2006 data complete and approved.

Test results.xls

Parallel Testing

	PN6010		PN6113		PN6115	
Date	**Production System**	**Proposed System**	**Production System**	**Proposed System**	**Production System**	**Proposed System**
11/17/2006	2558177	2558177	2409926	2409926		
	2559419	2559419	2412357	2412357		
	2559506	2559506	2413046	2413046		
			2506998	2506998		
11/20/2006	2558183	2558183	2392809	2392809	2402950	2402950
			2412944	2412944		
			2412984	2412984		
			2412988	2412988		
			2413461	2413461		
			2413462	2413462		
			2508882	2508882		
			2561111	2561111		
			2566127	2566127		
			2610004	2610004		
			2612180	2612180		
			2615902	2615902		
11/16/2006			2560345	2560345		
			2612188	2612188		

Parallel Testing

	PN6118		PN6119		PN6120	
Date	Production System	System Under Dicsussion	Production System	System Under Dicsussion	Production System	System Under Dicsussion
11/17/2006	2406865	2406865	2406865	2406865	2564016	2564016
11/17/2006	2409926	2409926	2412319	2412319		
11/17/2006	2412311	2412311	2412324	2412324		
11/17/2006	2412336	2412336	2412363	2412363		
11/17/2006	2412338	2412338	2412829	2412829		
11/17/2006	2412339	2412339	2412837	2412837		
11/17/2006	2412344	2412344	2443673	2443673		
11/17/2006	2412345	2412345	2560172	2560172		
11/17/2006	2412352	2412352	2560286	2560286		
11/17/2006	2412354	2412354	2561788	2561788		
11/17/2006	2412356	2412356				
11/17/2006	2412357	2412357				
11/17/2006	2412372	2412372				
11/17/2006	2412819	2412819				
11/17/2006	2412836	2412836				
11/17/2006	2412843	2412843				
11/17/2006	2416053	2416053				
11/17/2006	2417020	2417020				
11/17/2006	2426785	2426785				
11/17/2006	2439686	2429942				
11/17/2006	2500772	2439686				
11/17/2006	2507839	2500772				
11/17/2006	2554848	2507839				
11/17/2006	2557427	2554848				
11/17/2006	2560286	2557427				
11/17/2006	2561115	2560286				
11/17/2006	2562630	2561115				
11/17/2006	2562728	2562630				
11/17/2006	2562984	2562728				
11/17/2006	2563404	2562984				
11/17/2006	2598348	2563404				
11/17/2006	2614435	2598348				
11/17/2006	2615037	2614435				
11/17/2006	2615052	2615037				
11/17/2006	2615507	2615052				
11/17/2006	2615628	2615507				
11/17/2006	2616421	2615628				
11/17/2006	2616424	2616421				
11/17/2006	2616821	2616424				
		2616821				
11/16/2006	2408495	2408495	2412247	2412247	2613411	2613411
11/16/2006	2412263	2412263	2412279	2412279		
11/16/2006	2412274	2412274	2412282	2412282		
11/16/2006	2412279	2412279	2412299	2412299		
11/16/2006	2412282	2412282	2412304	2412304		
11/16/2006	2412296	2412296	2443930	2443930		
11/16/2006	2412304	2412304	2558277	2558277		

Parallel Testing

	PN6118		PN6119		PN6120	
Date	Production System	System Under Dicsussion	Production System	System Under Dicsussion	Production System	System Under Dicsussion
11/16/2006	2412807	2412807	2608495	2608495		
11/16/2006	2425854	2425854	2610615	2610615		
11/16/2006	2435740	2435740	2612666	2612666		
11/16/2006	2437366	2437366				
11/16/2006	2438271	2438271				
11/16/2006	2439813	2439813				
11/16/2006	2493279	2493279				
11/16/2006	2503565	2503565				
11/16/2006	2506302	2506302				
11/16/2006	2507296	2507296				
11/16/2006	2507502	2507502				
11/16/2006	2508220	2508220				
11/16/2006	2510650	2510650				
11/16/2006	2511757	2511757				
11/16/2006	2558277	2558277				
11/16/2006	2560774	2560774				
11/16/2006	2607706	2607706				
11/16/2006	2611888	2611888				
11/16/2006	2611934	2611934				
11/16/2006	2612156	2612156				
11/16/2006	2613409	2613409				
11/16/2006	2614032	2614032				
11/16/2006	2614040	2614040				
11/16/2006	2614155	2614155				
11/16/2006	2614353	2614353				
11/16/2006	2614733	2614733				
11/16/2006	2618263	2618263				

Parallel Testing

	UW0011CC		UW0012C2		UW0013CC	
Date	**Production System**	**System Under Discussion**	**Production System**	**System Under Discussion**	**Production System**	**System Under Discussion**
11/17/2006					2568198	2516021
					2516021	2568198
11/16/2006					2421290	2421290
					2422960	2422960
					2568757	2568757

Parallel Testing

	UW0014C2		UW0015CC		UW0016C2	
Date	**Production System**	**System Under Discussion**	**Production System**	**System Under Discussion**	**Production System**	**System Under Discussion**
11/17/2006						
11/16/2006			2470611	2470611		

Parallel Testing

	UW0017CC		UW0018C2		UW0026CC		UW0027C2	
Date	**Production System**	**System under Discussion**	**Production System**	**System under Discussion**	**Production System**	**System under Discussion**	**Production System**	**System under Discussion**
11/17/2006					2423535	2423535		
11/16/2006	2517092	2517092			2422998 2561909	2422998 2561909	2614160	2614160

Certificate of mail

Printed on the next pages is the certificate of mail. This report is generated in a daily process on Hyperion Intelligence and exported to the CWG network as a PDF file, it can then be printed on the central printer in the Des Moines office.

Data from the test system was used to create this report. The policy numbers you see along with the insured names and agency names are ficticious. You will also notice the dates on this form range from 10/30/2006 to 11/30/2006. Because the test system has a limited amount of data we had to pull 30 days of policies to get results comparable to a day of activity on the production system.

Certificate of Mail Report for PN6010

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
1	DXS	5003208 - 20	N. & L. Lovell 604 NW 10th Street Ontario, OR 97914	0.39	0.30								

Certificate of Mail Report for PN6113

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
1	FDK	2407882 - 23	St Jacob Township Fire Protection District C/O Ed Miller PO Box 314 St Jacob, IL 62281	0.39	0.30								
2	FDK	2407883 - 23	Buckley Fire Protection District 102 E Lincoln P O Box 145 Buckley, IL 60918	0.39	0.30								
3	FDK	2425580 - 24	Elizabeth Community Ambulance Service An Illinois Not for Profit Corporation (P O Box 325) Elizabeth, IL 61028	0.39	0.30								
4	MCP	5003524 - 21	Trailer Interchange PO Box 191 Eagle, NE 68347	0.39	0.30								
5	BOP	5004979 - 20	DKE E/Q Zipcode & Print 1 Main Cary, IL 60013	0.39	0.30								
6	CC	5008011 - 20	Dennis Beckly Box 129 Chester, IL 62233	0.39	0.30								
7	CWP	5008017 - 20	IL HINO TEST 1 MAIN ST ROCKFORD, IL 61100	0.39	0.30								

Certificate of Mail Report for PN6115

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill.
POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
1	CDP	5003008 - 20	Reg Test Bedford DBA Old West Petroleum PO Box 105 Glen Haven, CO 80532	0.39	0.30								
2	CDP	5003010 - 20	Reg Test Mike & Archie Brown DBA Brown's Feed & Chemical 404 South 5th Street Carmi, IL 62821	0.39	0.30								
3	CDP	5003013 - 20	Reg Test Mike Ahlers DBA Ahlers Oil Company 918 6th St Sheldon, IA 51201	0.39	0.30								
4	CDP	5003015 - 20	Reg Test Ed & Bob Hamblin DBA Hamblin Petroleum Products 425 N Mahaffie Olathe, KS 66061	0.39	0.30								
5	CDP	5003032 - 20	Reg Test Randy Acklin DBA Custom Feed & Tack PO Box 68 Barnard, MO 64423	0.39	0.30								
6	CDP	5003036 - 20	Reg Test Willard & Judy Bartels DBA Bartels Service 207 Ash Street Tobias, NE 68453	0.39	0.30								
7	CDP	5003043 - 20	Reg Test Peter K Chase DBA Chase Oil PO Box 908 Waukesha, WI 53187	0.39	0.30								

Certificate of Mail Report for PN6115

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill.
POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
8	CDP	5003394 - 20	Van Wyhe Testing terrorism 100 South Main Road Clarence, IL 60960	0.39	0.30								
9	CDP	5004067 - 20	Norma Van Wyhe Testing Deductibles South Holland, IL 60473	0.39	0.30								
10	CDP	5004153 - 20	Norma Van Wyhe Testing deductibles 191 Main Columbus, IN 47201	0.39	0.30								
11	CDP	5004158 - 21	Norma Van Wyhe 32950380 KNiss Ave Bailey, CO 80421	0.39	0.30								
12	CDP	5005294 - 20	BMM inst testing several installs PO Box 68 Barnard, MO 64423	0.39	0.30								
13	CDP	5005434 - 20	BMM auto auto DOC coverage 207 Ash Street Tobias, NE 68453	0.39	0.30								
14	CDP	5005953 - 20	bmm auto year 2006 Racine, WI 53400	0.39	0.30								

Certificate of Mail Report for PN6115

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
15	CDP	5006258 - 20	LJD-OR-Testing SLWR 6JTL4K Acorn Park, OR 97402	0.39	0.30								
16	CDP	5006409 - 20	LJD-OR-Testing SLWR 6JTL4K Acorn Park, OR 97402	0.39	0.30								
17	CDP	5006509 - 20	LJD-OR-Testing SLWR 6JTL4K Acorn Park, OR 97402	0.39	0.30								
18	CDP	5006511 - 20	LJD-WY-Testing SLWR 6JTL4K Acme, WY 82839	0.39	0.30								
19	CDP	5006605 - 20	LJD-WY-Testing SLWR 6JTL4K Acme, WY 82839	0.39	0.30								
20	CDP	5006617 - 20	LJD-WY-Testing SLWR 6JTL4K Acme, WY 82839	0.39	0.30								
21	CDP	5006828 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for OR Cairo, OR 97914	0.39	0.30								

Certificate of Mail Report for PN6115

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
22	CDP	5006832 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for UT Beeton, UT 84309	0.39	0.30								
23	CDP	5006833 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for WY Daniel, WY 83115	0.39	0.30								
24	CDP	5007672 - 20	BAGE 6JPMWK JRG 748 Mechanical Breakdown Luverne, MN 56156	0.39	0.30								
25	CDP	5007674 - 20	BAGE 6JPMWK JRG 748 Mechanical Breakdown Sioux Falls, SD 57103	0.39	0.30								
26	CDP	5009156 - 20	BMM UT ded valid auto ded Clinton, UT 84015	0.39	0.30								
27	CDP	5009188 - 20	bmm rmr forms Ten Roundwind Road Cairo, OR 97914	0.39	0.30								
28	CDP	5009220 - 20	bmm rmr forms Ten Roundwind Road Daniel, WY 83115	0.39	0.30								

Certificate of Mail Report for PN6115

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
29	CDP	5009233 - 20	bmm forms UT Ten Roundwind Road Beeton, UT 84309	0.39	0.30								
30	CDP	5009257 - 20	Van Wyhe Testing debris removal Baker, MT 59313	0.39	0.30								
31	CDP	5010014 - 20	Van Wyhe Testing debris removal Luverne, MN 56156	0.39	0.30								
32	CDP	5010256 - 20	Gyberg IL0985 Madison, WI 53700	0.39	0.30								
33	CDP	5010257 - 20	Gyberg IL0985 Luverne, MN 56156	0.39	0.30								
34	CDP	5010259 - 20	Gyberg IL0985 Fargo, ND 58102	0.39	0.30								
35	CDP	5010260 - 20	Gyberg IL0985 Sioux Falls, SD 57103	0.39	0.30								

Certificate of Mail Report for PN6115

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
36	CDP	5010261 - 20	Gyberg IL0985 Medill, MO 63445	0.39	0.30								
37	CDP	5010262 - 20	Gyberg IL0985 Boise, ID 83637	0.39	0.30								

Certificate of Mail Report for PN6118

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
1	CNP	2603326 - 20	Munce test forms Main Sioux Falls, SD 57103	0.39	0.30								
2	CWP	5002721 - 20	LDD Terror 123 Terror Boise, ID 83704	0.39	0.30								
3	CWP	5002724 - 20	LDD Terror 123 Terror Seattle, WA 98115	0.39	0.30								
4	CWP	5002726 - 20	LDD Terror 123 Terror Salt Lake City, UT 84120	0.39	0.30								
5	CWP	5002728 - 20	LDD Terror 123 Terror Portland, OR 97200	0.39	0.30								
6	CWP	5002732 - 20	LDD Terror 123 Terror Butte, MT 59701	0.39	0.30								
7	CGL	5004876 - 21	Munce Stop Gap Main Fargo, ND 58102	0.39	0.30								

