(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 December 2010 (02.12.2010)

PCT

(10) International Publication Number
WO 2010/138906 A1

(51) International Patent Classification:
G06F 17/00 (2006.01)

(21) International Application Number:
PCT/US2010/036717

(22) International Filing Date:
28 May 2010 (28.05.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/182,085 28 May 2009 (28.05.2009) US

(71) Applicant *(for all designated States except US)*: **THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEWYORK** [US/US]; 116 Street And Broadway, New York, NY 10027 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ANDERSON, Roger, N.** [US/US]; 106 Morningside Drive, New York, NY 10027 (US). **CHOW, Maggie** [US/US]; 4 Irving Place, New York, NY 10004 (US). **BOULANGER, Albert** [US/US]; 560 Riverside Drive, New York, NY 10027 (US).

(74) Agent: **CHUMNEY, Jason, C.**; BAKER BOTTS LLP, 30 Rockefeller Plaza, New York, NY 10112-4498 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: CAPITAL ASSET PLANNING SYSTEM

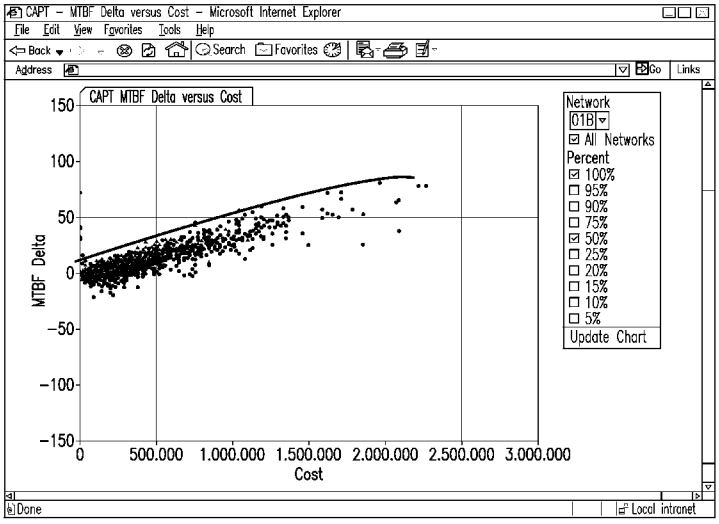


FIG.17

(57) **Abstract**: A capital asset planning system for selecting assets for improvement within an infrastructure that includes data sources descriptive of the infrastructure, databases coupled to the one or more data sources, to compile the one or more data sources, processors coupled to and having respective communication interfaces to receive data from the databases The processor includes a predictor to generate a first metric of estimated infrastructure effectiveness based upon a current status of the infrastructure, a second metric of estimated infrastructure effectiveness based upon a user-selected, proposed changed configuration of the infrastructure, and a net metric of infrastructure effectiveness based upon said first metric and said second metric The system also includes a display, coupled to have the processors, for visually presenting the net metric of infrastructure effectiveness, in which the assets for improvement are selected based upon the net metric of infrastructure effectiveness

WO 2010/138906 A1

BAKER BOTTS L.L.P.

30 ROCKEFELLER PLAZA

NEW YORK, NEW YORK 10112

CAPITAL ASSET PLANNING SYSTEM

of which the following is a

SPECIFICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority from U.S. Provisional Application Serial No. 61/182,085 filed on May 28, 2009, the entirety of the disclosures of which is explicitly incorporated by reference herein.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by any one of the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

FIELD

[0003] The disclosed subject matter relates to techniques for prioritizing use of capital assets for infrastructure improvements using a capital asset planning tool (CAPT) system.

BACKGROUND

[0004] Infrastructures, particularly mature infrastructures, are in constant need of improvement and upgrade. Furthermore, regulatory and environmental concerns dictate the removal of older equipment in favor of newer, more-efficient equipment. Due to commercial realities, there is a limited budget that is available for such improvements to infrastructure. There remains a need to logically and quantitatively determine which assets within a complex infrastructure to select for improvement in order to maximize the benefit obtained therefrom.

SUMMARY

[0005] The present application provides methods and systems for prioritizing use of capital assets for infrastructure improvements.

[0006] One aspect of the present application provides a capital asset planning system for selecting assets for improvement within an infrastructure that includes one or more data sources descriptive of the infrastructure, one or more databases, coupled to the one or more data sources, to compile the one or more data sources, one or more processors, each coupled to and having respective communication interfaces to receive data from the one or more databases. The processor includes a predictor to generate a first metric of estimated infrastructure effectiveness based, at least in part, on a current status of the infrastructure, a second metric of estimated infrastructure effectiveness based, at least in part, on a user-selected, proposed changed configuration of the infrastructure, and a net metric of infrastructure effectiveness based, at least in part, on said first metric and said second metric. The system also includes a display, coupled to have the one or more processors, for visually presenting the net metric of infrastructure

effectiveness, in which the assets for improvement are selected based, at least in part, on the net metric of infrastructure effectiveness.

[0007] Another aspect of the present application also provides a method for selecting assets for improvement within an infrastructure that includes accessing one or more data sources descriptive of the infrastructure, compiling the one or more data sources into one or more databases, generating a first metric of estimated infrastructure effectiveness based, at least in part, on a current status of the infrastructure, generating a second metric of estimated infrastructure effectiveness based, at least in part, on a user-selected, proposed changed configuration of the infrastructure, generating a net metric of infrastructure effectiveness based, at least in part, on said first metric and said second metric, and displaying the net metric of infrastructure effectiveness, in which the assets for improvement are selected based, at least in part, on the net metric of infrastructure effectiveness.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Further objects, features and advantages of the disclosed subject matter will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the disclosed subject matter, in which:

[0009] FIG. 1 depicts an exemplary web application layer (GUI) according to one CAPT embodiment of the presently disclosed subject matter.

[0010] FIG. 2 depicts CAPT system components and their relationship according to an exemplary embodiment of the present application.

[0011] FIG. 3 depicts the Feeder change evaluation ranking system that re-ranks feeders and updates Support Vector Machine scores to evaluate What-If impact from Capital or Operational changes such as those described in Tiers 1-4 according to one embodiment of the presently disclosed subject matter.

[0012] FIG. 4 depicts the goodness-of-fit validation method for establishing that the Predicted OA Feeder Failure Rate (1/MTBF) that is computed from a linear regression of the Machine Learning SVM scores and rankings of FIG. 4 represents a statistically valid representation of the Actual observed OA Feeder Failure Rate (1/MTBF).

[0013] FIG. 5 depicts the Graphical User Interface of a representative embodiment that presents options for Load Relief (Tier 1), PILC Section Replacement (Tier 2), Reliability Replacement (Tier 3) and LPW Selection (Tier 4). All critical component compositions of the feeder are displayed along with their baseline count and user selected new count.

[0014] FIG. 6 depicts the CAPT rollup view that lists the minimal costs per largest benefits attainable from replacement of a selectable percentage of bad components in each Tier.

[0015] FIG. 7 depicts a crossplot of % Delta MTBF (vertical y-axis) vs estimated costs (horizontal x-axis) used to evaluate which feeders in a given network or circuit provide the maximum benefit for optimal cost.

[0016] FIG. 8 depicts a Geographic Information System display (Google Earth) of the Machine Learning Ranking of each component section of each feeder in a representative network. The higher the "window pane", the more susceptible each section is to impending failure.

[0017] FIG. 9 depicts a selected feeder from the FIG. 8 network with several candidate runs of PILC cable sections, the replacement of which will lower the risk (and lower the window pane height in the visual).

[0018] FIG. 10 depicts a zoom into the tolopogical layout of the feeder (upper left, white box) that shows component by component connectivity between feeder sections and the manholes (M) containing the joints connecting the sections. T and V are transformers.

[0019] FIG. 11 depicts a screen shot of an exemplary embodiment of the CAPT tool evaluating Tier 1 Load Relief options for a representative feeder.

[0020] FIG. 12 depicts a drilldown into the Tier 1 Load Relief section selection process that displays the runs of overloaded cable sections that are to be selected from for possible replacement.

[0021] FIG. 13 depicts the predicted change in MTBF and Machine Learning susceptibility rank for the feeder if a recommended 55 most at-risk PILC sections from the Tier 2 PILC Replacement evaluation are made.

[0022] FIG. 14 depicts the replacement of 50% of the Stop Joints (total of 11) in a representative feeder during Tier 3 evaluation.

[0023] FIG. 15 depicts a screen capture of the CAPT tool Tier 4 evaluation of Load Pocket Weight repairs that are possible for a representative feeder.

[0024] FIG. 16 depicts the predicted MTBF improvement of 8 days if the changes selected in FIG. 15 are made, i.e., closing 8 open switches and repairing 4 Open Mains.

[0025] FIG. 17 depicts the Efficient Frontier of cost vs benefit (in MTBF Delta Days) for 50% and 100% PILC cable replacement options from Tier 2 evaluation.

[0026] While the disclosed subject matter will now be described in detail with reference to the figures, it is done so in connection with illustrative, non-limiting embodiments.

DETAILED DESCRIPTION

[0027] One aspect of the present application provides a capital asset planning system for selecting assets for improvement within an infrastructure that includes one or more data sources descriptive of the infrastructure, one or more databases, coupled to the one or more data sources, to compile the one or more data sources, one or more processors, each coupled to and having respective communication interfaces to receive data from the one or more databases. The processor includes a predictor to generate a first metric of estimated infrastructure effectiveness based, at least in part, on a current status of the infrastructure, a second metric of estimated infrastructure effectiveness based, at least in part, on a user-selected, proposed changed configuration of the infrastructure, and a net metric of infrastructure effectiveness based, at least in part, on said first metric and said second metric. The system also includes a display, coupled to have the one or more processors, for visually presenting the net metric of infrastructure effectiveness, in which the assets for improvement are selected based, at least in part, on the net metric of infrastructure effectiveness.

[0028] In one embodiment, the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness are both based, at least in part, on an estimated length of time that the infrastructure produces a commodity. For example, the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness can be based, at least in part, on the estimated mean time between failure of one or more components of the electrical grid, such as the estimated mean time between failure of an electrical feeder. In other embodiments, the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness is estimated using a feeder index, the index based, at least in part, on a predicted likelihood of feeder failure. The feeder can be updated in view of changed conditions using machine learning.

[0029] In various embodiments, the infrastructure to which the capital asset planning system and methods of the presently disclosed subject matter can be applied to is without limitation. In one embodiment, the infrastructure is selected from the group consisting of a chemical processing operation, a petroleum refining operation, a product manufacturing operation, a telecommunication grid, a transportation infrastructure, a gas network, a commodity pipeline network, and a water treatment network.