Certificate of Mail Report for PN6118

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
8	CNP	5004981 - 20	DKE Construction Company 1212 Main Lees Summit, MO 64086	0.39	0.30								
9	CWP	5006620 - 21	Munce test Try to add bopaddin 900 Main to cwp...Should get error Sioux Falls, SD 57101	0.39	0.30								
10	CWP	5006624 - 21	Munce test CRIME forms 900 Main for Linda Sioux Falls, SD 57101	0.39	0.30								
11	CNP	5006952 - 20	LDD CNP Forms 123 Pigeon Boise, ID 83704	0.39	0.30								
12	CNP	5006953 - 20	LDD CNP Forms 123 Pigeon Seattle, WA 98115	0.39	0.30								
13	CWP	5007700 - 30	Munce Test for BJV 91583 Main Also Test Builders risk sewer BU Sioux Falls, SD 57103	0.39	0.30								
14	IMD	5007997 - 20	DKE COV F & Water Dam 1 Main Waukee, IA 50263	0.39	0.30								

Certificate of Mail Report for PN6118

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
15	IMD	5008032 - 20	D RaddleIMD 1 Main Adrian, MN 56110	0.39	0.30								
16	CWP	5008076 - 20	SAW Terror 123 Terror Portland, OR 97200	0.39	0.30								
17	CWP	5010104 - 20	LDD Earthquake III 123 Earthquake Laramie, WY 82070	0.39	0.30								
18	CWP	5010114 - 20	LDD Earthquake III 123 Earthquake Laramie, WY 82070	0.39	0.30								

Certificate of Mail Report for PN6119

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
1	MCP	5001457 - 22	Receipts/Value 640 Third St Box 74 Westside, IA 51467	0.39	0.30								
2	MCP	5001464 - 23	Composite/Units 640 Third St Box 74 Westside, IA 51467	0.39	0.30								
3	MCP	5001480 - 20	Hastings Trucking 11258 State St Des Moines, IA 50309	0.39	0.30								
4	CA	5002700 - 20	LDD Auto forms 123 Auto Sheridan, WY 82801	0.39	0.30								
5	CWP	5002728 - 20	LDD Terror 123 Terror Portland, OR 97200	0.39	0.30								
6	CA	5003416 - 21	LDD PIP 123 PIP Portland, OR 97200	0.39	0.30								
7	MCP	5003524 - 21	Trailer Interchange PO Box 191 Eagle, NE 68347	0.39	0.30								

Certificate of Mail Report for PN6119

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
8	CDP	5005294 - 20	BMM inst testing several installs PO Box 68 Barnard, MO 64423	0.39	0.30								
9	ABP	5005336 - 20	bmm changes testing new decs Sioux Falls, SD 57103	0.39	0.30								
10	CA	5005736 - 21	Munce Test BMC90 Print Main Sioux Falls, SD 57101	0.39	0.30								
11	CA	5005738 - 21	Munce Test BMC90 Print Main Sioux Falls, SD 57101	0.39	0.30								
12	PEP	5005793 - 20	DKE Test Commissions 1 Main Waukee, IA 50263	0.39	0.30								
13	PEP	5005837 - 20	DKE Test Commissions 1 Main Waukee, IA 50263	0.39	0.30								
14	CDP	5006258 - 20	LJD-OR-Testing SLWR 6JTL4K Acorn Park, OR 97402	0.39	0.30								

Certificate of Mail Report for PN6119

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
15	CDP	5006409 - 20	LJD-OR-Testing SLWR 6JTL4K Acorn Park, OR 97402	0.39	0.30								
16	CDP	5006509 - 20	LJD-OR-Testing SLWR 6JTL4K Acorn Park, OR 97402	0.39	0.30								
17	CDP	5006511 - 20	LJD-WY-Testing SLWR 6JTL4K Acme, WY 82839	0.39	0.30								
18	CDP	5006605 - 20	LJD-WY-Testing SLWR 6JTL4K Acme, WY 82839	0.39	0.30								
19	CDP	5006617 - 20	LJD-WY-Testing SLWR 6JTL4K Acme, WY 82839	0.39	0.30								
20	CA	5006658 - 21	Fuerstenberg MCS90B 1 West Main Sioux Falls, SD 57103	0.39	0.30								
21	CA	5006675 - 20	DKE GARAGEKEEPERS 1 Main Waukee, IA 50263	0.39	0.30								

Certificate of Mail Report for PN6119

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
22	CWP	5006688 - 20	DKE Test 2nd Memo Copy 1 Main Waukee, IA 50263	0.39	0.30								
23	CWP	5006817 - 20	DKE Test Memo Copy 1 Main Waukee, IA 50263	0.39	0.30								
24	CWP	5006827 - 20	DKE Test MC1612c 1 Main Waukee, IA 50263	0.39	0.30								
25	CDP	5006828 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for OR Cairo, OR 97914	0.39	0.30								
26	CDP	5006833 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for WY Daniel, WY 83115	0.39	0.30								
27	ABP	5006845 - 20	Van Wyhe Ten Roundwind Road testing abp eff for OR Cairo, OR 97914	0.39	0.30								
28	ABP	5006893 - 20	Van Wyhe Ten Roundwind Road testing abp eff for wy Daniel, WY 83115	0.39	0.30								

Certificate of Mail Report for PN6119

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
29	CA	5007681 - 21	Rauk - Test BMC90, MC1632a, MC2444a 20725 Jewel Drive Sioux Falls, SD 57102	0.39	0.30								
30	CA	5007705 - 21	Rauk Test PIP stacking 789 Test Sioux Falls, SD 57102	0.39	0.30								
31	CWP	5008017 - 20	IL HINO TEST 1 MAIN ST ROCKFORD, IL 61100	0.39	0.30								
32	CWP	5008076 - 20	SAW Terror 123 Terror Portland, OR 97200	0.39	0.30								
33	ABP	5008547 - 21	BMM ABPTRIA TRIA testing Garner, IA 50438	0.39	0.30								
34	CDP	5009188 - 20	bmm rmr forms Ten Roundwind Road Cairo, OR 97914	0.39	0.30								
35	ABP	5009217 - 20	bmm forms Ten Roundwind Road Cairo, OR 97914	0.39	0.30								

Certificate of Mail Report for PN6119

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
36	CDP	5009220 - 20	bmm rmr forms Ten Roundwind Road Daniel, WY 83115	0.39	0.30								
37	ABP	5009232 - 20	bmm rmr forms Ten Roundwind Road Daniel, WY 83115	0.39	0.30								
38	CA	5022600 - 20	AEB Iowa Rates eff 1/1/07 1234 Main St Clive, IA 50325	0.39	0.30								
39	MCP	5023395 - 21	Dab Test 2366947 2260 220th St Humboldt, IA 50548	0.39	0.30								
40	MCP	5023449 - 21	Dab Test 2366947 2260 220th St Slater, MO 65349	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
1	CNP	2603326 - 20	Munce test forms Main Sioux Falls, SD 57103	0.39	0.30								
2	CWP	5002554 - 20	Shar Stables 12250 S 82nd Roca, NE 68430	0.39	0.30								
3	BOP	5002563 - 20	Shar Test Policy PO Box 410 Elm Creek, NE 68836	0.39	0.30								
4	CP	5002695 - 20	LDD Forms 123 Forms Laramie, WY 82070	0.39	0.30								
5	CWP	5002721 - 20	LDD Terror 123 Terror Boise, ID 83704	0.39	0.30								
6	CWP	5002724 - 20	LDD Terror 123 Terror Seattle, WA 98115	0.39	0.30								
7	CWP	5002726 - 20	LDD Terror 123 Terror Salt Lake City, UT 84120	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill.
POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
8	CWP	5002728 - 20	LDD Terror 123 Terror Portland, OR 97200	0.39	0.30								
9	CWP	5002732 - 20	LDD Terror 123 Terror Butte, MT 59701	0.39	0.30								
10	FRP	5002735 - 20	LDD Terror 123 Terror Seattle, WA 98115	0.39	0.30								
11	CP	5002776 - 20	Shar 1111 Old Cheney Lincoln, NE 68512	0.39	0.30								
12	CP	5003393 - 20	LDD Company 123 Company Salt Lake City, UT 84120	0.39	0.30								
13	BOP	5003402 - 21	LDD BCEG 123 BCEG Seattle, WA 98115	0.39	0.30								
14	BOP	5003406 - 21	LDD BCEG 123 BCEG Boise, ID 83704	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
15	FRP	5003408 - 21	LDD BCEG 123 BCEG Seattle, WA 98115	0.39	0.30								
16	BOP	5003413 - 20	LDD BOPDO 123 DO Laramie, WY 82070	0.39	0.30								
17	BOP	5004526 - 21	Fuerstenberg EQ Zip codes 1 West Main Luverne, MN 56156	0.39	0.30								
18	FRP	5004573 - 21	Fuerstenberg EQ Zip codes 1 West Main Luverne, MN 56156	0.39	0.30								
19	BOP	5004871 - 21	Fuerstenberg 1 West Main Luverne, MN 56156	0.39	0.30								
20	BOP	5004979 - 20	DKE E/Q Zipcode & Print 1 Main Cary, IL 60013	0.39	0.30								
21	CNP	5004981 - 20	DKE Construction Company 1212 Main Lees Summit, MO 64086	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
22	BOP	5004994 - 20	D RaddleE/Q Zipcode & Print 1 Main Luverne, MN 56156	0.39	0.30								
23	BOP	5005007 - 21	Fuerstenberg Test EQ 1 West Main Minneapolis, MN 55401	0.39	0.30								
24	BOP	5005009 - 21	Fuerstenberg EQ 1 West Main Butte, ND 58723	0.39	0.30								
25	BOP	5005017 - 21	Fuerstenberg EQ 1 West Main Butte, ND 58723	0.39	0.30								
26	BOP	5005019 - 21	Fuerstenberg Test EQ 1 West Main Minneapolis, MN 55400	0.39	0.30								
27	BOP	5005021 - 21	Fuerstenberg Test EQ 1 West Main Carlock, SD 57533	0.39	0.30								
28	BOP	5005029 - 21	Fuerstenberg Test EQ 1 West Main Clark, SD 57225	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
29	CDP	5005294 - 20	BMM inst testing several installs PO Box 68 Barnard, MO 64423	0.39	0.30								
30	ABP	5005336 - 20	bmm changes testing new decs Sioux Falls, SD 57103	0.39	0.30								
31	CWP	5005805 - 20	D RaddleSRUP TSHE 6JTL3S Luverne, MN 56156	0.39	0.30								
32	FRP	5006616 - 21	Munce FRP Test Test Main Sioux Falls, SD 57101	0.39	0.30								
33	CWP	5006620 - 21	Munce test Try to add bopaddin 900 Main to cwp...Should get error Sioux Falls, SD 57101	0.39	0.30								
34	CWP	5006624 - 21	Munce test CRIME forms 900 Main for Linda Sioux Falls, SD 57101	0.39	0.30								
35	BOP	5006659 - 21	Munce BOP Blanket test Main Sioux Falls, SD 57104	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
36	CWP	5006688 - 20	DKE Test 2nd Memo Copy 1 Main Waukee, IA 50263	0.39	0.30								
37	CWP	5006817 - 20	DKE Test Memo Copy 1 Main Waukee, IA 50263	0.39	0.30								
38	CWP	5006827 - 20	DKE Test MC1612c 1 Main Waukee, IA 50263	0.39	0.30								
39	CDP	5006828 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for OR Cairo, OR 97914	0.39	0.30								
40	CDP	5006832 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for UT Beeton, UT 84309	0.39	0.30								
41	CDP	5006833 - 20	Van Wyhe Ten Roundwind Road testing cdp eff for WY Daniel, WY 83115	0.39	0.30								
42	ABP	5006845 - 20	Van Wyhe Ten Roundwind Road testing abp eff for OR Cairo, OR 97914	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
43	FRP	5006855 - 20	DKE CW2418 1 Main Waukee, IA 50263	0.39	0.30								
44	ABP	5006893 - 20	Van Wyhe Ten Roundwind Road testing abp eff for wy Daniel, WY 83115	0.39	0.30								
45	BOP	5006901 - 20	agency copy main Altoona, IA 50009	0.39	0.30								
46	CWP	5007700 - 30	Munce Test for BJV 91583 Main Also Test Builders risk sewer BU Sioux Falls, SD 57103	0.39	0.30								
47	BOP	5007799 - 21	Munce BP0457 test Main Sioux Falls, SD 57101	0.39	0.30								
48	BOP	5007801 - 21	Munce BP0457 test Main Luverne, MN 56156	0.39	0.30								
49	FRP	5007808 - 21	Munce BP0457 test Main Fargo, ND 58102	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill.
POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
50	BOP	5007813 - 21	Munce CW2555 test Main Sioux Falls, SD 57101	0.39	0.30								
51	CWP	5008076 - 20	SAW Terror 123 Terror Portland, OR 97200	0.39	0.30								
52	CWP	5008077 - 20	SAW Rsk Typ 100 Main Des Moines, IA 50310	0.39	0.30								
53	CWP	5008303 - 20	D Raddlecpgl Adrian, MN 56110	0.39	0.30								
54	ABP	5008547 - 21	BMM ABPTRIA TRIA testing Garner, IA 50438	0.39	0.30								
55	CDP	5009156 - 20	BMM UT ded valid auto ded Clinton, UT 84015	0.39	0.30								
56	CDP	5009188 - 20	bmm rmr forms Ten Roundwind Road Cairo, OR 97914	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill.
POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
57	ABP	5009217 - 20	bmm forms Ten Roundwind Road Cairo, OR 97914	0.39	0.30								
58	CDP	5009220 - 20	bmm rmr forms Ten Roundwind Road Daniel, WY 83115	0.39	0.30								
59	ABP	5009232 - 20	bmm rmr forms Ten Roundwind Road Daniel, WY 83115	0.39	0.30								
60	CDP	5009233 - 20	bmm forms UT Ten Roundwind Road Beeton, UT 84309	0.39	0.30								
61	ABP	5009234 - 20	bmm rmr CP0415 Ten Roundwind Road Beeton, UT 84309	0.39	0.30								
62	CP	5009343 - 24	LDD Ancienter 123 Ancienter Seattle, WA 98115	0.39	0.30								
63	CWP	5010017 - 20	Norma Van Wyhe Testing various items Great Falls, MT 59401	0.39	0.30								

Certificate of Mail Report for PN6121

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill.
POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
64	CWP	5010104 - 20	LDD Earthquake III 123 Earthquake Laramie, WY 82070	0.39	0.30								
65	CWP	5010114 - 20	LDD Earthquake III 123 Earthquake Laramie, WY 82070	0.39	0.30								

Certificate of Mail Report for PN6128

10/30/2006 to 11/30/2006

Continental Western Group
11201 Douglas Avenue
P.O. Box 1594
Des Moines, IA 50306-1594

Affix Stamp here if issued as certificate of mailing or for additional copies of this bill. POSTMARK AND DATE OF

Line	Sym	Article Number	Name, Street, and Post Office Address	Postage	Fee	Handling Charge	Act. Value (If Regis.)	Insured Value	Due Sender If COD	R.R. Fee	S.D. Fee	S.H. Fee	Rest. Del. Fee Remarks
1	CWP	5002724 - 20	LDD Terror 123 Terror Seattle, WA 98115	0.39	0.30								
2	CNP	5006953 - 20	LDD CNP Forms 123 Pigeon Seattle, WA 98115	0.39	0.30								

Total Number Of Pieces Listed By Sender 170

Total Number Of Pieces Received at Post Office

POSTMASTER, PER (Name of receiving employee)

Letter status report

The letter status report is created as a reference to be used by the distribution serves staff. It has most of the same data was the letter spreadsheet but has additional columns that define the status of a letter (active, expired or pending) and the date it will start and stop printing. The data for the report is created in the master query and stored in the SQL Server table phn_letter_data.

This report is created on Hyperion Intelligence and is not printed as part of the daily process. It is being created on an as-needed basis for the distribution services staff. As mentioned in the definitions the distribution services staff consists of the mail room and print operations.

Policyholder Notice and Renewal Letter Information

Form Number	Form Edition	Status	Start Print Date	End Print Date	Type	Mail To	Pages	Effective Date	Expiration Date	Term	Days Prior	Copies	Duplex	Return Envelope	Imaged	Cert Of Mail	Inet Copy
PN6010	07-05	ACTIVE	03/11/06	03/11/07	Policyholder Notice	Insured	1	05/15/06	05/15/07	1 year	65	1	Simplex	N	N	Y	Y
PN6113	05-06	ACTIVE	01/31/06	01/31/07	Policyholder Notice	Insured	1	05/01/06	05/01/07	1 year	90	1	Simplex	N	N	Y	Y
PN6115	01-06	ACTIVE	12/26/05	12/26/06	Policyholder Notice	Insured	1	03/01/06	03/01/07	1 year	65	1	Simplex	N	N	Y	Y
PN6118	06-06	ACTIVE	06/03/06	06/03/07	Policyholder Notice	Insured	1	09/01/06	09/01/07	1 year	90	1	Simplex	N	N	Y	Y
PN6119	10-06	ACTIVE	11/03/06	11/03/07	Policyholder Notice	Insured	6	02/01/07	02/01/08	1 year	90	1	Duplex	N	N	Y	Y
PN6119	10-06	ACTIVE	10/03/06	10/03/07	Policyholder Notice	Insured	6	01/01/07	01/01/08	1 year	90	1	Duplex	N	N	Y	Y
PN6119	10-06	ACTIVE	07/03/06	07/03/07	Policyholder Notice	Insured	6	10/01/06	10/01/07	1 year	90	1	Duplex	N	N	Y	Y
PN6119	10-06	PENDING	12/01/06	12/02/07	Policyholder Notice	Insured	6	03/01/07	03/01/08	1 year	90	1	Duplex	N	N	Y	Y
PN6120	06-06	ACTIVE	06/03/06	06/03/07	Policyholder Notice	Insured	1	09/01/06	09/01/07	1 year	90	1	Simplex	N	N	Y	Y
PN6121	06-06	ACTIVE	07/03/06	07/03/07	Policyholder Notice	Insured	2	10/01/06	10/01/07	1 year	90	1	Simplex	N	N	Y	Y
PN6128	05-06	ACTIVE	07/17/06	07/17/07	Policyholder Notice	Insured	1	10/15/06	10/15/07	1 year	90	1	Simplex	N	N	Y	Y
PN6130	06-06	ACTIVE	09/02/06	09/02/07	Policyholder Notice	Insured	1	12/01/06	12/01/07	1 year	90	1	Simplex	N	N	Y	Y
PN6131	07-06	ACTIVE	09/02/06	09/02/07	Policyholder Notice	Insured	1	12/01/06	12/01/07	1 year	90	1	Simplex	N	N	Y	Y
PN6133	07-06	ACTIVE	11/03/06	11/03/07	Policyholder Notice	Insured	1	02/01/07	02/01/08	1 year	90	1	Simplex	N	N	Y	Y
PN6134	07-06	ACTIVE	11/03/06	11/03/07	Policyholder Notice	Insured	1	02/01/07	02/01/08	1 year	90	1	Simplex	N	N	Y	Y
PN6141	07-06	ACTIVE	11/03/06	11/03/07	Policyholder Notice	Insured	1	02/01/07	02/01/08	1 year	90	1	Simplex	N	N	Y	Y
UW0011CC	06/20	ACTIVE	01/01/01	12/31/99	Renewal Letter	Agent	1	01/01/01	12/31/99	on-going	120	2	Simplex	N	Y	N	N
UW0012C2	06/20	ACTIVE	01/01/01	12/31/99	Renewal Letter	Insured	1	01/01/01	12/31/99	on-going	90	2	Simplex	Y	Y	N	N

Form Number	Form Edition	Status	Start Print Date	End Print Date	Type	Mail To	Pages	Effective Date	Expiration Date	Term	Days Prior	Copies	Duplex	Return Envelope	Imaged	Cert Of Mail	Inet Copy
UW0013CC	06/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Agent	1	10/01/03	12/31/99	on-going	120	2	Simplex	N	Y	N	N
UW0014C2	06/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Insured	1	10/01/03	12/31/99	on-going	90	2	Simplex	Y	Y	N	N
UW0015CC	06/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Agent	1	10/01/03	12/31/99	on-going	120	2	Simplex	N	Y	N	N
UW0016C2	06/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Insured	1	10/01/03	12/31/99	on-going	90	2	Simplex	Y	Y	N	N
UW0017CC	06/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Agent	1	10/01/03	12/31/99	on-going	120	2	Simplex	N	Y	N	N
UW0018C2	06/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Insured	1	10/01/03	12/31/99	on-going	90	2	Simplex	Y	Y	N	N
UW0020	07/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Agent	1	10/01/03	12/31/99	on-going	90	2	Duplex	N	Y	N	N
UW0021	07/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Insured	1	10/01/03	12/31/99	on-going	90	2	Duplex	N	Y	N	N
UW0022	07/20	ACTIVE	10/01/03	12/31/99	Renewal Letter	Agent	1	10/01/03	12/31/99	on-going	90	2	Simplex	N	Y	N	N
UW0026CC	10/20	ACTIVE	01/01/01	12/31/99	Renewal Letter	Agent	1	01/01/01	12/31/99	on-going	120	2	Simplex	N	Y	N	N
UW0027C2	10/20	ACTIVE	01/01/01	12/31/99	Renewal Letter	Insured	1	01/01/01	12/31/99	on-going	90	2	Simplex	N	Y	N	N
UW0033	06-05	ACTIVE	01/01/01	12/31/99	Renewal Letter	Agent	1	01/01/01	12/31/99	on-going	120	2	Duplex	Y	Y	N	N
UW0033A	06-05	ACTIVE	01/01/01	12/31/99	Renewal Letter	Agent	1	01/01/01	12/31/99	on-going	120	2	Duplex	Y	Y	N	N
UW0037	10-05	ACTIVE	05/01/06	12/31/99	Renewal Letter	Agent	1	05/01/06	12/31/99	on-going	120	2	Simplex	Y	Y	N	N
UW0040	02-06	ACTIVE	01/01/06	12/31/99	Renewal Letter	Agent	1	01/01/06	12/31/99	on-going	65	2	Simplex	Y	Y	N	N
UW0040	02-06	ACTIVE	01/01/06	12/31/99	Renewal Letter	Agent	1	01/01/06	12/31/99	on-going	60	2	Simplex	Y	Y	N	N
PN6051	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	2	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6052	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6054	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6055	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y

Form Number	Form Edition	Status	Start Print Date	End Print Date	Type	Mail To	Pages	Effective Date	Expiration Date	Term	Days Prior	Copies	Duplex	Return Envelope	Imaged	Cert Of Mail	Inet Copy
PN6056	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6057	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6058	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6059	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6060	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	6	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6061	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6062	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	N	N	Y	Y
PN6064	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6066	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	2	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6067	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6073	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6075	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6081	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6085	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	6	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6087	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6091	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6092	07-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	4	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6098	08-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	6	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6099	08-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	6	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y
PN6100	08-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	6	12/15/05	12/15/06	1 year	90	1	Duplex	1	N	Y	Y

Form Number	Form Edition	Status	Start Print Date	End Print Date	Type	Mail To	Pages	Effective Date	Expiration Date	Term	Days Prior	Copies	Duplex	Return Envelope	Imaged	Cert Of Mail	Inet Copy
PN6104	02-05	EXPIRED	05/03/05	05/03/06	Policyholder Notice	Insured	2	08/01/05	08/01/06	1 year	90	1	Duplex	N	N	Y	Y
PN6105	01-06	EXPIRED	11/02/05	11/02/06	Policyholder Notice	Insured	4	01/01/06	01/01/07	1 year	60	1	Duplex	N	N	Y	Y
PN6108	09-05	EXPIRED	07/03/05	07/03/06	Policyholder Notice	Insured	1	10/01/05	10/01/06	1 year	90	1	Simplex	N	N	Y	Y
PN6114	11-05	EXPIRED	09/16/05	09/16/06	Policyholder Notice	Insured	1	12/15/05	12/15/06	1 year	90	1	Simplex	N	N	Y	Y
PN6116	11-05	EXPIRED	11/17/05	09/17/06	Policyholder Notice	Insured	1	02/15/06	12/16/06	1 year	90	1	Simplex	N	N	Y	Y
UW0030IA	03-04	EXPIRED	08/03/05	08/03/06	Renewal Letter	Agent	1	12/01/05	12/01/06	1 year	120	2	Duplex	N	Y	N	N
UW0030IL	03-04	EXPIRED	06/03/05	06/03/06	Renewal Letter	Agent	1	10/01/05	10/01/06	1 year	120	2	Duplex	N	Y	N	N
UW0030KS	03-04	EXPIRED	08/03/05	08/03/06	Renewal Letter	Agent	1	12/01/05	12/01/06	1 year	120	2	Simplex	N	Y	N	N
PN6137	07-06	PENDING	12/01/06	12/02/07	Policyholder Notice	Insured	1	03/01/07	03/01/08	1 year	90	1	Simplex	N	N	Y	Y
PN6138	07-06	PENDING	12/01/06	12/02/07	Policyholder Notice	Insured	1	03/01/07	03/01/08	1 year	90	1	Simplex	N	N	Y	Y

Status Definition: ACTIVE- Today's date falls between the Start Printing Date and End Printing Date
EXPIRED - Today's date is on or after the End Printing Date
PENDING - The form has been put into the system but today's date is prior to the Start Printing Date

Policy list for renewal letters

The policy list is similar to the certificate of mail, except it is created for the renewal letters. Since the renewal letters do not require a certificate of mail we created this report to be used internally by the mail staff and the underwriting department to verify that a letter was sent. Like the certificate of mail the policy list gets created in a daily job on Hyperion Intelligence and is exported to the CWG network as a PDF file. It is then printed on the central printer in Des Moines.