[0030] In one embodiment, the infrastructure is an electrical grid. Data sources descriptive of the electrical grid include one or more of data representative of at least one of electrical feeder data, electrical cable data, electrical joint data, electrical transformer data, electrical outage data, electrical test pass or fail data, electrical load data, and past capital improvement cost data.

[0031] In certain embodiments of the presently disclosed subject matter, the predictor further generates a cost metric based, at least in part, on the cost of the user-selected, proposed change of the infrastructure. For example, the cost metric can be generated based, at least in part, on a user-specified cost of one or more specific actions encompassed by the user-selected, proposed change of the infrastructure. Alternatively, the cost metric is based, at least in part, on past capital improvement cost data. In one embodiment, the predictor further comprises a prioritizer to determine the user-selected, proposed changed configuration of the infrastructure that provides the maximum value based, at least in part, on the net metric of infrastructure effectiveness and the cost metric.

[0032] One aspect of the present application also provides a method for selecting assets for improvement within an infrastructure that includes accessing one or more data sources descriptive of the infrastructure, compiling the one or more data sources into one or more databases, generating a first metric of estimated infrastructure effectiveness based, at least in part, on a current status of the infrastructure, generating a second metric of estimated infrastructure effectiveness based, at least in part, on a user-selected, proposed changed configuration of the infrastructure, generating a net metric of infrastructure effectiveness based, at least in part, on said first metric and said second metric, and displaying the net metric of infrastructure effectiveness, in which the assets for improvement are selected based, at least in part, on the net metric of infrastructure effectiveness.

[0033] In one embodiment, the infrastructure is an electrical grid, and the first metric, the second metric and the net metric is based, at least in part, on the estimated mean time between failure of

an electrical feeder within the electrical grid. Additionally, the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness can be estimated using a feeder index, the index based, at least in part, on a predicted likelihood of feeder failure. In one embodiment, the method further includes generating a cost metric based on the cost of the user-selected, proposed change of the infrastructure, such that cost benefit analysis can be performed, and the infrastructure receives a maximum bang for the buck.

[0034] In one embodiment, the present application provides methods and systems for quantitatively predicting an effectiveness of a proposed capital improvement project based on establishing the changes in attributes predicting feeder performance based on changes in assets from the improvement project and using either SVM raking then regression of ranks to MTBF or SVM Regression to estimate MTBF directly. In one embodiment, the benefit is the change in MTBF from the project and the cost is the cost of the project. The system allows comparison of cost vs. benefits amongst alternative projects given budget constraints providing an enhanced portfolio selection of projects.

[0035] As used herein, a "user-selected, proposed changed configuration of the infrastructure," or more generally, a "capital improvement project" or an "improvement" to the infrastructure refers to any change in an infrastructure, including, but not limited to, the replacement of any one or more components of the infrastructure for any purpose, such as to improve the output of the infrastructure, for purposes of preventative maintenance, and/or for regulatory compliance.

[0036] The presently disclosed subject matter can be further described with reference to the following non-limiting embodiments. In one particular embodiment, as shown in Figure 1, the

CAPT system is used to analyze how to improve an allocation of a fixed budget devoted to the modernization of cable (i.e., the replacement of older PILC cable (paper insulated lead cable insulation) to newer XLP cross-linked polyethylene insulation) and EPR (ethylene propylene rubber insulation) cable in a particular electrical distribution feeder (4X52) within an electrical grid.

[0037] The CAPT web application can be linked to database(s) that provides baseline feeder attributes (10) regarding, for example, the type of cable currently in place (e.g., paper insulated cable, etc.), connecting joints currently in place, and details regarding the transformers used in a particular feeder to convert power from distribution voltages to consumer voltages. The user can define a variety of proposed upgrades to these attributes (20), based on, for example, a) the percentage of PILC cable that is replaced, i.e., swapped out with more modern cable (e.g., XLP or EPR cable), b) load relief for overstressed cable and/or c) general system reliability and/or preventative maintenance. The CAPT web application layer can also be provided with historical data regarding observed failure and performance for the particular component mix of each individual feeder within a network, circuit, and electrical grid, such as actual MTBF (30), actual OA (Open Automatic feeder failures) counts (40), and actual replacement costs based on the actual performance history of each specific feeder of a network, circuit and electrical grid. Baseline feeder rankings (50), baseline estimated MTBF (60) and estimated FOT (failure on test) prediction data (70) can also be provided, based on the SVM model (either ranking or regression) and optionally, a regression (if SVM ranking is used) prediction that accounts for the particular feeder performance (here, 4X51) before any improvements have been made.

[0038] After the user has inputted details regarding the proposed changes, the CAPT system outputs summary data regarding the upgrade work proposed (80), including the estimated cost per section (90) and total cost (100) to perform the proposed work. These cost estimates can be determined based on the historical data for cable replacement efforts that is stored in the database for statistically similar feeders. Alternatively, the cost data can be a user-input field to account for particular features of the particular feeder, or simply a system wide estimate based on for example, previous work or public service commission data.

[0039] The CAPT web application, in communication with one or more processors, also outputs new feeder rankings (110), new estimated MTBF (120), and new estimated FOT (130) in view of the upgrades proposed. From these metrics, one can ascertain the statistically estimated (e.g., a SVM model based estimated) benefit of the proposed upgrades in view of the costs, i.e. a cost benefit analysis that determines the best overall "bang for the buck." For the overall portfolio of replacement and preventive maintenance work being planned at each increment in time.

[0040] This output can be stored and analyzed vs., for example, predicted MTBF improvements of other proposed changes to determine how to best allocate a fixed budget that is devoted to this type of capital asset and operations and maintenance work. The CAPT system, and the CAPT web application specifically, offers the ability to graphically output this data point along with other data points derived from other proposed changes to other feeders or other possible replacement strategies and levels of replacement in the same feeder in graphical form to assist in budgeting (see Figure 7), as for example in a Geographic Information System such as Google Earth or Microsoft Bling.

[0041] With reference to Figure 2, an overview of an exemplary CAPT system architecture is provided. Source data (200) from the infrastructure (e.g., an electrical grid) is tapped, which includes, for example, data regarding the network topology, reliability (e.g., a network reliability index or "Jeopardy" database of most at risk networks and electrical feeders), feeder history and manufacturing statistics, outage and previous failure data, information from past capital improvement project costs and benefits, past electric load data, electrical cable component data by section and joint, transformer data, secondary low voltage electrical cable (main) data, GIS and mapping tool outputs such as Contingency Analysis and performance variances, systems and Hi-Pot ranking data (data used to derive a score for deciding which feeders to perform high potential tests in which voltage for a feeder that is over its normal operating voltage is applied to the feeder to stress the feeder and test its reliability).

[0042] This data is sent to databases (210) that compile all source data SVM modeling inputs and outputs in order to organize it into a form conducive for further analysis and data mining. Non-limiting databases for use in accordance with the presently disclosed subject matter include Vision Snapshot and Jeopardy database, which includes cable, joint, transformer data. The databases also include "ODDS" data, which is a machine learning system that uses SVMs (support vector machines) to predict a ranking (or MTBF using SVM regression) from most likely to least likely feeders susceptible to impending failure. See, e.g., International Published PCT Application No. WO 09/117742, hereby incorporated by reference in its entirety.

[0043] The databases are also in communication with other machine learning subsystems (220), that process data from the database and determines, for example, cable and joint rankings similar

to the feeder rankings discussed above. The machine learning subsystems automatically rerank potential capital asset, operations, and maintenance work (230) according to the optimal mix of predicted benefits at the minimal costs as more data is continually received over time and changes are enacted on the real system based on the predictions of CAPT outputs (see Figure 3).

[0044] In this example, the databases are also in communication with geospatial databases (240), including mapping data, such as Google Earth mapping data to create a GIS visualization subsystem (250). The geospatial database also receives data from a cable subsystem (270), that processes additional details regarding runs (a contiguous series) of the same type of cable.

[0045] The processed data from the database and visualization data from the visualization subsystem are communicated to the CAPT web application layer (GUI) (260), an example of which is provided in Figure 1. The web application layer can output, among other things, ranking visualizations, MTBF graphing systems, network/borough MTBF vs. cost display grids and feeder cost benefit analysis details, including both tables and charts. Representative code for CAPT web application layer according to one specific embodiment of the present application is set forth in Appendix A. This program is in communication with the representative code set forth in Appendix C, which derives the new feeder attributes from the asset changes on the feeder.

[0046] Further details regarding feeder re-ranking data are provided in Figure 3. The CAPT web application (300) is in communication with relational databases (310), examples of which are discussed throughout the application, and borough and network feeder parameters, strategies and cable, joint and transformer change options (PILC replacement percentages) (320). As shown in

the Figure, the relational database is in communication with one or more of machine learning processors. These processors include an attribute change computation processor (330), which sends to the Link Web service the attributes for a changed feeder and receives updated rankings and updates the database with the new rankings. In this particular, non-limiting embodiment, the attribute change computation processor is written in .net/C#. For each feeder, attribute changes can be computed based on, among other things, changed sections selected by the CAPT interface, and the data is sent as an input string to the linked web service (340).

[0047] The linked web service can be a stand alone, command-line based web service that receives a string of feeder attributes as parameters and returns updated rankings and MTBF predictions. In this particular non-limiting embodiment, the link web service is written in Python programming language and invokes the Python-based Machine Learning re-ranking process.

[0048] The link web service can also be in communication with a re-ranking processor (350), also written in Python to return updated scores and rankings to the link web service. The re-ranking processor runs a machine learning algorithm such as ODDS to re-rank feeders based on MTBF improvement based on SVM processing of component attributes. Representative code for the re-ranking processor is set forth in Appendix B. This program is in communication with the representative code set forth in Appendix C, which derives the new feeder attributes from the asset changes on the feeder. The re-ranking processor can be in communication with the SVM Light processor (360). The SVM Light processor is a 3rd party open source program that receives an input of a dictionary of attributes mapped to their new values and outputs a string of comma-separated numbers representing the new ranking and/or MTBF.

[0049] As noted previously in connection with Figure 1, embodiments of the presently disclosed subject maintain an actual baseline performance metric, and a predicted baseline performance metric. Comparison of the predicted performance metric with a predicted baseline performance metric can be preferable to comparing the predicted performance metric to the actually observed baseline performance metric, since it is best to be consistent in using the model to obtain both a baseline and projected changes to system performance. Upon receiving proposed capital, operations, and maintenance improvement funds, the actual vs predicted performance metrics are all updated. Providing the actually observed performance metrics therefore provides the "reality check," in that by comparing the observed baseline performance metric to the predicted baseline performance metric, one can measure the extent to which the model matches the conditions actually observed in the field.