The data for this report was taken from the test system so the policy numbers, insured names and agent names are all ficticious. As with the certificate of mail we generated this report from 30 days of policies on the test system.

Policy List for UW0015CC

Mail Date: 10/30/2006

Seq	Policy Number	Mail To	Agency Number	Agency Name	Insured Name
1	5008699	Agent	5669	THE ESTENSON COMPANY INC	Henry Geggerson

Policy List for UW0016C2

Mail Date: 10/30/2006

Seq	Policy Number	Mail To	Agency Number	Agency Name	Insured Name
1	5004638	Insured	5644	TRI-STATE AGENCY OF MN	Peter Mathson

Policy List for UW0018C2

Mail Date: **10/30/2006**

Seq	Policy Number	Mail To	Agency Number	Agency Name	Insured Name
1	5004192	Insured	5644	TRI-STATE AGENCY OF MN	John Coiner

Collector car vehicle letter

We have ten letters that are sent out on collector car policies that print variable data about vehicles. These letters did not work well print with MS Word so we created a job for each letter in Hyperion Intelligence and treated them as a report.

Like the certificate of mail and policy list these 10 letters are created in a daily job on Intelligence, exported as PDF files to the CWG network and then printed on the central printer in Des Moines. We used the form UW0015CC as an example; we did not print all ten letters for the report because they are all very similar in structure, printing all of them would have been redundant.

The data for this letter was pulled from the test system so the insured, agency and vehicle data are all ficticious. The second page prints what is called a "dummy" letter. All letters generate a dummy copy at the end of the file so the print operations staff knows they are at the end of a group of letters.

10/30/2006

THE ESTENSON COMPANY INC
419 E MAIN
PO BOX 868
LUVERNE, MN 56156-0868

RE: Henry Geggerson
Box 1120

Sioux Falls, SD 57101
Collector Car 5008699
Modern Collectable or Exotic Vehicle

Dear Agent:

We are in the process of reviewing the above-referenced Collector Car policy for renewal. The policy premium is based on the insured's prior agreement to limit the listed vehicle(s) use to a maximum of 2500 miles per year. We need a current odometer reading prior to renewing the policy in order to continue coverage on the vehicle(s) listed below.

Current odometer reading for:			Current Odometer Reading
2005 PORSCHE	Carrera GT	WP0CA29895L0(	_____

This information must be received prior to 12/31/2006 in order to continue offering coverage on the above-listed vehicle(s).

Sincerely,

AL STOAKES
Collector Car Department

TRI-STATE OFFICE

Phone: 507-283-9561

UW 00 15 CC 06 03

10/30/2006

DUMMY AGCY
DUMMY ADDR1
DUMMY ADDR2
DUMMY CITY, ST XXXXX-XXXX

RE: DUMMY INSD
DUMMY ADDR1
DUMMY ADDR2
DUMMY ADDR3
DUMMY CITY, ST XXXXX-XXXX
Collector Car 9999999
Modern Collectable or Exotic Vehicle

Dear Agent:

We are in the process of reviewing the above-referenced Collector Car policy for renewal. The policy premium is based on the insured's prior agreement to limit the listed vehicle(s) use to a maximum of 2500 miles per year. We need a current odometer reading prior to renewing the policy in order to continue coverage on the vehicle(s) listed below.

Current odometer reading for:			Current Odometer Reading
9999 DUMMY MAKE	DUMMY MDL	DUMMY SER#	_____

This information must be received prior to 12/31/2006 in order to continue offering coverage on the above-listed vehicle(s).

Sincerely,

XXX
Collector Car Department

TRI-STATE OFFICE

Phone: 507-283-9561

UW 00 15 CC 06 03

Policyholder notice word document

The next two pages provide an example of a policyholder notice that print from MS Word. The example we provided in PN6010, it is a one page letter and generated two copies, one is from a policy on the test system and the other is the dummy letter. Again, because the data was pulled from the test system the insured information, agency information and policy numbers are ficitious.

The letters generated in MS Word use the phn_policy_data table on SQL Server as their data source and are printed on the central printer in Des Moines by a macro that is run daily.

Continental Western Group

Continental Western Insurance Company • Tri-State Insurance Company of Minnesota • Union Insurance Company
11201 Douglas Avenue • PO Box 1594 • Des Moines, Iowa 50306-1594 • (800)235-2942 • (515) 473-3000

11/1/2006

1
N. & L. Lovell
604 NW 10th Street
Ontario, OR 97914

RE: Policy Number: 5003208
Expiration Date: 1/15/2007

NOTICE TO POLICYHOLDERS

Dear Policyholder:

Your Liability renewal policy, including your Umbrella Liability renewal policy, if applicable, will contain an endorsement that excludes coverage for all liability claims resulting from Silica. Please read this endorsement carefully when you receive your renewal policy as this notice is not intended to replace the provisions of your policy.

Please note that your renewal policy will be forthcoming.

We appreciate your business and look forward to serving you through the Continental Western Group of companies. If you have any questions, please contact your agent.

HULL & CO MID-AMERICA INC.
6940 O ST STE, 404
PO BOX 5788
LINCOLN, NE 68505
Phone: (402) 466-9400
Agency No.: 9547

Continental Western Group

Continental Western Insurance Company • Tri-State Insurance Company of Minnesota • Union Insurance Company
11201 Douglas Avenue • PO Box 1594 • Des Moines, Iowa 50306-1594 • (800)235-2942 • (515) 473-3000

11/1/2006

2
DUMMY INSD
DUMMY ADDR1
DUMMY ADDR2
DUMMY ADDR3
DUMMY CITY, ST XXXXX-XXXX

RE: Policy Number: 9999999
Expiration Date: 12/31/2099

NOTICE TO POLICYHOLDERS

Dear Policyholder:

Your Liability renewal policy, including your Umbrella Liability renewal policy, if applicable, will contain an endorsement that excludes coverage for all liability claims resulting from Silica. Please read this endorsement carefully when you receive your renewal policy as this notice is not intended to replace the provisions of your policy.

Please note that your renewal policy will be forthcoming.

We appreciate your business and look forward to serving you through the Continental Western Group of companies. If you have any questions, please contact your agent.

DUMMY AGCY
DUMMY ADDR1
DUMMY ADDR2
DUMMY CITY, ST XXXXX-XXXX
Phone: (999)999-9999
Agency No.: 99999

APPENDIX B: TECHNICHAL INFORMATION

System Component Table

Table B.1: System components

Description	Location	File name	Tool
Master Query	Dev: Datamart Prod: SQL27	tcs_phn_mail.sql	IBM Informix SQL Editor
Print format query	Dev: Datamart Prod: SQL27	tcs_phn_print_format.sql	SQL Server Query Analyzer
Letter Input Spreadsheet	Dev: Datamart Prod: SQL27	Letter.xls	MS Excel
Policy Data table	Dev: Datamart Prod: SQL27	cwg_work..phn_policy_data	MS SQL Server
Letter Data Table	Dev: Datamart Prod: SQL27	cwg_work..phn_letter_data	MS SQL Server
Certificate of Mail	Hyperion Intelligence Server	Phn – Certificate of mail.bqy	Hyperion Intelligence Explorer
Hyperion FTP server file	Hyperion Intelligence Server	Phn – FTP.xml	Microsoft XML Notepad
Policy List	Hyperion Intelligence Server	PHN – UW Letter Policy Lists.bqy	Hyperion Intelligence Explorer

Letter Status Report	Hyperion Intelligence Server	PHN – Letter Status.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0011CC.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0012C2.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0013CC.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0014C2.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0015CC.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0016C2.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0017CC.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0018C2.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0026CC.bqy	Hyperion Intelligence Explorer
CC Vehicle Letter	Hyperion Intelligence Server	PHN – UW0027C2.bqy	Hyperion Intelligence Explorer

Connection to Data Source	Dev: Network Folder 'PHN' Prod: Network Folder 'PHN_prod'	Phn_data_source_connection.dsn	
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6010.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6010.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6113.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_pn6113.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6115.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6115.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6118.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6118.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6119.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6119.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6120.doc	MS Word

Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6120.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6121.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6121.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6128.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6128.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6130.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6130.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6131.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6132.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6133.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6133.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6134.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6134.dqy	MS Query

Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6137.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6137.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6138.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6138.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_PN6141.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_PN6141.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_UW0020.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_UW0020.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_UW0021.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_UW0021.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_uw0022.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_UW0022.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_uw0033.doc	MS Word

Data Source	Prod: Network folder 'PHN_prod'	Phn_UW0033.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_uw0033A.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_UW0033a.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_uw0037.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_UW0037.dqy	MS Query
Policyholder Notice	Dev: Network folder 'PHN'	Phn_UW0040.doc	MS Word
Data Source	Prod: Network folder 'PHN_prod'	Phn_UW0040.dqy	MS Query
MS Word Macro	Dev: Network Folder 'PHN' Prod: Network Folder 'PHN_prod'	PHNMacros.bas	Visual Basic
bat file to print documents	Dev: Network Folder 'PHN' Prod: Network Folder 'PHN_prod'	Filechk.bat	

Letter Spreadsheet

The letter spreadsheet contains data specific to each policyholder notice and renewal letter. You can see a link to the actual spreadsheet in Appendix B, see Table B.1 below for a list of the fields.

Table B.2: Letter Spreadsheet layout– letter.xls

Field	Description
Form Num	The number assigned to the letter by Research and Development, it begins with 'PN' on policyholder notices and 'UW' for renewal letters
Seq	A sequencing number defaulted to 1. Some letters have different effective dates for different states or policy symbols. When you have a letter that has two effective dates we distinguish them by changing the 'Seq' number. For example, PN6119 has four sequences.
Edition	This is the edition date of the filing for the form, it is in mm-yy format.
Description	The free form description of the form
Developer	The BSA team member that developed the form
Type	The type of form, it can be PN for policyholder notice or RL or renewal letter
Pages	The number of pages.
Eff Dt	The date the form take effect on policies
Exp Dt	The date the form expires. It should be the first day the form should not attach. So if the expiration date is 12/1/2006, that means the form will attach on 11/30/2006 but not on 12/1/2006.
Term	The term of the form, this is normally one-year or on-going. On-going means the form will attach indefinitely
Days Prior	The number of days prior to expiration on which the form should print. This is normally 90 days for policyholder notices and 120 days for renewal letters, but any value can be entered.
Copies	The number of copies the computer operations technician must print.
Rtn Env?	A Y/N indicator that indicates of a return envelope should be included

	with the letter. This is normally Y on renewal letters.
Duplex?	A Y/N indicator that tells the computer technician if the form is duplex. Forms that are more than one page are normally Y.
Imaged?	A Y/N indicator that tells the mail staff if this form should be imaged. Normally renewal letters are set to Y and policyholder notices N.
Mail To	This indicates where the letter is mail, 'I' means it is mailed to the Insured and 'A' means it is mailed to the agent.
I Net Copy?	A Y/N indicator that shows if the form should be posted to the CWG Intranet for agent's reference. This is set to 'Y' for policyholder notices and 'N' for renewal letters.

Letter Table

Table B.3 shows the layout of the letter table that is created by the master query. It resides on the cwg_work dataname on both the datamart and SQL27 servers and is named phn_letter_data.

Table B.3: Layout of the letter data table on the warehouse

Field	Description	Type	Length
Form_number	The filed number of the form	Char	12
Sequence	Sequence number	Smallint	2
Form_edition	The edition date of the form in mm-yy format	Char	5
Status	The status of the form 'Active', 'Pending' or 'Cancelled'	Varchar	15
Description	The free form description of the letter	Varchar	100
Start_print_date	The date the form should start printing, calculated by subtracting the days prior from the effective date	Datetime	
End_print_date	The date the letter will stop printing, calculated by subtracting the days prior from the expiration date	Datetime	

Type	Either a policyholder notice or renewal letter, spelled out	Varchar	20
Pages	The number of pages	Smallint	2
Effective date	The effective date of the letter	Datetime	
Expiration date	The expiration date of the letter	Datetime	
Term	The length of time the letter will be attaching	Varchar	15
Days_prior	The number of days prior to expiration the letter should be printed	Int	4
Copies	The number of copies to print	Smallint	2
Return_envelope	Y/N indicator for return envelope	Char	1
Duplex	Y/N indicator for duplexing	Char	1
Imaged	Y/N indicator for imaging the letter	Char	1
Cert_of_mail	Y/N indicator for certificate of mail	Char	1
mail_to	Either Insured or Agent	Varchar	15
Inet_copy	Y/N indicator for posting to the intranet	Char	1

Policy Data Table

Table B.4 shows the layout of the policy data table that is created by the master query and then formatted by another query in SQL Server. It resides on the cwg_work database on both the datamart and SQL27 servers.

Table B.4: Layout of the policy data table on the warehouse

Field	**Description**	**Type**	**Length**
Form_name	The name of the letter from the letter table	Char	16
Form_seq	The sequence number from the letter table	Smallint	2
Mail_date	This will be equal to the data of the last DSS cycle, normally the prior day. On a Monday it will be the previous Friday	Datetime	

Policy_number	Seven digit A+PLUS policy number	Char	7
Policy_mod	The two-digit mod assigned in A+PLUS, it increment by one on every renewal	Char	2
Policy_symbol	The three character policy symbol defined by CWG	Char	3
Tran_proc_date	The date of the most current transaction in A+PLUS	Text	16
Agency_number	CWG assigned ID number for the agency	Int	4
Branch_office	The underwriting service office, could be one of four different locations	Varchar	50
Policy_eff_date	The effective date of the policy	Datetime	
Policy_exp_date	The expiration date of the policy	Datetime	
Rating_company	The legal company, can be Continental Western or Union Insurance Company	Varchar	50
Undw_id	The name of the underwriter	Varchar	30
Insured_name	The name of the policyholder	Varchar	50
Insured_addr1	First address line of the insured	Varchar	50
Insured_addr2	Second address line of the insured	Varchar	50
Insured_addr3	Third address line of the insured	Varchar	50
Insured_city	City where the insured is domiciled	Varchar	30
Insured_st	State the insured is domiciled	Char	2
Insured_zip	Zip code of the insured	Char	10
Agency_name	Name of the agency	Varchar	50
Agency_addr1	First address line of the agency	Varchar	50
Agency_addr2	Second address line of the agency	Varchar	50
Agency_city	City of the agency	Varchar	30
Agency_st	State of the agency	Char	2

Agency_phone	Agency business phone	Char	14
Mail_date_print	Reformatted mailing date used to print on the letter	Text	16
Policy_eff_date_print	Reformatted eff date used to print on the letter	Text	16
Policy_exp_date_print	Reformatted exp date used to print on the letter	Text	16
Driver_name	Name of a driver on a collector car policy	Varchar	40
Veh_year	The year of a vehicle on a collector car policy	Int	4
Veh_make	The make of a vehicle on a collector car policy	Varchar	30
Veh_model	The model of a vehice on a collector car policy	Varchar	50
Veh_serial_number	The serial number of a vehicle on a collector car policy	Varchar	30