[0050] Figure 4 plots the actual OA feeder failure rates versus the predicted OA feeder failure rates for underground feeders within a large and mature electrical grid in the northeast. The predicted OA rate is based on SVM regression. No further regression procedures, besides those associated with the SVM machine learning, need be employed. Alternatively, regression analysis can be performed on top of the SVM machine learning ranks.

[0051] A non-limiting embodiment of a CAPT Feeder Section Change Aid is shown in Figure 5, which allows users to calculate the cost and benefits of changing the mix of user-selected cable sections in view of, for example, Load Relief, PILC Replacement and/or preventative maintenance reliability goals. In this example, the proposed strategy of the capital improvements to the infrastructure is based on the feeder that the user selects (700), and the percentage of PILC

cable that is replaced (here 100% or complete removal and replacement of PILC cable) (710). As shown on the right, the CAPT system is provided with mapping functionality (720) to assist the user in selecting the desired feeder sections.

[0052] After the user inputs his or her selections from a set of possible strategies, the proposed changes are summarized in terms of the total sections of cable changed, the cost/section and the total cost of proposed upgrade (730). Other feeder statistics (740) are provided, including actual MTBF, OA feeder failure counts, entire outage counts that include scheduled outage work, and historical data regarding overall feeder performance. As shown on the right, charts (750) are available to visualize the cost vs. benefit tradeoffs of different replacement strategies and feeder selections (see Figures 8 and 9). Information about cable sections (760), "stop" joints (770) that connect PILC cable section types with XLP and EPR section types, transformers (780), and electric Load (790) on the feeder are also provided. This information is a subset of the attributes used in building the SVM model for predicting the MTBF of the feeder after vs before the contemplated changes are made.

[0053] An exemplary output of the CAPT System is provided in Figure 6, in which cost benefit data from various feeders are compared based on user-defined filters and sorts (810) (e.g. Jeopardy network reliability index, ODDS failure susceptibility index, feeder region). Capital improvement data, such as the number of PILC cable sections replaced (820) is also provided. In addition to estimated MTBF and the delta MTBF between predicted and baseline discussed above, the delta MTBF can be normalized by size or cost, such as a % Delta MTBF (830), and Cost Per Day % Delta (840). In this way a small change in MTBF for a poorly performing

feeder with a small baseline MTBF can be established as more valuable than a large change in a good feeder with an already large MTBF because the % Delta MTBF will be higher. Running Cost per feeder (890) and running Total Cost (900) are also provided. In this particular embodiment, cost-per-day MTBF (850), Projected MTBF (860), Current rank before improvement (870) and Projected rank (880) after improvement are also provided.

[0054] Figure 7 demonstrates a cost benefit analysis of proposed capital improvement projects in graphical form, which in turn can be used to determine replacement of PILC cable which provides the most bang for the buck. In this particular embodiment, one is trying to ascertain the cost vs. benefit of based on: a) 15% replacement of PILC cable in a given feeder; b) 75% replacement of PILC cable, and c) 100% replacement of PILC cable within a large, mature electrical grid in the northeast. The network (1000), and selection order of the PILC cable section replacement can also be specified. For a selected network, 100% PILC cable section replacement, sometimes called feeder "purification" or "backboning" efforts can be analyzed based on the benefit, i.e., the percent delta MTBF plotted along the y-axis (1010) against the cost of the feeder purification (1020) along the x-axis.

[0055] As can be ascertained from the line with the highest slope, one can visualize which capital improvement project obtains maximum "bang for the buck." Inflection points, such as the decrease in marginal MTBF gains shown at (1030) indicate areas of diminishing returns. The highlighted line in Figure 7 shows the percent delta MTBF for PILC cable replacement for feeder 1X23. The first point (1040) shows "bang for buck" given a 15% replacement, the second (1030) for 75% replacement and the third (1050) for 100% replacement of PILC. The CAPT

system informs the user that it would be most effective to replace 75% of the PILC sections in this feeder, and devoting resources that would have been devoted to 100% replacement of PILC sections to more work on other feeders instead. For example, as to feeder 01X26, the CAPT system indicates that 100% replacement of PILC cable sections would provide more "bang for buck" as the slope from 75% to 100% (1060) is much higher.

[0056] Various embodiments of the CAPT system provide mapping functionality. Figure 8 is a topographical display of feeders in a particular network (3B), ranked according to their susceptibility to failure. Higher "window panes" or bars indicate a higher likelihood of failure for each feeder section. Figure 9 is a topographical display of cable sections ranked according to their susceptibility to impending failure. Again, higher panes or bars indicate a higher likelihood of failure. Figure 10 provides an alternative display of feeder components, such as stop joints, cable sections and manhole information.

[0057] Figure 11 displays an alternative embodiment of the presently disclosed subject matter, in which the cost/benefit of capital improvement projects is determined in view of providing load relief to cable sections and runs of a feeder, as opposed to being in view of replacing PILC cable with more modern cable as in the previous example. A user can select that all overloaded sections and overloaded PILC runs overloaded at more than 100% or rating be replaced, or only those sections and runs that are overloaded by more than 105%, or more than 110% based on the section and run ratings. It is noted that, alternatively, the proposed capital improvement project can be also described in terms of system reliability, meaning operations and maintenance

performance instead of, for example, in terms of capital PILC cable replacement, or in terms of cable and run load.

[0058] In an alternative embodiment, a "tiered strategy" can be employed to additively provide a strategy for proposed capital improvement projects. In one example, the first mandate is to replace all overloaded sections (Tier 1), then the mandated yearly replacement of PILC sections (Tier 2) -- only a fraction of which is feasible to replace in any one year, and then reliability improvements comprising replacing XLP cable or replacing stop joints, if there is remaining budget (Tier 3). Operationally, Load Pocket Weight (Tier 4) can then be evaluated by comparing cost vs MTBF benefit for closing transformer switches that are currently open, bringing transformer banks that are currently offline back online, repairing low voltage cables called Open Mains, and/or repairing SCADA reporting problems. In this example, the CAPT system is careful in that selections of cable sections in Tier 1 are not available for replacement in Tier 2 or Tier 3 actions to avoid double-counting, and that different combinations of Tier 4 work activities can be summed to produce additional operational risk reduction via increased MTBF at times of stress to the electric grid and its feeders (such as peak summer heat days). Also cable selections chosen in Tier 2 are not available in Tier 3. In addition, reliability actions in each Tier are additive to other actions chosen for every other Tier so that decisions can be made within each Tier and in the overall, combined work plan for the portfolio of all Tiers.

[0059] Figure 12 provides more detail on replacement strategy that is based on the load of the sections (Tier 1 as identified above) and PILC runs (Tier 2 as identified above). For a particular feeder (4X52), in which the user has selected that all overloaded sections and overloaded PILC

runs be replaced, the CAPT systems displays specifics about the overloaded runs (1500) and overloaded non-PILC sections (1510). Similar to outputs in which capital improvement projects are proposed based on % of PILC cable replacement, the CAPT system details feeder statistics and attributes (e.g., transformer data, cable section data) based on capital improvement projects that are prioritized based on load data.

[0060] Figure 13 provides estimated MTBF and delta MTBF data for a capital improvement project that is planned based on the replacement of PILC sections. In this particular example, the number of PILC cable sections is reduced from 55 to 0 in feeder 3B92. This increases the estimated MTBF by 90 days and improves the susceptibility to failure rank of the feeder from 32 to 289.

[0061] Figure 14 provides an example of a proposed capital improvement project that is prioritized based on system reliability. In this example, stop joints are targeted for replacement based on their susceptibility to failure.

[0062] Figure 15 provides a depiction of an additional alternative embodiment according to the presently disclosed subject matter, in which the attribute "Load Pocket Weight" is used as a basis for formulating a proposed operations and maintenance improvement strategy (Tier 4), and the strategy thus proposed is similarly analyzed to determine maximum "bang for the buck," (e.g., delta MTBF vs. cost). Generally, Load Pocket Weight (LPW) is a metric used to define trouble in transferring load from local geographic load pockets of a feeder to the secondary grid via transformers in the distribution network (thusly the Load Pocket Weight nomenclature). Various indicators of such trouble, like as "open (transformer) switches (circuit breakers)," "banks off

transformers offline for various maintenance reasons (e.g., "Live End Capping" that severs several transformers at a terminus of the feeder in order to get the rest of the feeder back online in a load crisis), "open fuses" also for network or circuit protection, "open mains" or secondary cable that is burned out, and Supervisory Control And Data Acquisition (SCADA) reporting problems such as "missing", "open" and "old" SCADA readings, etc., are assigned point values and the total used to indicate where Load Pocket problems are via this total Weight. The proposed capital, operations and maintenance improvement project is then planned using the CAPT tool based on "lowering towards zero" the total load pocket weight of a feeder, i.e., actions are proposed like "close open switches," "bring online banks off When Ready (WR)," "repair open fuses," "repair open mains," and "fix reporting problems" that reduce the LPW toward zero. As shown in Figure 16, a delta MTBF can be determined based on the ODDS predictions of MTBF improvement from actions taken to reduce the load pocket weight.

[0063] Alternatively, the LPW itself can become one of the performance metrics of the CAPT system. A user can select individual maintenance actions to the infrastructure, and the projected LPW based on the changes can be compared to the projected LPW based on the current status of the infrastructure.

[0064] Figure 17 exemplifies a methodology for determining the capital improvement project with the best for the buck for multiple feeders in multiple networks, circuits, or electric grids. Data points corresponding to the cost and delta MTBF for replacing, for example, 50% or 100% of the PILC cable sections in the various feeders are provided. The percentage of PILC cable replaced is a user-specified parameter. Excluding outliers, the upper range of the curve, i.e.,

those points with the highest delta MTBF for a given cost, define an efficient frontier in portfolio management theory and should receive priority given a limited capital improvement budget.

[0065] While the CAPT web application is described largely in the context of a capital, operations and maintenance improvement project within an electrical grid (e.g., replacing stop joints and PILC cable sections in electrical feeders), it is important to note that it is equally applicable to a wide range of other capital improvement, operations and preventive maintenance processes, including but not limited to, chemical processing operations, product manufacturing operations, telecommunications, transportation, civil, gas, pipeline, storage, steam, water, sewer, subway, rail, solar, wind, nuclear and other infrastructure projects. So long as there is a quantifiable performance metric associated with the capital, operations and maintenance improvement, and one or more attributes that also affect the performance metric besides the capital improvement project itself, the CAPT methods of the present application can be used to estimate the costs vs benefits of the improvement project, individually and in totum for the portfolio of the activity.

[0066] In the context of an electrical grid, the performance metric can be, for example, based on the failure (or non-failure) of the component under investigation. Attributes are selected that also impact whether or not the component of the electrical grid fails in the future. In this particular context, attributes can be obtained, for example, based on the results of a "marti-ranking" machine learning algorithm disclosed in International Published Application No. WO 2007/087537, which is hereby incorporated by reference in its entirety.

[0067] When a feeder fails, its substation protection circuitry will isolate if from its power supply in the substation automatically, which is know in the art as an "open auto" or "OA." In one embodiment, the attribute value is the number of OA failures of the feeder under investigation for specified time period, termed the Mean Time Between Failures (MTBF). In another embodiment, the attribute is the number of all emergency outages except planned (SCHEDULED) and including Out on Emergency outages (OOE1, OOE2) identified by the field workers upon examination of specific equipment. For example, the attribute value in one embodiment can be the number of OA outages, "fail on test" outages ("FOT"), and "cut-in open auto" ("CIOA") failures that open upon initial energization after a repair of any kind.

[0068] Additional details regarding the machine learning techniques that can be used in accordance with the presently disclosed systems and methods can be found in U.S. Patent No. 7,395,252, which is hereby incorporated by reference.

[0069] It will be understood that the foregoing is only illustrative of the principles of the disclosed subject matter, and that various modifications can be made by those skilled in the art without departing from the scope and spirit thereof.

APPENDIX A

```
Imports System.Data

Imports System.Data.SqlClient

'Imports Modules

Imports WebserviceClient

Imports UpdateAttributes

Namespace EdisonProgram

  Partial Class CapitalPlanning_Rerank

    Inherits System.Web.UI.Page

#Region " Page Setup "

    Dim VisitedGlbStructures As New ArrayList

    Dim VisitedGlbJoints As New ArrayList

    Dim Duplicate_sections_list As Integer

    Dim Visited_sections_count As Integer

    Shared Uattr As UpdateAttributes.UpdateAttributes


    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load

      If Not Page.IsPostBack Then
```

```
        Uattr = New UpdateAttributes.UpdateAttributes

        SetupPage()



    End If

End Sub

Private Sub SetupPage()

    BindSearchNetworkList()

    BindSearchFeederList()

    CAPTWizard.ActiveStepIndex = 0

    SetupFields()



End Sub

Private Sub SetupFields()

    'ovsec.Visible = True

    'ovruns.Visible = True

    'Runs.Visible = True

    'for now the quury runs too slow....

    'XLP.Visible = False
```

```
'vs2220: added clearing logic for globals

visitedGlbStructures.Clear()

visitedGlbJoints.Clear()

SectionsSummary.Items.Clear()

VisitedGlbRuns.Items.Clear()

Session("strategies_load") = "0"

Session("strategies_pilc") = "0"

Dim visited_visionIds As New ArrayList

visited_visionIds.Add(0)

Session("visited_sections_nonPILC") = visited_visionIds



Dim visited_openswitches As New ArrayList

visited_openswitches.Add(0)

Session("visited_openswitches") = visited_openswitches



ViewState("sortdirection") = ""
```

```
'visitedGlbRuns.Clear()

duplicate_sections_list = 0

visited_sections_count = 0



"added by shringika on 07-30-2008 to clear up summary fields

TotalCostSum.Text = ""

CostPerSectionSum.Text = ""

SectionsChangedSum.Text = ""

sections_area.InnerText = ""




BindAttributes()




BindRankingLPW()

BindFOT()

Reset_New_Attributes()
```

```
'Assumes replace PILC is strategy of choice

'Replace_Paper()

'Calculate_New_Load()

Reset_MTBFs()

Initial_Costs()

'modification by shringika on 03/11/2008 for LPW tier

'BindAttributes1()

'modification ends here 03/11/2008

'ovsec.Visible = False

'ovruns.Visible = False



'Runs.Visible = False

'XLP.Visible = False




VaultDetailsPanel.Visible = False

Dim feeder_id As String = feeder_withzeroes_fix(cmbSearchFeeder.SelectedValue)
```

```
    Uattr.ChangeFeeder(feeder_id)




End Sub



Private Sub SectionsSummarySetup()

    'ovsec.Visible = True

    'ovruns.Visible = True



    'Runs.Visible = True

    'for now the quury runs too slow....

    'XLP.Visible = False



    'vs2220: added clearing logic for globals

    visitedGlbStructures.Clear()

    visitedGlbJoints.Clear()

    SectionsSummary.Items.Clear()
```

```
sections_area.InnerText = ""


SectionsChangedSum.Text = ""

CostPerSectionSum.Text = ""

TotalCostSum.Text = ""

duplicate_sections_list = 0

visited_sections_count = 0


VisitedGlbRuns.Items.Clear()

Session("strategies_load") = "0"

Session("strategies_pilc") = "0"


Dim visited_visionIds As New ArrayList

visited_visionIds.Add(0)

Session("visited_sections_nonPILC") = visited_visionIds


Dim visited_openswitches As New ArrayList

visited_openswitches.Add(0)
```

```
            Session("visited_openswitches") = visited_openswitches



            reset_visitedLPWSessionVariables()

            VaultDetailsPanel.Visible = False

        End Sub

    #End Region



    #Region " Search "

        Protected Sub cmbSearchBorough_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbSearchBorough.SelectedIndexChanged

            '--added by shringika on 20-dec-2007.

            cmbFeederType.SelectedIndex = 0

            cmbPercentage.SelectedIndex = 3

            cmbPercentage.Enabled = False

            '--*modification ends here shringika

            BindSearchNetworkList()

            BindSearchFeederList()

        End Sub
```

```
Protected Sub cmbSearchNetwork_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbSearchNetwork.SelectedIndexChanged

  BindSearchFeederList()

End Sub



Private Sub BindSearchNetworkList()

  cmbSearchNetwork.DataSource = GetNetworkTable(cmbSearchBorough.SelectedValue)

  cmbSearchNetwork.DataValueField = "Network_ID"

  'cmbSearchNetwork.DataValueField = "Network"

  cmbSearchNetwork.DataTextField = "Network_Name"

  'cmbSearchNetwork.DataTextField = "Network"

  cmbSearchNetwork.DataBind()



  '--added by Shringika on 07-Jan-2008 for including "All Networks" option for feeder
selection

  cmbSearchNetwork.Items.Insert(0, New ListItem("All Networks", "All"))
```

```
If cmbSearchBorough.SelectedIndex <> 0 Then cmbSearchNetwork.SelectedIndex = 1

'--*modification by Shringika on 07-jan-2008 ends here


End Sub


Private Sub BindSearchFeederList()

'--modified by shringika to check if overloading percentage selection returns no records on
dec 20 '2007

'--modified on 2 january 2008 to include wizard start disable/enable function

'clear section summary

SectionsSummarySetup()

CAPTWizard.DisplaySideBar = True

Dim btn As Button =
TryCast(CAPTWizard.FindControl("StartNavigationTemplateContainerID").FindControl("StartNextButton"),
Button)

btn.Enabled = True


Dim dl As DataList =
TryCast(CAPTWizard.FindControl("SideBarContainer").FindControl("SideBarList"), DataList)
```

```
dl.Enabled = True

msglabel.Text = ""

'--*modification on 2nd jan ends



If cmbFeederType.SelectedValue.Contains("All") Then



    cmbSearchFeeder.DataSource = GetFeederTable(cmbSearchNetwork.SelectedValue)

    cmbSearchFeeder.DataValueField = "feeder"

    cmbSearchFeeder.DataTextField = "feeder"

    cmbSearchFeeder.DataBind()

    cmbSearchFeeder.SelectedIndex = 0

    SetupFields()

    Exit Sub
```

```
End If

Dim sql1 As String

Dim myConnection1 As New SqlConnection

myConnection1 = New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

' sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "'"

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "'"

If cmbFeederType.SelectedValue.Contains("Load") Then

Dim pct2 As Integer
```

```
pct2 = Int32.Parse(cmbPercentage.SelectedValue)


'--modified by shringika on 7-jan-2008 to include all netwrok case

If cmbSearchNetwork.SelectedIndex <> 0 Then

' sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' AND percentage >= " & pct2 & " "

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' AND percentage >= " & pct2 & " "

Else

If cmbSearchBorough.SelectedIndex <> 0 And cmbSearchNetwork.SelectedValue.Contains("All") Then

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' AND percentage >= " & pct2 & " ORDER BY feeder "

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT
```

zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' AND percentage >= " & pct2 & " ORDER BY feeder "

Else

'all boroughs all n/w case

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND percentage >= " & pct2 & " ORDER BY feeder"

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND percentage >= " & pct2 & " ORDER BY feeder"

End If

End If

End If

```
If cmbFeederType.SelectedIndex = 2 Then

'myConnection1 = New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim pct2 As Integer

pct2 = Int32.Parse(cmbPercentage.SelectedValue)

'--modified by shringika on 7-jan-2008 to include all netwrok case

If cmbSearchNetwork.SelectedIndex <> 0 Then

sql1 = "SELECT [feeder]=(case when (left([EdisonML].[dbo].[Feeders_With_PILC].[feeder],1)=0) then right([EdisonML].[dbo].[Feeders_With_PILC].[feeder],4) else [EdisonML].[dbo].[Feeders_With_PILC].[feeder] end)"

sql1 = sql1 + " FROM [EdisonML].[dbo].[Feeders_With_PILC] INNER JOIN [EdisonML].[dbo].[Feeder_run_length]"

sql1 = sql1 + " ON [EdisonML].[dbo].[Feeders_With_PILC].[feeder]=[EdisonML].[dbo].[Feeder_run_length].[feeder]"

sql1 = sql1 + " WHERE network='" & cmbSearchNetwork.SelectedValue & "' "
```

sql1 = sql1 + " AND (([feeder_pilcruns_length] *100/[feeder_cable_length]) >= " & pct2 & ") "

'sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) "

sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) "

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Feeders_With_PILC WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' AND (pct_PILC >= " & pct2 & ") "

Else

If cmbSearchBorough.SelectedIndex <> 0 And cmbSearchNetwork.SelectedValue.Contains("All") Then

sql1 = "SELECT [feeder]=(case when (left([EdisonML].[dbo].[Feeders_With_PILC].[feeder],1)=0) then right([EdisonML].[dbo].[Feeders_With_PILC].[feeder],4) else [EdisonML].[dbo].[Feeders_With_PILC].[feeder] end)"

sql1 = sql1 + " FROM [EdisonML].[dbo].[Feeders_With_PILC] INNER JOIN [EdisonML].[dbo].[Feeder_run_length]"

sql1 = sql1 + " ON [EdisonML].[dbo].[Feeders_With_PILC].[feeder]=[EdisonML].[dbo].[Feeder_run_length].[feeder]"

sql1 = sql1 + " WHERE borough='" & cmbSearchBorough.SelectedValue & "' "

sql1 = sql1 + " AND (([feeder_pilcruns_length] *100/[feeder_cable_length]) >= " & pct2 & ") "

"sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT zfeeder from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) "

sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_Canonicalfeeder WHERE iscanonical=1) "

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Feeders_With_PILC WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' AND (pct_PILC >= " & pct2 & ") ORDER BY feeder "

Else

'all boroughs all n/w case

sql1 = "SELECT [feeder]=(case when (left([EdisonML].[dbo].[Feeders_With_PILC].[feeder],1)=0) then

right([EdisonML].[dbo].[Feeders_With_PILC].[feeder],4) else [EdisonML].[dbo].[Feeders_With_PILC].[feeder] end)"

sql1 = sql1 + " FROM [EdisonML].[dbo].[Feeders_With_PILC] INNER JOIN [EdisonML].[dbo].[Feeder_run_length]"

sql1 = sql1 + " ON [EdisonML].[dbo].[Feeders_With_PILC].[feeder]=[EdisonML].[dbo].[Feeder_run_length].[feeder]"

'sql1 = sql1 + " WHERE borough='" & cmbSearchBorough.SelectedValue & "' "

sql1 = sql1 + " AND (([feeder_pilcruns_length] *100/[feeder_cable_length]) >= " & pct2 & ") "

' sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) "

sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) "

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Feeders_With_PILC WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND pct_PILC >= " & pct2 & " ORDER BY feeder"

End If

End If

End If

```
'--*modification by shringika on 7-jan-2008 ends here
Dim myCommand1 As New SqlDataAdapter(sql1, myConnection1)
myCommand1.SelectCommand.CommandTimeout = 180
Dim dt1 As DataTable = New DataTable
myCommand1.Fill(dt1)
If dt1.Rows.Count > 0 Then
   msglabel.Text = " "
   FMFeeder.Enabled = True

   '--modification begins here on 20th dec.2007

   'cmbSearchFeeder.DataSource = GetFeederTable(cmbSearchNetwork.SelectedValue)
   cmbSearchFeeder.DataSource = dt1
   cmbSearchFeeder.DataValueField = "feeder"
   cmbSearchFeeder.DataTextField = "feeder"
   cmbSearchFeeder.DataBind()
```

```
            cmbSearchFeeder.SelectedIndex = 0

            SetupFields()



        Else

            FMFeeder.Enabled = False

            msglabel.Text = "no feeder found matching the criteria"

            cmbSearchFeeder.Items.Clear()

            ResetFeederAttributes()




            ' CAPTWizard.DisplaySideBar = False


            ' Dim btn1 As Button =
TryCast(CAPTWizard.FindControl("StartNavigationTemplateContainerID").FindControl("StartNextButton"),
Button)

            ' btn1.Enabled = False

            btn.Enabled = False

            dl.Enabled = False
```

"--*modification on 2 january'2008 for wizard start button display ends

End If

myCommand1 = Nothing

myConnection1.Close()

myConnection1 = Nothing

TargetCostCheck.Visible = False

actualTotalCostPILC.Text = ""

actualPercentPILC.Text = ""

SectionsChangedPILC.Text = ""

TotalCostPILC.Text = ""

RblTotalCostPercentagePILC.Items(0).Enabled = True

downloadtoExcelPanel.Visible = False

RblTotalCostPercentagePILC.SelectedIndex = 0

```
        PercentPILC.SelectedIndex = 0

        CostPercentPanel.Visible = True



        '--modification on 20-dec-2007 ends here

        '--original bindsearchfeederlist() sub



        'cmbSearchFeeder.DataSource = GetFeederTable(cmbSearchNetwork.SelectedValue)

        'cmbSearchFeeder.DataValueField = "FeederID"

        'cmbSearchFeeder.DataTextField = "FeederID"

        'cmbSearchFeeder.DataBind()

        'SetupFields()

    End Sub

    Private Function GetNetworkTable(ByVal borough As String) As DataTable
```

'--modified by shringika on 3-jan-2008 to add "All" borough case

Dim sql As String

If cmbSearchBorough.SelectedValue.Contains("A") Then

'enable all boroughs again -- agb

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE borough <>'M' AND Network_ID IN (SELECT DISTINCT Nwk FROM Attributes.dbo.Rerank_Attributes as A INNER JOIN Attributes.dbo.tblFeeders_2007 as F on (A.Feeder= F.FeederID)) ORDER BY Network_Name"

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Network_ID IN (SELECT DISTINCT Nwk FROM Attributes.dbo.Rerank_Attributes as A INNER JOIN Attributes.dbo.tblFeeders_2007 as F on (A.Feeder= F.FeederID)) ORDER BY Network_Name"

'commented on 01/12/2009 for getting only network_id

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Network_ID IN (SELECT DISTINCT Network from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) ORDER BY Network_Name"

sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Network_ID IN (SELECT DISTINCT Network from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) ORDER BY Network_Name"

'sql = "Select DISTINCT Network from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1"

Else

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Borough='" & borough & "' AND Network_ID IN (SELECT DISTINCT Nwk FROM Attributes.dbo.Rerank_Attributes as A INNER JOIN Attributes.dbo.tblFeeders_2007 as F on (A.Feeder= F.FeederID)) ORDER BY Network_Name"

'commented below on 01/12/2009 for getting only network_id

sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE District_cd ='" & borough & "' AND Network_ID IN (SELECT DISTINCT Network from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) ORDER BY Network_Name"

' sql = "SELECT DISTINCT Network from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1 and Borough='" & borough & "' "

```
End If

'--**end of modification for "All" boroughs case 3-jan-2008 shringika

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(Sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.SelectCommand.CommandTimeout = 120

myCommand.Fill(dt)

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Return dt

End Function
```

```
Private Function GetFeederTable(ByVal network As String) As DataTable

    '--modified by shringika on dec 20'2007

    Dim sql As String

    Dim myConnection As New SqlConnection

    myConnection = New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

    '   '--modified by shringika 7-jan-2008 included if clause for "All" network case--feedrtype = all case



    'cmbfeedertype=all case



    'specific borugh case

    If cmbSearchBorough.SelectedIndex <> 0 Then



      If cmbSearchNetwork.SelectedValue.Contains("All") Then



        'sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) AND borough='" & cmbSearchBorough.SelectedValue & "'ORDER BY feeder"
```

'commented on 01/12/2009 for edisonml

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "'ORDER BY feeder"

sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "'ORDER BY feeder"

Else

'commented on 01/12/2009 for edisonml

'sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) AND borough='" & cmbSearchBorough.SelectedValue & "' and network='" & cmbSearchNetwork.SelectedValue & "'ORDER BY feeder"

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' and network='" & cmbSearchNetwork.SelectedValue & "'ORDER BY feeder"

sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from

EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' and network='" & cmbSearchNetwork.SelectedValue & "'ORDER BY feeder"

End If

Else

'all borough case

If cmbSearchNetwork.SelectedValue.Contains("All") Then

'sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) ORDER BY feeder"

'commented on 01/12/2009 for edisonml

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) ORDER BY feeder"

sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) ORDER BY feeder"

Else

'commented on 01/12/2009 for edisonml

' sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) AND network='" & cmbSearchNetwork.SelectedValue & "' ORDER BY feeder"

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' ORDER BY feeder"

sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' ORDER BY feeder"

End If

End If

'--*modification ends 9-jan-2008

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
        Dim dt As DataTable = New DataTable
        myCommand.SelectCommand.CommandTimeout = 120
        myCommand.Fill(dt)
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing

        Return dt
    End Function

#End Region

#Region " Attributes "

    '0 for now!!
    Private Function drcheck(ByVal dr As DataRow, ByVal item As String) As String
      If IsDBNull(dr(item)) Then Return "9999" Else Return dr(item)
```

```
End Function


Private Sub BindAttributes()
  BindNetworkJeopardyAttributes()
  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
  If cmbSearchFeeder.Items.Count > 0 Then
    Dim fdrvalue As String
    fdrvalue = cmbSearchFeeder.SelectedValue
    If cmbSearchFeeder.SelectedValue.Length < 5 Then
      fdrvalue = "0" + cmbSearchFeeder.SelectedValue
    End If
    '
    'FeederOAChartLink.NavigateUrl = "http://edpro.conedison.net/EdisonProgram/CapitalPlanning/Chart/OA.aspx?feeder=" & fdrvalue
    FeederOAChartLink.NavigateUrl = "Chart/OA.aspx?feeder=" & fdrvalue
    Dim sql As String = "SELECT * FROM Rerank_Attributes WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

Label2.Text = cmbSearchFeeder.SelectedValue

Feeder.Text = cmbSearchFeeder.SelectedValue

Feeder1.Text = cmbSearchFeeder.SelectedValue

Feeder2.Text = cmbSearchFeeder.SelectedValue

Feeder4.Text = cmbSearchFeeder.SelectedValue
```

```
' Feeder5.Text = cmbSearchFeeder.SelectedValue

Feeder6.Text = cmbSearchFeeder.SelectedValue

Feeder7.Text = cmbSearchFeeder.SelectedValue

feeder11.Text = cmbSearchFeeder.SelectedValue

feeder12.Text = cmbSearchFeeder.SelectedValue

feeder20.Text = cmbSearchFeeder.SelectedValue

Feeder_Name.Text = cmbSearchFeeder.SelectedValue

SelectedFeederName.Text = cmbSearchFeeder.SelectedValue




FLoad.Text = drcheck(dr, "mon_Load")

Normal_Rating.Text = drcheck(dr, "mon_Normal_Rating")

Emergency_Rating.Text = drcheck(dr, "mon_Emergency_Rating")

Shift.Text = drcheck(dr, "mon_Shift")

Delta.Text = drcheck(dr, "mon_Delta")

Emergency_Load.Text = drcheck(dr, "mon_Emergency_Load")

Feeders_4kv.Text = drcheck(dr, "mon_4kv")

Paper_GT40.Text = drcheck(dr, "mon_Paper_GT40")
```

```
Paper_LE40.Text = drcheck(dr, "mon_Paper_LE40")

'changed by shringika to exclude mon_solid_LT16 and mon_solid_GT16

Solid_LT16.Text = drcheck(dr, "mon_Solid_LT16")

Solid_70_75.Text = drcheck(dr, "mon_Solid_70_75")

Solid_GT16.Text = drcheck(dr, "mon_Solid_GT16")

Percent_Paper_GT40.Text = drcheck(dr, "mon_percent_Paper_GT40")

Percent_Paper_LE40.Text = drcheck(dr, "mon_percent_Paper_LE40")