SQL Code used in the DSS Query

Below is a sample of the SQL code written in Informix SQL Editor. At the time of this writing not all the policyholder notices and renewal letters had been programmed into the code. The remaining coding will be done by the BSA group as part of their training. The letters that are programmed into this code are PN6010, PN6113, PN6115, PN6118, PN6119, PN6120, UW0011CC, UW0012C2, UW0013CC, UW0014C2, UW0015CC, UW0016C2, UW0017CC, UW0018C2, UW0026CC and UW0027C2.

```
--------------------------------------------------------------------------------
-- Policyholder Notice and Renewal Letters
--------------------------------------------------------------------------------
-- Generating a file with data needed to create the
-- daily policyhodler notices and renewal letters
--------------------------------------------------------------------------------
-- CREATE AND FILL THE letter TABLE
--------------------------------------------------------------------------------
--Step 1 - Create the letter table
--------------------------------------------------------------------------------

CREATE TEMP TABLE letter (
  form_nm                 VARCHAR(20),
  seq                     INTEGER,
  form_ed                 CHAR(5),
  descr                   VARCHAR(100),
  developer               VARCHAR(20),
  type                    CHAR(2),
  pages                   INTEGER,
  eff                     DATE,
  exp                     DATE,
  term                    VARCHAR(20),
  days_prior              INTEGER,
  copies                  INTEGER,
  rtn_env                 CHAR(1),
  duplex                  VARCHAR(10),
  imaged                  CHAR(1),
  com                     CHAR(1),
  mail_to                 CHAR(1),
  inet                    CHAR(1)
  ) WITH NO LOG;

-- Step 2 - Load data from the letter.txt file
--------------------------------------------------------------------------------
-- Loading the letter.txt file as pipe delimited
LOAD FROM '<File path entered here>'
INSERT INTO letter;

-- Step 3 - Set the last run date and process date
--------------------------------------------------------------------------------
-- Here we get the date of the last dss cycle.  We check for copany 40 because that mean CWG
-- We also make sure the load type is spaces becaus if an entry is put in there it means a special
-- cycle was run outside of the daily cycle process
```

```
select     max (date(d_dss_book)) as last_run                -- ** For testing replace d_dss_book with date('<Your last run
date>')
from       s_cd_dss_batch
where      n_berk_co = 40
           and
           c_load_type = ''
into temp last_run_tbl with no log;

-- Get Today's process date
-- We move the value of 'current' into the table process_date_tbl

select distinct      date (current) as process_date          -- ** For testing replace current with <'Your current date'>
from                 s_cd_dss_batch
into temp            process_date_tbl;

-- Step 4 - Calculate the start_print, stop_print and status fields
-------------------------------------------------------------------------------------
-- On the 'letter' table we have three fields that are calculated
-- the start_print, stop_print and status fields.  The default values
-- are '12/31/2099', '12/31/2099' and 'UNKNOWN'

-- Calculate the date to start and stop printing and the status
-- Date to stop printing  = Exp Date - # of days prior
-- Date to start printing = Eff Date = # of day prir
-- Status - Active if today's date is between the eff and exp dates
--        - Expired if today's date is greater than the exp date
--        - Pending if today's date is prior to the eff date

select     *,
           case
             when term = 'on-going' then eff
             else eff - days_prior
           end as start_print,
           case
             when term = 'on-going' then exp
             else exp - days_prior
           end as end_print,
           case
             when (select process_date from process_date_tbl) between (eff - days_prior) and (exp - days_prior)
               then 'ACTIVE'
             when (select process_date from process_date_tbl) >= (exp - days_prior)
               then 'EXPIRED'
             when (select process_date from process_date_tbl) < (eff - days_prior)
               then 'PENDING'
           end as status
from       letter
into temp letter_status with no log;

-- Step 4 - Store the letter_status table in a text file
-------------------------------------------------------------------------------------
-- We need to store letter_status into a text file so it can be mapped
-- back to SQL Server and linked to Hyerion Intelligence
-- Also rearranging the fields

unload to '<File Path entered here>'
select     form_nm, seq, form_ed, status, descr, start_print, end_print,
           case
             when type = 'PN' then 'Policyholder Notice'
             else 'Renewal Letter'
           end as type,
           pages, eff, exp, term, days_prior, copies, rtn_env,
           case
```

```
            when duplex = 'Y' then 'Duplex'
            else 'Simplex'
          end as duplex,
          imaged, com,
          case
            when mail_to = 'I' then 'Insured'
            else 'Agent'
          end as mail_to,
          inet
from      letter_status;

-- Step 5 - Get the range of dates for each letter
--------------------------------------------------------------------------------
SELECT DISTINCT form_nm, seq,
  (SELECT last_run FROM last_run_tbl) + (SELECT a.days_prior FROM letter_status a WHERE a.form_nm = b.form_nm
AND a.seq = b.seq) begin,
  ((SELECT process_date FROM process_date_tbl) - 1) + (SELECT a.days_prior FROM letter_status a WHERE a.form_nm
= b.form_nm
    AND a.seq = b.seq) end
FROM letter_status b
WHERE b.status = 'ACTIVE'
INTO TEMP letter_range_1;

-- Step 6 - We will now find the unique set of begin and end dates for all policies, we'll use
-- this later to generate a table of policies to work from
--------------------------------------------------------------------------------
SELECT          begin, end
FROM            letter_range_1
GROUP BY        begin, end
INTO TEMP       letter_range_2 WITH NO LOG;

-- Step 7 - Now we take the different ranges of dates found in step 7 and use them
-- against the 'pol' table to find all the policies with those expiration dates
--------------------------------------------------------------------------------
SELECT          n_pol, n_pol_mod, MAX (d_tran_proc) d_tran_proc
FROM            pol a INNER JOIN letter_range_2 b ON a.d_pol_exp BETWEEN b.begin AND b.end
GROUP BY        n_pol, n_pol_mod
INTO TEMP       policy_1 WITH NO LOG;

--------------------------------------------------------------------------------
-- GET GENERAL POLICY DATA FROM AGEN
--------------------------------------------------------------------------------
-- We now link 'policy_1' to the agen record to get all the necessary
-- general policy data.

-- Step 1 - Get data from the 'agen' table
--------------------------------------------------------------------------------

SELECT a.n_pol, a.n_pol_mod, sym, a.d_tran_proc, agcy_num, branch_office, eff_dt, exp_dt, pol_status, prim_st,
          renewal_status, corporatereportnum, undw_id, pol_prog, pol_type, bus_ctgry, facility_cd, sbu_cnvrt,
          trsm_rsk_act_ind, misc_ind,
          TRIM(replace(replace(replace(replace(replace(replace (insd_name, '<', ''), '>', ''), '[', ''), ']', ''), '(', ''), ')', ''))
          insd_name,
          TRIM(replace(replace(replace(replace(replace(replace (insd_addr1_or_dba, '<', ''), '>', ''), '[', ''), ']', ''), '(', ''), ')',''))
          insd_addr1_or_dba,
          TRIM(replace(replace(replace(replace(replace(replace (insd_addr2_or_dba, '<', ''), '>', ''), '[', ''), ']', ''), '(', ''), ')',''))
          insd_addr2_or_dba,
          TRIM(insd_addr3) insd_addr3, TRIM (insd_city) insd_city, insd_st,
          CASE
              WHEN insd_zip [7,10] = 0 THEN insd_zip [2,6]
              ELSE insd_zip [2,6] || '-' || insd_zip [7,10]
          END insd_zip
```

```
FROM    policy_1 a INNER JOIN agen b ON
            a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod AND a.d_tran_proc = b.d_tran_proc
INTO TEMP policy_2 WITH NO LOG;

-- Step 2 - Remove any cancellations or non-renewals

DELETE FROM policy_2
WHERE pol_status = 'CN' OR renewal_status = 'NR';

-----------------------------------------------------------------------------------------------------
-- GET AGENCY ADDRESS DATA FROM AGCY
-----------------------------------------------------------------------------------------------------
-- Link the policy_2 table to the agcy table to pull in the agency address and
-- contact information

SELECT a.*,TRIM( t_agcy_nm) t_agcy_nm, TRIM (t_agcy_addr1) t_agcy_addr1, TRIM (t_agcy_addr2) t_agcy_addr2,
            TRIM (t_agcy_city) t_agcy_city, t_agcy_st, TRIM (ROUND (n_agcy_bus_phone) || ' ') n_agcy_bus_phone,
            TRIM (t_agcy_zip || ' ') t_agcy_zip
FROM    policy_2 a INNER JOIN agcy b ON a.agcy_num = b.n_agcy
INTO TEMP policy_3 WITH NO LOG;

-- Now that we've converted the agency zip and phone to a charactor we can
-- convert them to their fomatted or printable versions.

SELECT n_pol, n_pol_mod, sym, d_tran_proc, agcy_num, branch_office, eff_dt, exp_dt, pol_status, prim_st,
        renewal_status, corporatereportnum, undw_id, pol_prog, pol_type, bus_ctgry, facility_cd, sbu_cnvrt,
        trsm_rsk_act_ind, misc_ind, insd_name, insd_addr1_or_dba, insd_addr2_or_dba, insd_addr3, insd_city, insd_st,
        insd_zip, t_agcy_nm, t_agcy_addr1, t_agcy_addr2, t_agcy_city, t_agcy_st,
        CASE
          WHEN t_agcy_zip [6,9] <> '0000' THEN t_agcy_zip [1,5] || '-' || t_agcy_zip [6,9]
          ELSE t_agcy_zip [1,5]
        END t_agcy_zip,
        '(' || n_agcy_bus_phone [1,3] || ') ' || n_agcy_bus_phone [4,6] || '-' || n_agcy_bus_phone [7,10] n_agcy_bus_phone
FROM    policy_3
INTO TEMP policy_mstr WITH NO LOG;

-----------------------------------------------------------------------------------------------------
-- BUILD THE policy_input TABLE
-----------------------------------------------------------------------------------------------------
CREATE TEMP TABLE policy_input (
  form_nm               VARCHAR (12),
  seq                   INT,
  mail_dt               DATE,
  n_pol                 INT,
  n_pol_mod             INT,
  sym                   CHAR (3),
  d_tran_proc           DATETIME YEAR TO FRACTION,
  agcy_num              INT,
  branch_office         INT,
  eff_dt                DATE,
  exp_dt                DATE,
  pol_status            CHAR (2),
  prim_st               CHAR (2),
  renewal_status        CHAR (2),
  corporatereportnum    INT,
  undw_id                       CHAR (3),
  pol_prog              VARCHAR (12),
  pol_type                      INT,
  bus_ctgry             CHAR (3),
  facility_cd           VARCHAR (12),
  sbu_cnvrt             CHAR (1),
  trsm_rsk_act_ind      CHAR (1),
```

```
  misc_ind              VARCHAR (12),
  insd_name             VARCHAR (50),
  insd_addr1_or_dba     VARCHAR (50),
  insd_addr2_or_dba     VARCHAR (50),
  insd_addr3            VARCHAR (50),
  insd_city                     VARCHAR (20),
  insd_st               CHAR (2),
  insd_zip                      CHAR (10),
  t_agcy_nm             VARCHAR (50),
  t_agcy_addr1          VARCHAR (50),
  t_agcy_addr2          VARCHAR (50),
  t_agcy_city           VARCHAR (20),
  t_agcy_st             CHAR (2),
  t_agcy_zip            CHAR (10),
  n_agcy_bus_phone      CHAR (14),
  first_name            VARCHAR (20),
  middle_name           VARCHAR (20),
  last_name             VARCHAR (20),
  veh_yr                INT,
  veh_make              VARCHAR (30),
  veh_mdl                       VARCHAR (30),
  veh_ser               VARCHAR (25)
) WITH NO LOG;

-------------------------------------------------------------------------------------------------------
-- BUILDING POLICY INPUT TABLE FOR EVERYTHING OTHER THAN
-- COLLECTOR CAR
-------------------------------------------------------------------------------------------------------
-- The next series of queries will find policies that apply to each specific
-- form
-------------------------------------------------------------
-- PN6010 - DXS Notice to Policyholders
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------
-- Sequence #1
-- Days Prior = 65
-- Attachment:  Sym = DXS and Umb/Exc = 'U' (cap2)
-- Join the policy_mstr table to cap2 to check the Umb/Exc code

-- Insert policy data into the table
INSERT INTO     policy_input

SELECT
DISTINCT        'PN6010' form_nm, 1 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name,'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl,'vehicle_serial_number' as veh_ser

FROM            policy_mstr a INNER JOIN cap2 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod
WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6010' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6010' and seq = 1)
                AND
                b.d_tran_proc = (select max(d_tran_proc) from cap2 c where b.n_pol = c.n_pol and b.n_pol_mod =
c.n_pol_mod)
                AND
                sym = 'DXS'
                AND
                umbexcspol_typ_ind = 'U';
```

```
-------------------------------------------------------------------------
-- PN6105 - Policyholder Notice - Advisory for personal auto policy
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------------------
-- Sequence #1
-- Days Prior = 60
-- Attachment:  Sym = CC or CC2
-------------------------------------------

-- Join the policy_mstr table to cap2 to check the Umb/Exc code

INSERT INTO     policy_input

SELECT
DISTINCT        'PN6105' form_nm, 1 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6105' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6105' and seq = 1)
                and
                sym in ('CC', 'CC2');

-------------------------------------------------------------------------
-- PN6113 - Policyholder Notice - Illinois Defense Costs
-- Created:  10/21/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #1
-- Days Prior = 90
-- Attachment:  All Illinois policies with liability coverage
-------------------------------------------

-- Since there are so many screens where you can have liability coverage we will identify the policies by
-- using the cvgos table.  We will look for a risk stateof IL and a coverage code of 005 (Auto BI),
-- 008 (Businessowners) or 031 products BI or 37 (Umbrella)

INSERT INTO     policy_input

SELECT
DISTINCT        'PN6113' form_nm, 1 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name,
                'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make, 'vehicle_model' as veh_mdl,
                'vehicle_serial_number' as veh_ser

FROM            policy_mstr a inner join cvgos b on
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
                and
                c_rsk_st = 'IL'
                and
                c_cvg_cvg in ('005', ' 008', '031', '037');
```

```
-------------------------------------------------------------------------
-- PN6115 - CDP notice
-- Created:  11/6/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #1
-- Days Prior = 65
-- Attachment:  Symbol = CDP
-------------------------------------------

INSERT INTO     policy_input

SELECT
DISTINCT        'PN6115' form_nm, 1 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
                and
                sym = 'CDP';

-------------------------------------------------------------------------
-- PN6118 - Contractors' wrap-up amendment of insurance
-- Created:  11/6/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #1
-- Days Prior = 90
-- Attachment:  A contractors class code is entered on the GL coverage screen
-------------------------------------------

-- GL class codes in the 9000 range are contractors/service codes.  According to the renewal spec that was created for this
notice
-- the form was attaching when the class code was on logical table PPGL0191.  Besides the 9000 level codes that table has
the code
-- 51956.  So we will check for the class code to be in the range of 90000 to 99999 or 51956



INSERT INTO     policy_input

SELECT
DISTINCT        'PN6118' form_nm, 1 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a inner join cgld b on
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
```

```
                and
                (
                  (b.class >= 90000 and b.class <= 99999) or b.class = 51956
                );

-------------------------------------------------------------------------
-- PN6119 - Commercial auto forms revision - SD, WY, OR
-- Created:  11/6/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #1
-- Days Prior = 90
-- Attachment:  Symbol <> GO, FDK, CC or CC2, prim_st = SD, WY, OR and commercial auto coverage
-----------------------------------------------

-- To determine if there is commercial auto coverage we will check for the PCCA1-ST field to be
-- SD, OR or WY

INSERT INTO     policy_input

SELECT
DISTINCT        'PN6119' form_nm, 1 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a inner join cca1 b on
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
                and
                sym not in ('GO', 'FDK', 'CC', 'CC2')
                and
                b.st in ('SD', 'WY', 'OR');

-------------------------------------------------------------------------
-- PN6119 - Commercial auto forms revision - MO
-- Created:  11/6/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #2
-- Days Prior = 90
-- Attachment:  Symbol <> GO, FDK, CC or CC2, prim_st = MO and commercial auto coverage
-----------------------------------------------

-- To determine if there is commercial auto coverage we will check for the PCCA1-ST field to be
-- SD, OR or WY

INSERT INTO     policy_input

SELECT
DISTINCT        'PN6119' form_nm, 2 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a inner join cca1 b on
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod
```

```
WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
                and
                sym not in ('GO', 'FDK', 'CC', 'CC2')
                and
                b.st = 'MO';

-------------------------------------------------------------------------
-- PN6119 - Commercial auto forms revision - IA
-- Created:  11/6/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #3
-- Days Prior = 90
-- Attachment:  Symbol <> GO, FDK, CC or CC2, prim_st = IA and commercial auto coverage
-----------------------------------------------

-- To determine if there is commercial auto coverage we will check for the PCCA1-ST field to be
-- SD, OR or WY

INSERT INTO     policy_input

SELECT
DISTINCT        'PN6119' form_nm, 3 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a inner join cca1 b on
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
                and
                sym not in ('GO', 'FDK', 'CC', 'CC2')
                and
                b.st = 'IA';




-------------------------------------------------------------------------
-- PN6119 - Commercial auto forms revision - IL
-- Created:  11/6/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #4
-- Days Prior = 90
-- Attachment:  Symbol <> GO, FDK, CC or CC2, prim_st = IL and commercial auto coverage
-----------------------------------------------

-- To determine if there is commercial auto coverage we will check for the PCCA1-ST field to be
-- SD, OR or WY
```

```
INSERT INTO     policy_input

SELECT
DISTINCT        'PN6119' form_nm, 4 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a inner join cca1 b on
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
                and
                sym not in ('GO', 'FDK', 'CC', 'CC2')
                and
                b.st = 'IL';

-------------------------------------------------------------------------
-- PN6120 - Wrap up CW3254 & Amendment of insured CW3255
-- Created:  11/6/2006 by Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-- Reason for Modification -
-------------------------------------------------------------------------
-- Sequence #1
-- Days Prior = 90
-- Attachment:  Symbol = CU and Package Group = CNP
-----------------------------------------------

-- To determine if there is commercial auto coverage we will check for the PCCA1-ST field to be
-- SD, OR or WY

INSERT INTO     policy_input

SELECT
DISTINCT        'PN6120' form_nm, 1 seq, (select last_run from last_run_tbl) as mail_dt, a.*, 'first_name' as first_name,
                'middle_name' as middle_name, 'last_name' as last_name, 9999 as veh_yr, 'vehicle_make' as veh_make,