Percent_Solid_LT16.Text = drcheck(dr, "mon_percent_Solid_LT16")

Percent_Solid_70_75.Text = drcheck(dr, "mon_percent_Solid_70_75")

Percent_Solid_GT16.Text = drcheck(dr, "mon_percent_Solid_GT16")

Sec_CNT.Text = drcheck(dr, "mon_Sec_CNT")

E21GT93.Text = drcheck(dr, "mon_E21GT93")

E21LE93.Text = drcheck(dr, "mon_E21LE93")

R31GT94.Text = drcheck(dr, "mon_R31GT94")

R31LE93.Text = drcheck(dr, "mon_R31LE93")

OtherStop_LT7.Text = drcheck(dr, "mon_OtherStop_LT7")

OtherStop7_12.Text = drcheck(dr, "mon_OtherStop7_12")

OtherStop_GT12.Text = drcheck(dr, "mon_OtherStop_GT12")
```

```
AllStop.Text = drcheck(dr, "mon_AllStop")

Percent_ALLStop.Text = drcheck(dr, "mon_percent_AllStop")

PJ_LT8.Text = drcheck(dr, "mon_PJ_LT8")

PJ8_16.Text = drcheck(dr, "mon_PJ8_16")

PJ16_24.Text = drcheck(dr, "mon_PJ16_24")

' PJ_GT24.Text = drcheck(dr, "mon_PJ_GT24")

PJ24_32.Text = drcheck(dr, "mon_PJ24-32")

PJ32_40.Text = drcheck(dr, "mon_PJ32-40")

PJGT40.Text = drcheck(dr, "mon_PJGT40")

AllPaper.Text = drcheck(dr, "mon_AllPaper")

Percent_AllPaper.Text = drcheck(dr, "mon_Percent_AllPaper")

SJ_LT8.Text = drcheck(dr, "mon_SJ_LT8")

SJ8_16.Text = drcheck(dr, "mon_SJ8_16")

SJ16_24.Text = drcheck(dr, "mon_SJ16_24")

SJ_GT24.Text = drcheck(dr, "mon_SJ_GT24")

AllSolid.Text = drcheck(dr, "mon_AllSolid")

Percent_AllSolid.Text = drcheck(dr, "mon_percent_AllSolid")

Percent_E21GT93.Text = drcheck(dr, "mon_percent_E21GT93")
```

```
Percent_E21LE93.Text = drcheck(dr, "mon_percent_E21LE93")

Percent_R31GT94.Text = drcheck(dr, "mon_percent_R31GT94")

Percent_R31LE93.Text = drcheck(dr, "mon_percent_R31LE93")

Percent_OtherStop_LT7.Text = drcheck(dr, "mon_percent_OtherStop_LT7")

Percent_OtherStop7_12.Text = drcheck(dr, "mon_percent_OtherStop7_12")

Percent_OtherStop_GT12.Text = drcheck(dr, "mon_percent_OtherStop_GT12")

Percent_PJ_LT8.Text = drcheck(dr, "mon_percent_PJ_LT8")

Percent_PJ8_16.Text = drcheck(dr, "mon_percent_PJ8_16")

Percent_PJ16_24.Text = drcheck(dr, "mon_percent_PJ16_24")

'Percent_PJ_GT24.Text = drcheck(dr, "mon_percent_PJ_GT24")

Percent_PJ24_32.Text = drcheck(dr, "mon_percent_PJ24-32")

Percent_PJ32_40.Text = drcheck(dr, "mon_percent_PJ32-40")

Percent_PJGT40.Text = drcheck(dr, "mon_percent_PJGT40")

Percent_SJ_LT8.Text = drcheck(dr, "mon_percent_SJ_LT8")

Percent_SJ8_16.Text = drcheck(dr, "mon_percent_SJ8_16")

Percent_SJ16_24.Text = drcheck(dr, "mon_percent_SJ16_24")

Percent_SJ_GT24.Text = drcheck(dr, "mon_percent_SJ_GT24")

Joint_CNT.Text = drcheck(dr, "mon_Joint_CNT")
```

```
XFMR_GT40.Text = drcheck(dr, "mon_XFMR_GT40")

XFMR30_39.Text = drcheck(dr, "mon_XFMR30_39")

XFMR20_29.Text = drcheck(dr, "mon_XFMR20_29")

XFMR10_19.Text = drcheck(dr, "mon_XFMR10_19")

XFMR0_9.Text = drcheck(dr, "mon_XFMR0_9")

W_XFMR_GT40.Text = drcheck(dr, "mon_W_XFMR_GT40")

W_XFMR30_39.Text = drcheck(dr, "mon_W_XFMR30_39")

W_XFMR20_29.Text = drcheck(dr, "mon_W_XFMR20_29")

W_XFMR10_19.Text = drcheck(dr, "mon_W_XFMR10_19")

W_XFMR0_9.Text = drcheck(dr, "mon_W_XFMR0_9")

Percent_XFMR_GT40.Text = drcheck(dr, "mon_percent_XFMR_GT40")

Percent_XFMR30_39.Text = drcheck(dr, "mon_percent_XFMR30_39")

Percent_XFMR20_29.Text = drcheck(dr, "mon_percent_XFMR20_29")

Percent_XFMR10_19.Text = drcheck(dr, "mon_percent_XFMR10_19")

Percent_XFMR0_9.Text = drcheck(dr, "mon_percent_XFMR0_9")

Percent_W_XFMR_GT40.Text = drcheck(dr, "mon_percent_W_XFMR_GT40")

Percent_W_XFMR30_39.Text = drcheck(dr, "mon_percent_W_XFMR30_39")

Percent_W_XFMR20_29.Text = drcheck(dr, "mon_percent_W_XFMR20_29")
```

```
            Percent_W_XFMR10_19.Text = drcheck(dr, "mon_percent_W_XFMR10_19")

            Percent_W_XFMR0_9.Text = drcheck(dr, "mon_percent_W_XFMR0_9")

            XFMR_CNT_SUMMED.Text = drcheck(dr, "mon_XFMR_CNT_SUMMED")

            fJeopardy.Text = drcheck(dr, "Jeopardy")

            fCNT.Text = drcheck(dr, "CNT")




            myCommand = Nothing

            myConnection.Close()

            myConnection = Nothing

        End If

    End Sub


    Private Sub BindNetworkJeopardyAttributes()

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found


        'deals with the All network selection case
```

```
If cmbSearchNetwork.SelectedValue.Contains("All") Then

  base_ntwk_jeopardy_ranking.Text = ""

  Exit Sub

End If



If cmbSearchFeeder.Items.Count > 0 Then



  Dim sql As String = "SELECT [2008 Ranking] as ntwk_ranking, ntwk FROM
[Network_Jeopardy_2008_5_31_08] where ntwk = '" & cmbSearchNetwork.SelectedValue & "'"



  Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

  Dim myCommand As New SqlDataAdapter(sql, myConnection)



  Dim dt As DataTable = New DataTable

  myCommand.Fill(dt)
```

```
        Dim dr As DataRow = dt.Rows(0)

        base_ntwk_jeopardy_ranking.Text = drcheck(dr, "ntwk_ranking")




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If

End Sub

Private Sub Reset_New_Attributes()




    mon_Load.Text = FLoad.Text

    mon_Normal_Rating.Text = Normal_Rating.Text

    mon_Emergency_Rating.Text = Emergency_Rating.Text

    mon_Shift.Text = Shift.Text

    mon_Delta.Text = Delta.Text

    mon_Emergency_Load.Text = Emergency_Load.Text
```

mon_4kv.Text = Feeders_4kv.Text

mon_Paper_GT40.Text = Paper_GT40.Text

mon_Paper_LE40.Text = Paper_LE40.Text

mon_Solid_LT16.Text = Solid_LT16.Text

mon_Solid_70_75.Text = Solid_70_75.Text

mon_Solid_GT16.Text = Solid_GT16.Text

mon_Percent_Paper_GT40.Text = Percent_Paper_GT40.Text

mon_Percent_Paper_LE40.Text = Percent_Paper_LE40.Text

mon_Percent_Solid_LT16.Text = Percent_Solid_LT16.Text

mon_Percent_Solid_70_75.Text = Percent_Solid_70_75.Text

mon_Percent_Solid_GT16.Text = Percent_Solid_GT16.Text

mon_Sec_CNT.Text = Sec_CNT.Text

mon_E21GT93.Text = E21GT93.Text

mon_E21LE93.Text = E21LE93.Text

mon_R31GT94.Text = R31GT94.Text

mon_R31LE93.Text = R31LE93.Text

mon_OtherStop_LT7.Text = OtherStop_LT7.Text

mon_OtherStop7_12.Text = OtherStop7_12.Text

mon_OtherStop_GT12.Text = OtherStop_GT12.Text

mon_AllStop.Text = AllStop.Text

mon_Percent_ALLStop.Text = Percent_ALLStop.Text

mon_PJ_LT8.Text = PJ_LT8.Text

mon_PJ8_16.Text = PJ8_16.Text

mon_PJ16_24.Text = PJ16_24.Text

'mon_PJ_GT24.Text = PJ_GT24.Text

mon_PJ24_32.Text = PJ24_32.Text

mon_PJ32_40.Text = PJ32_40.Text

mon_PJGT40.Text = PJGT40.Text

mon_AllPaper.Text = AllPaper.Text

mon_Percent_AllPaper.Text = Percent_AllPaper.Text

mon_SJ_LT8.Text = SJ_LT8.Text

mon_SJ8_16.Text = SJ8_16.Text

mon_SJ16_24.Text = SJ16_24.Text

mon_SJ_GT24.Text = SJ_GT24.Text

mon_AllSolid.Text = AllSolid.Text

mon_Percent_AllSolid.Text = Percent_AllSolid.Text

mon_Percent_E21GT93.Text = Percent_E21GT93.Text

mon_Percent_E21LE93.Text = Percent_E21LE93.Text

mon_Percent_R31GT94.Text = Percent_R31GT94.Text

mon_Percent_R31LE93.Text = Percent_R31LE93.Text

mon_Percent_OtherStop_LT7.Text = Percent_OtherStop_LT7.Text

mon_Percent_OtherStop7_12.Text = Percent_OtherStop7_12.Text

mon_Percent_OtherStop_GT12.Text = Percent_OtherStop_GT12.Text

mon_Percent_PJ_LT8.Text = Percent_PJ_LT8.Text

mon_Percent_PJ8_16.Text = Percent_PJ8_16.Text

mon_Percent_PJ16_24.Text = Percent_PJ16_24.Text

'mon_Percent_PJ_GT24.Text = Percent_PJ_GT24.Text

mon_Percent_PJ24_32.Text = Percent_PJ24_32.Text

mon_Percent_PJ32_40.Text = Percent_PJ32_40.Text

mon_Percent_PJGT40.Text = Percent_PJGT40.Text

mon_Percent_SJ_LT8.Text = Percent_SJ_LT8.Text

mon_Percent_SJ8_16.Text = Percent_SJ8_16.Text

mon_Percent_SJ16_24.Text = Percent_SJ16_24.Text

mon_Percent_SJ_GT24.Text = Percent_SJ_GT24.Text

mon_Joint_CNT.Text = Joint_CNT.Text

mon_XFMR_GT40.Text = XFMR_GT40.Text

mon_XFMR30_39.Text = XFMR30_39.Text

mon_XFMR20_29.Text = XFMR20_29.Text

mon_XFMR10_19.Text = XFMR10_19.Text

mon_XFMR0_9.Text = XFMR0_9.Text

mon_W_XFMR_GT40.Text = W_XFMR_GT40.Text

mon_W_XFMR30_39.Text = W_XFMR30_39.Text

mon_W_XFMR20_29.Text = W_XFMR20_29.Text

mon_W_XFMR10_19.Text = W_XFMR10_19.Text

mon_W_XFMR0_9.Text = W_XFMR0_9.Text

mon_Percent_XFMR_GT40.Text = Percent_XFMR_GT40.Text

mon_Percent_XFMR30_39.Text = Percent_XFMR30_39.Text

mon_Percent_XFMR20_29.Text = Percent_XFMR20_29.Text

mon_Percent_XFMR10_19.Text = Percent_XFMR10_19.Text

mon_Percent_XFMR0_9.Text = Percent_XFMR0_9.Text

mon_Percent_W_XFMR_GT40.Text = Percent_W_XFMR_GT40.Text

mon_Percent_W_XFMR30_39.Text = Percent_W_XFMR30_39.Text

```
            mon_Percent_W_XFMR20_29.Text = Percent_W_XFMR20_29.Text

            mon_Percent_W_XFMR10_19.Text = Percent_W_XFMR10_19.Text

            mon_Percent_W_XFMR0_9.Text = Percent_W_XFMR0_9.Text

            mon_XFMR_CNT_SUMMED.Text = XFMR_CNT_SUMMED.Text

            Jeopardy.Text = fJeopardy.Text

            CNT.Text = fCNT.Text

            LPW.Text = fLPW.Text

            AVG_LPW.Text = fAVG_LPW.Text

        End Sub



#End Region



#Region " Ranking & LPW "



        Private Sub BindRankingLPW()

            If cmbSearchFeeder.Items.Count > 0 Then

                Dim feeder As String
```

```
feeder = cmbSearchFeeder.SelectedValue

If Feeder.Length < 5 Then

  Feeder = "0" + Feeder.ToString

End If

' Dim sql As String = "SELECT * FROM FeederSuscept.dbo.rerank_dynamic_Attrib WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim sql As String = "SELECT * FROM LoadPocket.dbo.latest_lpw WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

'Dim sql As String = "SELECT * FROM [Transformer_OM_BO].[dbo].[loadpocketweight_test] WHERE Feeder = '" + feeder + "'"



'Dim sql As String = "SELECT * FROM rerank_dynamic_Attrib WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("DB2FeederSuscept").ConnectionString)

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300



Dim dt As DataTable = New DataTable



myCommand.Fill(dt)

If dt.Rows.Count = 0 Then

   fAVG_LPW.Text = "0"

   fLPW.Text = "0"

   Exit Sub

End If



Dim dr As DataRow = dt.Rows(0)

'getting the rank from the precomputed table now

'RankBaseLoad.Text = drcheck(dr, "Rank")
```

```
        Dim x, y, z As Integer



        x = CInt(drcheck(dr, "Total_weight"))

        y = CInt(drcheck(dr, "RMS_weight"))

        z = CInt(drcheck(dr, "no_of_xfrs"))

        fLPW.Text = Convert.ToString(x + y)

        ' IR_ScanTime.Text = dr("IR_nwk_scan_datetime").ToString

        ' fAVG_LPW.Text = drcheck(dr, "AVG_LPW")

        fAVG_LPW.Text = Convert.ToString(x / z)



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing



        IR_Scandatetime()

    End If

End Sub
```

```
Private Sub BindFOT()

    If cmbSearchFeeder.Items.Count > 0 Then

        'Dim sql As String = "SELECT Isnull(mtbf,9999) AS MTBF, Isnull(stdevmtbf,9999)
AS StDevMTBF, Isnull(countoa,0) AS CountOA, Isnull(countcioa,0)AS CountCIOA, Isnull(countalloutages,0) AS
CountAllOutages, ISnull(avgoutageduration,9999) AS AvgOutageDuration, IsNull(lastautodate, '1900-01-01') as
LastAutoDate,Isnull(Datediff(DAY,lastautodate,Getdate()),9999)  AS DaysLastOA, IsNull(lastscheduledworkdate,
'1900-01-01') as LastScheduledWorkDate,Isnull(lastsuccessfulhipotdate,'1900-01-01')  AS LastSuccessfulHipotDate
from FeederStudy.dbo.tblFeederStats WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

        'Dim sql As String = "SELECT Isnull(mtbf,9999) AS MTBF, Isnull(stdevmtbf,9999)
AS StDevMTBF, Isnull(countoa,0) AS CountOA, Isnull(countcioa,0)AS CountCIOA, Isnull(countalloutages,0) AS
CountAllOutages, ISnull(avgoutageduration,9999) AS AvgOutageDuration, IsNull(lastautodate, '1900-01-01') as
LastAutoDate,Isnull(Datediff(DAY,lastautodate,Getdate()),9999)  AS DaysLastOA, IsNull(lastscheduledworkdate,
'1900-01-01') as LastScheduledWorkDate,Isnull(lastsuccessfulhipotdate,'1900-01-01')  AS LastSuccessfulHipotDate
from B18003R1.ENG.dbo.tblFeederStats WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

        Dim sql As String = "SELECT * from
[Attributes].[dbo].[Rerank_CAPT_tblFeederStats] WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

        Dim myCommand As New SqlDataAdapter(sql, myConnection)

        myCommand.SelectCommand.CommandTimeout = 120

        Dim dt As DataTable = New DataTable
```

```
myCommand.Fill(dt)



ACTMtbf.Text = ""

'Other stats fill

'StDevMTBF.Text = dr("StDevMTBF")

CountOA.Text = ""

CountCIOA.Text = ""

CountAllOutages.Text = ""

LastAutoDate.Text = ""

DaysLastOA.Text = ""

LastScheduledWorkDate.Text = ""

LastSuccessfulHipotDate.Text = ""



If (dt.Rows.Count > 0) Then

   Dim dr As DataRow = dt.Rows(0)
```

```
ACTMtbf.Text = dr("MTBF")

'Other stats fill

'StDevMTBF.Text = dr("StDevMTBF")

CountOA.Text = dr("CountOA")

CountCIOA.Text = dr("CountCIOA")

CountAllOutages.Text = dr("CountAllOutages")

LastAutoDate.Text = dr("LastAutoDate")

DaysLastOA.Text = dr("DaysLastOA")

LastScheduledWorkDate.Text = dr("LastScheduledWorkDate")

LastSuccessfulHipotDate.Text = dr("LastSuccessfulHipotDate")



'LastSuccessfulHipotDate.Text = cmbSearchFeeder.SelectedValue



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Else
```

```
                LastAutoDate.Text = "No Stats New Feeder"


            End If

        End If


    End Sub


#End Region


        Protected Sub cmbSearchFeeder_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbSearchFeeder.SelectedIndexChanged
```

'--modified by shringika on 20 dec. included if an end if to check 'itemcount in cmbsearchfeeder

If cmbSearchFeeder.Items.Count > 0 Then

'ovsec.Visible = True

'ovruns.Visible = True

'PILC.Visible = True

'Runs.Visible = True

'XLP.Visible = False

'ELAS.Visible = True

' Dim fdrcheck As String

'fdrcheck = cmbSearchFeeder.SelectedValue

'If cmbSearchFeeder.SelectedValue.Length < 5 Then

'fdrcheck = "0" + cmbSearchFeeder.SelectedValue

'End If

'FeederOAChartLink.NavigateUrl =
"http://edpro.conedison.net/EdisonProgramDevAlpha/CapitalPlanning/Chart/OA.aspx?feeder=" & fdrcheck

'Link included in bindattributes

BindAttributes()

reset_lpwtotals()

BindRankingLPW()

BindFOT()

Reset_New_Attributes()

'Assumes replace PILC is strategy of choice

'Replace_Paper()

'Calculate_New_Load()

Reset_MTBFs()

Initial_Costs()

SectionsSummarySetup()

'modification by shringika on 03/11/2008 for LPW tier

'BindAttributes1()

```
'modification ends here LPW



'ovsec.Visible = False

'ovruns.Visible = False

'PILC.Visible = False

'Runs.Visible = False

'XLP.Visible = False



'delete later

'ELAS.Visible = False



ovsec.Visible = False

ovruns.Visible = False

'runstop.Visible = False



Runs.Visible = False

XLP.Visible = False
```

```
PILCCablesWithStopJoints.Visible = False

GVPILCStopJoints.Visible = False

GVRelStopsPanel.Visible = False

GVRelElastPanel.Visible = False

downloadtoExcelPanel.Visible = False

UpdateAttributesElast.Visible = False

RelGVElastPanel.Visible = False

PILCCalculationReset()

PILCCablesWorstStopJoints.Visible = False

Runs_Select.Visible = False

' change feeder for update attribute

Dim feeder_id As String



feeder_id = feeder_withzeroes_fix(cmbSearchFeeder.SelectedValue)



Uattr.ChangeFeeder(feeder_id)



TargetCostCheck.Visible = False
```

```
            actualTotalCostPILC.Text = ""

            actualPercentPILC.Text = ""

            SectionsChangedPILC.Text = ""

            TotalCostPILC.Text = ""

            RblTotalCostPercentagePILC.Items(0).Enabled = True

            downloadtoExcelPanel.Visible = False

            RblTotalCostPercentagePILC.SelectedIndex = 0



            PercentPILC.SelectedIndex = 0

            CostPercentPanel.Visible = True



        End If



    End Sub
```

```
    Protected Sub Paper_GT40_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Paper_GT40.TextChanged
```

```
Percent_Paper_GT40.Text = Int32.Parse(Paper_GT40.Text) / Int32.Parse(Sec_CNT.Text)
* 100.0

'Convert.ToDouble(Paper_GT40.Text).ToString()