                'vehicle_model' as veh_mdl, 'vehicle_serial_number' as veh_ser

FROM            policy_mstr a

WHERE           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'PN6113' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'PN6113' and seq = 1)
                and
                sym = 'CU'
                and
                misc_ind = 'CNP';

-------------------------------------------------------------------------
-- BUILDING THE policy_input and item_input TABLES
-- FOR COLLECTOR CAR.
-------------------------------------------------------------------------
-- Collector car policies use vehicle and driver data on several
-- renewal letters so they go down to the item level with some data
-- This table will hold item level data

CREATE TEMP TABLE item_input (
  form_nm       VARCHAR (12),
  seq           INT,
```

```
  n_pol          INT,
  n_pol_mod      INT,
  c_rsk_rep_type CHAR (4),
  n_rsk_item     INT,
  d_tran_proc    DATETIME YEAR TO FRACTION,
  first_name     VARCHAR (20),
  middle_name    VARCHAR (20),
  last_name      VARCHAR (20),
  veh_yr         INT,
  veh_make       VARCHAR (20),
  veh_mdl              VARCHAR (20),
  veh_ser        VARCHAR (25)
) WITH NO LOG;

--------------------------------------------------------------------------------
-- UW0011CC - CC Accident Prevention Course Discount
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
--------------------------------------------------------------------------------
-- Days Prior = 120
-- Attachment criteria:  Sym = CC and Defensive Driver date
-- is less than the expiration date


insert into      item_input

select distinct  'UW0011CC' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                 first_name, middle_name, last_name, cast (" as varchar (20)) veh_yr, cast (" as varchar (20)) veh_make,
                 cast (" as varchar (20)) veh_mdl, cast (" as varchar (20)) veh_ser

from             policy_mstr a INNER JOIN fpa7 b ON
                   a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where            exp_dt BETWEEN
                   (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                   AND
                   (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                 AND
                 sym = 'CC'
                 AND
                 dd_cr_dt < exp_dt;




--------------------------------------------------------------------------------
-- UW0012C2 - CC2 Accident Prevention Course Discount
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
--------------------------------------------------------------------------------
-- Days Prior = 90
-- Attachment criteria:  Sym = CC2 and Defensive Driver date
-- is less than the expiration date

insert into      item_input

select distinct  'UW0012C2' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                 first_name, middle_name, last_name, cast (" as varchar (20)) veh_yr, cast (" as varchar (20)) veh_make,
                 cast (" as varchar (20)) veh_mdl, cast (" as varchar (20)) veh_ser
```

```
from            policy_mstr a INNER JOIN fpa7 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC2'
                AND
                dd_cr_dt < exp_dt;

--------------------------------------------------------------------------------
-- UW0013CC - CC Limited Use discount
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
--------------------------------------------------------------------------------
-- Days Prior = 120
-- Attachment criteria:  Sym = CC, category = 1, 2 or 3, lmt_use = Y
-- is less than the expiration date

-- Get the item specific data for this form

insert into     item_input

select distinct 'UW0013CC' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast ('' as varchar (20)) first_name, cast ('' as varchar (20)) middle_name, cast ('' as varchar (20))
last_name,
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC'
                AND
                category IN (1, 2, 3)
                AND
                lmt_use = 'Y';



--------------------------------------------------------------------------------
-- UW0014C2 - CC2 Limited Use discount
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
--------------------------------------------------------------------------------
-- Days Prior = 90
-- Attachment criteria:  Sym = CC2, category = 1, 2 or 3, lmt_use = Y
-- is less than the expiration date

insert into     item_input

select distinct 'UW0014C2' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast ('' as varchar (20)) first_name, cast ('' as varchar (20)) middle_name, cast ('' as varchar (20))
last_name,
```

```
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC2'
                AND
                category IN (1, 2, 3)
                AND
                lmt_use = 'Y';

-------------------------------------------------------------------------------
-- UW0015CC - CC Odometer letter for category 9/10 vehicles
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------------------------
-- Days Prior = 120
-- Attachment criteria:  Sym = CC, Category = 9, 10
-- is less than the expiration date

insert into     item_input

select distinct 'UW0015CC' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast ('' as varchar (20)) first_name, cast ('' as varchar (20)) middle_name, cast ('' as varchar (20))
last_name,
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC'
                AND
                category IN (9, 10);




-------------------------------------------------------------------------------
-- UW0016C2 - CC2 Odometer letter for category 9/10 vehicles
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------------------------
-- Days Prior = 90
-- Attachment criteria:  Sym = CC2, Category = 9, 10
-- is less than the expiration date

insert into     item_input

select distinct 'UW0016C2' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast ('' as varchar (20)) first_name, cast ('' as varchar (20)) middle_name, cast ('' as varchar (20))
last_name,
```

```
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC2'
                AND
                category IN (9, 10)
                AND
                bi_occr_lmt > 0;


-------------------------------------------------------------------------------
-- UW0017CC - CC Odometer letter for special use vehicles
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------------------------
-- Days Prior = 120
-- Attachment criteria:  Sym = CC2, Category = 9, 10
-- is less than the expiration date

insert into             item_input

select distinct 'UW0017CC' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast (" as varchar (20)) first_name, cast (" as varchar (20)) middle_name, cast (" as varchar (20))
last_name,
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC'
                AND
                special_use = 'Y'
                AND
                bi_occr_lmt > 0;


-------------------------------------------------------------------------------
-- UW0018C2 - CC2 Odometer letter for special use vehicles
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------------------------
-- Days Prior = 90
-- Attachment criteria:  Sym = CC2, Category = 9, 10
-- is less than the expiration date

insert into     item_input

select distinct 'UW0018C2' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast (" as varchar (20)) first_name, cast (" as varchar (20)) middle_name, cast (" as varchar (20))
last_name,
```

```
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC2'
                AND
                special_use = 'Y'
                AND
                bi_occr_lmt > 0;

-------------------------------------------------------------------------------
-- UW0026CC - CC Restoration Form
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------------------------
-- Days Prior = 120
-- Attachment criteria:  Sym = CC, Restoration = Y
-- is less than the expiration date

insert into     item_input

select distinct 'UW0026CC' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast (" as varchar (20)) first_name, cast (" as varchar (20)) middle_name, cast (" as varchar (20))
last_name,
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC'
                AND
                rtor_ind = 'Y'




-------------------------------------------------------------------------------
-- UW0027C2 - CC2 Restoration Form
-- Created:  10/16/2006 - Tom Shide
-- Modified:  mm/dd/yyyy - XXX
-------------------------------------------------------------------------------
-- Days Prior = 120
-- Attachment criteria:  Sym = CC, Restoration = Y
-- is less than the expiration date

insert into     item_input

select distinct 'UW0027C2' form_nm, 1 seq, a.n_pol, a.n_pol_mod, b.c_rsk_rep_type, b.n_rsk_item, b.d_tran_proc,
                cast (" as varchar (20)) first_name, cast (" as varchar (20)) middle_name, cast (" as varchar (20))
last_name,
```

```
                veh_yr, veh_make, veh_mdl, veh_ser

from            policy_mstr a INNER JOIN ccc1 b ON
                  a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod and a.d_tran_proc = b.d_tran_proc

where           exp_dt BETWEEN
                  (SELECT begin FROM letter_range_1 WHERE form_nm = 'UW0013CC' AND seq = 1)
                  AND
                  (SELECT end FROM letter_range_1 WHERE form_nm = 'UW0013CC' and seq = 1)
                AND
                sym = 'CC2'
                AND
                rtor_ind = 'Y';

--------------------------------------------------------------------------------
-- REMOVE CANCELLED ITEMS FROM item_input
--------------------------------------------------------------------------------
SELECT a.*
FROM    item_input a INNER JOIN hedr b ON
          a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod AND a.d_tran_proc = b.d_tran_proc AND
          a.c_rsk_rep_type = b.c_rsk_rep_type AND a.n_rsk_item = b.n_rsk_item
WHERE  f_rep_canc <> '*'
INTO TEMP item_input_2 WITH NO LOG;

--------------------------------------------------------------------------------
-- JOIN ITEM INPUT TO THE POLICY MASTER TABLE
--------------------------------------------------------------------------------
-- Join the item_input_2 table to the policy_mstr to get all the policy level data
-- for the forms in the item_input_2 table.

select      form_nm, seq, (select last_run from last_run_tbl) as mail_dt, a.*, first_name, middle_name, last_name,
            veh_yr, veh_make, veh_mdl, veh_ser

from        policy_mstr a inner join item_input_2 b on
              a.n_pol = b.n_pol AND a.n_pol_mod = b.n_pol_mod AND a.d_tran_proc = b.d_tran_proc

into temp item_input_3 with no log;

--------------------------------------------------------------------------------
-- ADD THE item_input DATA TO THE policy_input table
--------------------------------------------------------------------------------
-- We already have all policy level forms in the policy_input table, now we will
-- do a union with the policy_input and item_input_3 tables to get our final list of policies

select * from policy_input
union
select * from item_input_3
into temp policy_input_2 with no log;

--------------------------------------------------------------------------------
-- REMOVE THE LEADING '40' FROM THE POLICY NUMBER
--------------------------------------------------------------------------------

UPDATE          policy_input_2
SET             n_pol = n_pol - 400000000
WHERE           n_pol <> 9999999;

--------------------------------------------------------------------------------
-- Remove The dummy values from the item level fields
--------------------------------------------------------------------------------
-- The driver name fields and vehicle fields all have dummy values
-- in them for policy-level forms, we will now replace the dummy values
```

```
-- wtih spaces

update    policy_input_2
set       first_name = ''
where     first_name = 'first_name';

update    policy_input_2
set       middle_name = ''
where     middle_name = 'middle_name';

update    policy_input_2
set       last_name = ''
where     last_name = 'last_name';

update    policy_input_2
set       veh_yr = 0
where     veh_yr = 9999;

update    policy_input_2
set       veh_make = ''
where     veh_make = 'vehicle_make';

update    policy_input_2
set       veh_mdl = ''
where     veh_mdl = 'vehicle_model';

update    policy_input_2
set       veh_ser = ''
where     veh_ser = 'vehicle_serial_number';

--------------------------------------------------------------------------------
-- ENTER A DUMMY RECORD FOR EACH LETTER
--------------------------------------------------------------------------------
-- Here we insert a dummy record for each letter.  This accomplishes two things
-- If there are no letters attaching on a particular day the dummy record will print by iteself
-- so it is easier for the mail room to cross check the letters
-- Secondly it is signal the last letter to be printed was we sort by policy number and the policy
-- number is set to 9999999.

INSERT INTO    policy_input_2
VALUES         ('PN6010', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
               ('12/31/2099'), DATE('12/31/2099'),
               'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
               ADDR1', 'DUMMY ADDR2', 'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY
               AGCY', 'DUMMY ADDR1', 'DUMMY ADDR2',
               'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
               9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO    policy_input_2
VALUES         ('PN6105', 1, (select last_run from last_run_tbl) , 9999999, 99, 'XXX', NULL, 99999, 99, DATE
               ('12/31/2099'), DATE('12/31/2099'), 'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX',
               'DUMMY INSD', 'DUMMY  ADDR1', 'DUMMY ADDR2', 'DUMMY ADDR3', 'DUMMY CITY', 'ST',
               'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1', 'DUMMY ADDR2', 'DUMMY
               CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
               9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO    policy_input_2
VALUES         ('PN6113', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
               ('12/31/2099'), DATE('12/31/2099'),
               'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
               ADDR1', 'DUMMY ADDR2',
```

```
'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
'DUMMY ADDR2',
'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6115', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6118', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6119', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6120', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6121', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6128', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
```

```
'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6130', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6131', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6133', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6134', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('PN6141', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0011CC', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');
```

```
INSERT INTO     policy_input_2
VALUES          ('UW0012C2', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0013CC', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0014C2', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE','LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0015CC', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0016C2', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0017CC', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
```

```
VALUES      ('UW0018C2', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
            ('12/31/2099'), DATE('12/31/2099'),
            'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
            ADDR1', 'DUMMY ADDR2',
            'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
            'DUMMY ADDR2',
            'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
            9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO policy_input_2
VALUES      ('UW0020', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
            ('12/31/2099'), DATE('12/31/2099'),
            'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
            ADDR1', 'DUMMY ADDR2',
            'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
            'DUMMY ADDR2',
            'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
            9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO policy_input_2
VALUES      ('UW0021', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
            ('12/31/2099'), DATE('12/31/2099'),
            'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
            ADDR1', 'DUMMY ADDR2',
            'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
            'DUMMY ADDR2',
            'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
            9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO policy_input_2
VALUES      ('UW0022', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
            ('12/31/2099'), DATE('12/31/2099'),
            'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
            ADDR1', 'DUMMY ADDR2',
            'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
            'DUMMY ADDR2',
            'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
            9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO policy_input_2
VALUES      ('UW0026CC', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
            ('12/31/2099'), DATE('12/31/2099'),
            'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
            ADDR1', 'DUMMY ADDR2',
            'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
            'DUMMY ADDR2',
            'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
            9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO policy_input_2
VALUES      ('UW0027C2', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
            ('12/31/2099'), DATE('12/31/2099'),
            'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
            ADDR1', 'DUMMY ADDR2',
            'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
            'DUMMY ADDR2',
            'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
            9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO policy_input_2
VALUES      ('UW0033', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
            ('12/31/2099'), DATE('12/31/2099'),
```

```
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0033A', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0037', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0040', 1, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

INSERT INTO     policy_input_2
VALUES          ('UW0040', 2, (select last_run from last_run_tbl), 9999999, 99, 'XXX', NULL, 99999, 99, DATE
                ('12/31/2099'), DATE('12/31/2099'),
                'XX', 'XX', 'XX', 9, 'XXX', 'XXX', 99, 'XXX', 'XXX', 'X', 'X', 'XXX', 'DUMMY INSD', 'DUMMY
                ADDR1', 'DUMMY ADDR2',
                'DUMMY ADDR3', 'DUMMY CITY', 'ST', 'XXXXX-XXXX', 'DUMMY AGCY', 'DUMMY ADDR1',
                'DUMMY ADDR2',
                'DUMMY CITY', 'ST', 'XXXXX-XXXX', '(999)999-9999', 'FIRST', 'MIDDLE', 'LAST',
                9999, 'DUMMY MAKE', 'DUMMY MDL', 'DUMMY SER#');

--------------------------------------------------------------------------------
-- UNLOAD THE policy_input file
--------------------------------------------------------------------------------
-- Loading the fields from policy_input into a text file that will be mapped to the table phn_policy_data
-- on the cwg_work database in SQL Server

UNLOAD TO '<File path entered here>'
SELECT form_nm, seq, mail_dt, n_pol, n_pol_mod, sym, d_tran_proc, agcy_num, branch_office, eff_dt, exp_dt,
        corporatereportnum, undw_id, insd_name, insd_addr1_or_dba, insd_addr2_or_dba, insd_addr3,
        insd_city, insd_st, insd_zip, t_agcy_nm, t_agcy_addr1, t_agcy_addr2, t_agcy_city, t_agcy_st, t_agcy_zip,
        n_agcy_bus_phone,
        case
          when middle_name <= ' ' then trim(first_name) || ' ' || trim(last_name)
          else trim(first_name) || ' ' || trim(middle_name) || ' ' || trim(last_name)
        end as driver_name,
```

```
        veh_yr, veh_make, veh_mdl, veh_ser
FROM policy_input_2;
```

SQL Code from Microsoft SQL Server Query Analyzer

After the text files created by the Informix SQL job are complete and the tables phn_policy_data and phn_letter_data are mapped into the cwg_work database there is some formatting necessary. The CWG data warehouse contains a table that links the three character underwriter ID on A+PLUS to there full name. Since the underwriter name appears on some letters we utilized this table to change the underwriter ID from a CHAR (3) field to a VARCHAR (30) field so we could display the full name. Because the onle date type used in SQL Server is the datetime stamp we reformatted the mailing data, policy effective date and policy expiration date to they print in mm/dd/yyyy format on the letters.

Additional formatting done was create descriptive names to replace code numbers for the branch office and rating company. A new mail back field was created, this value is set to 30 days after the current day, and it prints on some letters to serve as a deadline for the underwriter or policyholder to return information to the company.

```
/*******************************************/
/* Policyholder Notice Print Query         */
/*******************************************/

-- This query takes data pull from DSS and stroed in the
-- phn_policy_data table on cwg_work and modifies the
-- underwriter_ID, rating company and branch office fields
-- so they can be printable on a form.


-- Per Kathy Moore-Stark we need to replace the underwriter ID for
-- Sheryl Herronomous with the underwriter ID of Al Stoakes due to
-- Sheryl's retirement.

update    cwg_work.dbo.phn_policy_data
set       undw_id = 'ALS'
where     undw_id = 'SMH'

-- Creating the table phn_print_data by taking the policy data table, replacing
-- the undw_id with the underwriter's first and last name, and also giving a
-- more descriptive name to the Regional Office and the Rating Company (legal
-- entity)

-- Step 1 - Replace the underwriter ID with the underwriter name.
-- We do this by linking to the table reference.dbo.v_uw_codes.
-- We exclude the uw ID's for the agencies because we want to
-- print those in reverse order.

-- Because there are a number of entries on the UW table that have multiple
-- entries for the same uw code we will create a temp table that has just one
-- entry per code.  We will also exclude the inquiry only codes of D and N since
-- they indicate an expired record.
-- We bypass the 'XXX' code because that indicates a dummy record and would not
-- find a match on the table.

select    max (user_fname) as user_fname, max (user_lname) as user_lname, uw_code
into      #phn_uw
from      reference.dbo.v_uw_codes
where     uw_code not in ('SBU', 'FP1', 'XXX')
group by uw_code
order by uw_code

-- Spaecial work around for SBU.  The Small Business Unit Unwd code
-- has four entries in teh v_uw_codes table, one for union, TSI,
-- rocky mountain and central region

update    cwg_work.dbo.phn_policy_data
set       undw_id =
            case
              when branch_office = '70' then 'SBU-UNION'
              when branch_office = '50' then 'SBU-TSI'
              when branch_office = '31' then 'SBU-RM'
              when branch_office = '10' then 'SBU-CENTRAL'
            end
where     undw_id = 'SBU'

-- Special work around for Fire Pak (FP1).  We will code
-- 'FIREPAK' whenever we see an FP1 in case more records are
-- added for fire pack in the future.

update    cwg_work.dbo.phn_policy_data
set       undw_id = 'FIREPAK'
```

```
where     undw_id = 'FP1'

-- Now we use the table created above to fine the Underwriter name
-- for all other recoreds.  These are the non-agency codes

update    cwg_work.dbo.phn_policy_data
set       undw_id = (
                    select  user_fname + ' ' + user_lname
                    from    #phn_uw
                    where cwg_work.dbo.phn_policy_data.undw_id =
                                #phn_uw.uw_code

                  )
where     undw_id not in ('BL1', 'CU1', 'DJ1', 'DJ2', 'DJ3', 'DJ4', 'HU1',
            'KI1', 'RR1', 'TN1', 'WD', 'FIREPAK', 'XXX')
          and
          undw_id not like 'SBU%'

-- Step 2 - Load the uw name for the agencies.  For the agencies we will
-- print last name / first name because of how the names are stored in
-- the reference table


update    cwg_work.dbo.phn_policy_data
set       undw_id = (
                    select  user_lname + ' ' + user_fname
                    from    reference.dbo.v_uw_codes
                    where cwg_work.dbo.phn_policy_data.undw_id =
                                reference.dbo.v_uw_codes.uw_code
                  )
where     undw_id  in ('BL1', 'CU1', 'DJ1', 'DJ2', 'DJ3', 'DJ4', 'HU1',
            'KI1', 'RR1', 'TN1', 'WD')

-- Step 3 - Format the Rating Company
-- Assigning the official legal entity names to the rating copany

update    cwg_work.dbo.phn_policy_data
set       rating_company =
            case
              when rating_company = '4' then
                'Continental Western Insurance Company'
              when rating_company = '6' then
             'Union Insurance Company'
              when rating_company = '9' then
                 'DUMMY RATING COMPANY'
              else 'Unknown'
            end


-- Step 4 - Updating the branch office field
-- For the branch office we will show either Rocky Mountain,
-- Union, Tri-State or Central in place of the numeric indicator

update    cwg_work.dbo.phn_policy_data
set       branch_office =
            case
              when branch_office = '10' then
                'Central Regional Office'
              when branch_office = '31' then
            'Rocky Mountain Regional Office'
              when branch_office = '50' then
                    'Tri-State Regional Office'
```

```
            when branch_office = '70' then
                  'Union Regional Office'
            when branch_office = '99' then
                  'DUMMY BRANCH'
            else 'Unknown'
           end

-- Step 4 - Reformat the mail date, eff and exp dates so they can print on the
--documents

update    cwg_work.dbo.phn_policy_data
set       mail_date_print =
            ltrim(str(month(mail_date))) + '/' + ltrim(str(day (mail_date))) +
            '/' + ltrim(str(year(mail_date))),
          policy_eff_date_print =
            ltrim(str(month(policy_eff_date))) + '/' + ltrim(str(day (policy_eff_date))) +
            '/' + ltrim(str(year(policy_eff_date))),
          policy_exp_date_print = ltrim(str(month(policy_exp_date))) + '/' + ltrim(str(day (policy_exp_date))) +
            '/' + ltrim(str(year(policy_exp_date)))

-- Step 5 - Calculate the mail back date
-- On the phn_item_data table we have a mail back date field
-- This date is 30 days after the mailing date.  For dummy records
-- we will fill with 12/31/2099

update    cwg_work.dbo.phn_item_data
set       mail_back_date = (
            case
              when policy_number = 9999999 then '12/31/2099'
              else getdate() + 30
            end
          )

update    cwg_work.dbo.phn_item_data
set       mail_back_date_print =
            ltrim(str(month(mail_back_date))) + '/' + ltrim(str(day (mail_back_date))) +
            '/' + ltrim(str(year(mail_back_date)))
```

Visual Basic Macro for Word

Below is code that creates a macro called PHNMacros. It opens each policyholder notice and renewal letter created in MS Word, connects to the data source and sends to the printer. This was in use on the previous system and was adapted to the new system for the different file names and network directories.

```
Sub PHN()
'
' PHN - Policyholder Notice / Renewal Letter Generation
' Macro recorded 11/02/2006 by TCS
'
' 8 1/2 X 11 Duplex on Plain Paper

   Application.Run MacroName:="phn_PN6105"
   Application.Run MacroName:="phn_PN6119"
   Application.Run MacroName:="phn_PN6121"


' 8 1/2 x 11 Simplex Plain Paper
   Application.Run MacroName:="phn_PN6010"
   Application.Run MacroName:="phn_PN6113"
   Application.Run MacroName:="phn_PN6115"
   Application.Run MacroName:="phn_PN6118"
   Application.Run MacroName:="phn_PN6120"
   Application.Run MacroName:="phn_PN6128"
   Application.Run MacroName:="phn_PN6130"
   Application.Run MacroName:="phn_PN6133"
   Application.Run MacroName:="phn_PN6134"
   Application.Run MacroName:="phn_PN6137"
   Application.Run MacroName:="phn_PN6138"
   Application.Run MacroName:="phn_PN6141"
   Application.Run MacroName:="phn_UW0020"
   Application.Run MacroName:="phn_UW0021"
   Application.Run MacroName:="phn_UW0022"
   Application.Run MacroName:="phn_UW0033"
   Application.Run MacroName:="phn_UW0033A"
   Application.Run MacroName:="phn_UW0037"
   Application.Run MacroName:="phn_UW0040"

End Sub

Sub phn_PN6010()
'
' phn_PN6010 Macro
' Macro recorded 11/02/2006 by TCS
'
   ChangeFileOpenDirectory "<File path entered here>"
   Documents.Open FileName:="phn_PN60100803.doc", ConfirmConversions:=False, _
      ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
      PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
      WritePasswordTemplate:="", Format:=wdOpenFormatAuto
   With ActiveDocument.MailMerge
      .Destination = wdSendToPrinter
```

```
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub


Sub phn_PN6105()
'
' phn_PN6105 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
  Documents.Open FileName:="phn_PN61050106.doc", ConfirmConversions:=False, _
    ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
    PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
    WritePasswordTemplate:="", Format:=wdOpenFormatAuto
  With ActiveDocument.MailMerge
    .Destination = wdSendToPrinter
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub


Sub phn_PN6113()
'
' phn_PN6113 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
  Documents.Open FileName:="phn_PN61130506.doc", ConfirmConversions:=False, _
    ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
    PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
    WritePasswordTemplate:="", Format:=wdOpenFormatAuto
  With ActiveDocument.MailMerge
    .Destination = wdSendToPrinter
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6115()
'
' phn_PN6115 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
```

```
    Documents.Open FileName:="phn_PN61150106.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6118()
'
' phn_PN6118 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61180606.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6119()
'
' phn_PN6119 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61191006.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6120()
'
```

```
' phn_PN6120 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61200606.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6121()
'
' phn_PN6121 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61210606.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6128()
'
' phn_PN6128 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61280506.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
```

```
End Sub

Sub phn_PN6130()
'
' phn_PN6130 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61300606.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6131()
'
' phn_PN6131 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61310706.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
        End With
        .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6133()
'
' phn_PN6133 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_PN61330706.doc", ConfirmConversions:=False, _
        ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
        PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
        WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
        .Destination = wdSendToPrinter
        .SuppressBlankLines = True
        With .DataSource
            .FirstRecord = wdDefaultFirstRecord
            .LastRecord = wdDefaultLastRecord
```

```
      End With
      .Execute Pause:=False
   End With
   ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6134()
'
' phn_PN6134 Macro
' Macro recorded 11/02/2006 by TCS
'
   ChangeFileOpenDirectory "<File path entered here>"
   Documents.Open FileName:="phn_PN61340706.doc", ConfirmConversions:=False, _
      ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
      PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
      WritePasswordTemplate:="", Format:=wdOpenFormatAuto
   With ActiveDocument.MailMerge
      .Destination = wdSendToPrinter
      .SuppressBlankLines = True
      With .DataSource
         .FirstRecord = wdDefaultFirstRecord
         .LastRecord = wdDefaultLastRecord
      End With
      .Execute Pause:=False
   End With
   ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6137()
'
' phn_PN6137 Macro
' Macro recorded 11/02/2006 by TCS
'
   ChangeFileOpenDirectory "<File path entered here>"
   Documents.Open FileName:="phn_PN61370706.doc", ConfirmConversions:=False, _
      ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
      PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
      WritePasswordTemplate:="", Format:=wdOpenFormatAuto
   With ActiveDocument.MailMerge
      .Destination = wdSendToPrinter
      .SuppressBlankLines = True
      With .DataSource
         .FirstRecord = wdDefaultFirstRecord
         .LastRecord = wdDefaultLastRecord
      End With
      .Execute Pause:=False
   End With
   ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6138()
'
' phn_PN6138 Macro
' Macro recorded 11/02/2006 by TCS
'
   ChangeFileOpenDirectory "<File path entered here>"
   Documents.Open FileName:="phn_PN61380706.doc", ConfirmConversions:=False, _
      ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
      PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
      WritePasswordTemplate:="", Format:=wdOpenFormatAuto
   With ActiveDocument.MailMerge
      .Destination = wdSendToPrinter
```

```
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_PN6141()
'
' phn_PN6141 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
  Documents.Open FileName:="phn_PN61410706.doc", ConfirmConversions:=False, _
    ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
    PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
    WritePasswordTemplate:="", Format:=wdOpenFormatAuto
  With ActiveDocument.MailMerge
    .Destination = wdSendToPrinter
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_UW0020()
'
' phn_UW0020 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
  Documents.Open FileName:="phn_UW00200703.doc", ConfirmConversions:=False, _
    ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
    PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
    WritePasswordTemplate:="", Format:=wdOpenFormatAuto
  With ActiveDocument.MailMerge
    .Destination = wdSendToPrinter
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_UW0021()
'
' phn_UW0021 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
  Documents.Open FileName:="phn_UW00210703.doc", ConfirmConversions:=False, _
    ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
```

```
    PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
    WritePasswordTemplate:="", Format:=wdOpenFormatAuto
  With ActiveDocument.MailMerge
    .Destination = wdSendToPrinter
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_UW0022()
'
' phn_UW0022 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
  Documents.Open FileName:="phn_UW00220703.doc", ConfirmConversions:=False, _
    ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
    PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
    WritePasswordTemplate:="", Format:=wdOpenFormatAuto
  With ActiveDocument.MailMerge
    .Destination = wdSendToPrinter
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub

Sub phn_UW0033()
'
' phn_UW0033 Macro
' Macro recorded 11/02/2006 by TCS
'
  ChangeFileOpenDirectory "<File path entered here>"
  Documents.Open FileName:="phn_UW00330605.doc", ConfirmConversions:=False, _
    ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
    PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
    WritePasswordTemplate:="", Format:=wdOpenFormatAuto
  With ActiveDocument.MailMerge
    .Destination = wdSendToPrinter
    .SuppressBlankLines = True
    With .DataSource
      .FirstRecord = wdDefaultFirstRecord
      .LastRecord = wdDefaultLastRecord
    End With
    .Execute Pause:=False
  End With
  ActiveDocument.Close (wdDoNotSaveChanges)
End Sub
Sub phn_UW0033A()
'
' phn_UW0033A Macro
' Macro recorded 11/02/2006 by TCS
'
```

```
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_UW00330605A.doc", ConfirmConversions:=False, _
      ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
      PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
      WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
      .Destination = wdSendToPrinter
      .SuppressBlankLines = True
      With .DataSource
        .FirstRecord = wdDefaultFirstRecord
        .LastRecord = wdDefaultLastRecord
      End With
      .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub
Sub phn_UW0037()
'
' phn_UW0037 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_UW00371005.doc", ConfirmConversions:=False, _
      ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
      PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
      WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
      .Destination = wdSendToPrinter
      .SuppressBlankLines = True
      With .DataSource
        .FirstRecord = wdDefaultFirstRecord
        .LastRecord = wdDefaultLastRecord
      End With
      .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub
Sub phn_UW0040()
'
' phn_UW0040 Macro
' Macro recorded 11/02/2006 by TCS
'
    ChangeFileOpenDirectory "<File path entered here>"
    Documents.Open FileName:="phn_UW00400206.doc", ConfirmConversions:=False, _
      ReadOnly:=False, AddToRecentFiles:=False, PasswordDocument:="", _
      PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
      WritePasswordTemplate:="", Format:=wdOpenFormatAuto
    With ActiveDocument.MailMerge
      .Destination = wdSendToPrinter
      .SuppressBlankLines = True
      With .DataSource
        .FirstRecord = wdDefaultFirstRecord
        .LastRecord = wdDefaultLastRecord
      End With
      .Execute Pause:=False
    End With
    ActiveDocument.Close (wdDoNotSaveChanges)
End Sub
```