End Sub

''' <summary>

''' returns a table of overloaded sections for replacement

''' </summary>

''' <param name="Feeder"> selected fedeer</param>

''' <param name="PILC"> whether pilc cables or non pilc</param>

''' <returns> a table of ocverloaded sections</returns>

''' <remarks></remarks>



Private Function GetOvsecTable(ByVal Feeder As String, ByVal PILC As Boolean) As
DataTable

Dim sql As String

If Feeder.Length < 5 Then Feeder = "0" + Feeder.ToString
```

```
Dim pct As Integer

If StrategyLoadDropDown.SelectedIndex = 2 Then

  pct = 100

ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then

  pct = 105

Else

  pct = 110

End If

If PILC Then

  'sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder +
"'  AND isnull(insulation,'UNK') = 'paper'"

  sql = "SELECT  (select count(*) from EdisonML.dbo.Rerank_OVL WHERE Feeder =
'" + Feeder + "' AND load_percentage >=" & pct & " AND isnull(insulation,'UNK') = 'paper') as ovl_cnt,(case when
insulation is null then  'epr' else insulation end) as insulation  , [borough] ,[network],[vision_id],[feeder]
,[from_struct],[to_struct]  ,[load_percentage]  ,[run_id] ,[installation_date]  from edisonml.dbo.Rerank_OVL
WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') = 'paper'"


  'sql = "SELECT  borough,network,feeder,From_struct as [from struct], to_struct as [to
struct], insulation,load_percentage, run_id as ,vision_id , installation_date as year  from  edisonml.dbo.Rerank_OVL
WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') = 'paper'"
```

```
Else

sql = "SELECT  (select count(*) from EdisonML.dbo.Rerank_OVL WHERE Feeder =
'" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper') as ovl_cnt, (case
when insulation is null then  'epr' else insulation end) as insulation  , [borough] ,[network],[vision_id],[feeder]
,[from_struct],[to_struct]  ,[load_percentage]  ,[run_id]  ,[installation_date]  from  Edisonml.dbo.Rerank_OVL
WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

'sql = "SELECT  borough,network,feeder,From_struct as [from struct], to_struct as [to
struct], insulation,load_percentage, run_id, vision_id, installation_date as year  from  edisonml.dbo.Rerank_OVL
WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

End If

Dim  myConnection  As  New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 120

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



myCommand = Nothing

myConnection.Close()
```

```
        myConnection = Nothing



        Return dt

    End Function

    Private Function GetOvsecTable_EXCEL(ByVal Feeder As String, ByVal PILC As
Boolean) As DataTable

        Dim sql As String

        If Feeder.Length < 5 Then Feeder = "0" + Feeder.ToString

        Dim pct As Integer

        If StrategyLoadDropDown.SelectedIndex = 2 Then

          pct = 100

        ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then

          pct = 105

        Else

          pct = 110

        End If

        If PILC Then
```

sql = "SELECT borough, network,feeder, From_struct as [from struct], to_struct as [to struct], insulation, load_percentage, run_id ,vision_id , installation_date from edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' and run_id in(select run_id from edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') = 'paper')"

Else

'sql = "SELECT (select count(*) from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper') as ovl_cnt,* from Edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

sql = "SELECT borough, network, feeder, From_struct as [from struct], to_struct as [to struct], insulation, load_percentage, run_id, vision_id, installation_date from edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

myCommand = Nothing

```
        myConnection.Close()

        myConnection = Nothing



        Return dt

    End Function



    Private Function GetsecTable(ByVal Feeder As String, ByVal PILC As Boolean) As
DataTable

        Dim sql As String



        If (Feeder.Length < 5) Then

            Feeder = "0" + Feeder.ToString

        End If



        If PILC Then

            ' commented to change to edisonml 01/13/2009
```

```
sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND isnull(insulation,'UNK') = 'paper'"

'sql  =  "SELECT  *  from  EdisonML.dbo.Rerank_OVL  WHERE  (case  when
left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "'  AND isnull(insulation,'UNK') = 'paper'"

Else

sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND isnull(insulation,'UNK') != 'paper'"

'sql  =  "SELECT  *  from  EdisonML.dbo.Rerank_OVL  WHERE  (case  when
left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND isnull(insulation,'UNK') != 'paper'"

End If

Dim  myConnection  As  New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)


Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


myCommand = Nothing

myConnection.Close()
```

```
myConnection = Nothing


Return dt

End Function


Private Function GetOvrunTable(ByVal Feeder As String) As DataTable

'Dim sql As String = "SELECT * FROM Attributes.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "' and percentage_load >= 100"


'commented on 01/12/2009 for edisonml

'Dim sql As String = "SELECT (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" + Feeder + "' AND percentage_load >= 100) as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from Attributes.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND percentage_load >= 100"

'commented on 01/14/2009 to exclude deltamtbf column because of bulk_reran.....view missing
```

'Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE (case when left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND percentage_load >= 100) as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON (case when left(Runs.feeder,1)=0 then right(Runs.feeder,4) else Runs.feeder end)= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND percentage_load >= 100"

```
If (Feeder.Length < 5) Then
   Feeder = "0" + Feeder.ToString
End If


Dim pct As Integer
If StrategyLoadDropDown.SelectedIndex = 2 Then
   pct = 100
ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then
   pct = 105
Else
   pct = 110
End If
'commented below sql on 03/05/2009 for picking up delta information
```

' Dim sql As String = "SELECT *,length as ovl_cnt FROM EdisonML.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "' and percentage_load >=" & pct & ""

'added on 03/05/2009

Dim sql As String = "SELECT length as ovl_cnt , (case when delta is null then 0 else cast([delta] as decimal(4,1)) end) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND Runs.Run_id is not null and percentage_load >= " & pct & ""

'Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE feeder = '" + Feeder + "' AND percentage_load >= 100) as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN Attributes.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON (case when left(Runs.feeder,1)=0 then right(Runs.feeder,4) else Runs.feeder end)= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND percentage_load >= 100"

'Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE (case when left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND percentage_load >=" & pct & ") as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON (case when left(Runs.feeder,1)=0 then

right(Runs.feeder,4) else Runs.feeder end)= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' and percentage_load >=" & pct & ""

'SELECT Runs.feeder, Runs.Run_ID, Runs.rank,percentage_load, Runs.LoadXrank, Runs.length, (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, cast([delta] as decimal(4,1)) as delta from Attributes.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & Feeder & "' ORDER BY Runs.rank DESC

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

```
    Return dt

End Function


Private Function GetPILCLoadTable(ByVal Feeder As String) As DataTable

    If (Feeder.Length < 5) Then

        Feeder = "0" + Feeder.ToString

    End If


    Dim sql As String = "SELECT [csa], [nwk], [feeder], [from_struc], [to_struc], [Insulation]
,(case when [load_percentage] is null then 99999 else [load_percentage] END) as load_percentage, [run_id] ,
[length] , [From_Leg], [To_Leg], [Vision_ID] ,[installation_date], (case when [boro cable rank] is null then 99999
else [boro cable rank] END) as [boro cable rank] ,[boro cable 5tile grp] ,(case when [nwk cable rank] is null then
99999 else [nwk cable rank] END) as [nwk cable rank] ,[nwk cable 5tile grp]  ,(case when [feeder pilc rank] is null
then 99999 else [feeder pilc rank] END) as [feeder pilc rank] ,[feeder pilc 5tile grp],[description],(case when
[LoadXrank] is null then 0 else [Loadxrank] END) as Loadxrank FROM Rerank_pilc_load  WHERE Run_id is not
null and Feeder = '" + Feeder + "'"

    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)
```

```
    Dim dt As DataTable = New DataTable

    myCommand.SelectCommand.CommandTimeout = 60

    myCommand.Fill(dt)



    myCommand = Nothing

    myConnection.Close()

    myConnection = Nothing



    Return dt

End Function



Private Function GetRunLoadTable(ByVal Feeder As String) As DataTable

    If (Feeder.Length < 5) Then

        Feeder = "0" + Feeder.ToString

    End If
```

'probably need to provide the lower rank section as the rank

'Dim sql As String = "SELECT (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, * from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "'"

Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, * from Edisonml.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "'"

'Dim sql As String = "SELECT * FROM Attributes.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

```
Dim pertotColumn As DataColumn = New DataColumn



With totalColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "tsec"
   '    .Expression = "sum(length)"
End With
With perColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "psec"
End With
With pertotColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "ptsec"
End With
With dt.Columns
   .Add(totalColumn)
   .Add(perColumn)
```

```
        .Add(pertotColumn)
    End With

    myCommand = Nothing
    myConnection.Close()
    myConnection = Nothing

    Return dt
End Function

Private Function GetPILCRunLoadTable(ByVal Feeder As String) As DataTable
    If (Feeder.Length < 5) Then
        Feeder = "0" + Feeder.ToString
    End If

    'probably need to provide the lower rank section as the rank
```

'commented on 01/15/2009 for excluding delta mtbf column

'Dim sql As String = "SELECT Runs.feeder, Runs.Run_ID, Runs.rank,percentage_load, Runs.LoadXrank, Runs.length, (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, cast([delta] as decimal(4,1)) as delta from Attributes.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & Feeder & "' ORDER BY Runs.rank DESC"

'modified on 03/05/2009 for including delta info from bulk_rerank

'Dim sql As String = "SELECT feeder,Run_ID,rank,percentage_load,LoadXrank,length, (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt from EdisonML.dbo.Rerank_PILC_Runs WHERE feeder = '" & Feeder & "' ORDER BY rank DESC"

'(case when [Feeder Joint Rank] is null then 99999 else [feeder joint rank] END)

Dim sql As String = "SELECT Runs.feeder, Runs.Run_ID,(case when Runs.rank is null then 99999 else Runs.rank end) as rank, (case when [percentage_load] is null then 99999 else [percentage_load] END) as percentage_load, (case when [LoadXrank] is null then 0 else [Loadxrank] END) as Loadxrank, Runs.length, (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, (case when delta is null then 0 else cast([delta] as decimal(4,1)) end) as delta, is_spur, is_basket from EdisonML.dbo.Rerank_PILC_Runs AS Runs LEFT OUTER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & Feeder & "' and Runs.Run_id is not null ORDER BY Runs.rank DESC"

'modification on 03/05/2009 for delta ends here

'Dim sql As String = "SELECT feeder,Run_ID,rank,percentage_load,LoadXrank,length, (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt from Attributes.dbo.Rerank_PILC_Runs WHERE feeder = '" & Feeder & "' ORDER BY rank DESC"

'Dim sql As String = "SELECT * FROM Attributes.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

```
With totalColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "tsec"
'    .Expression = "sum(length)"
End With
With perColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "psec"
End With
With pertotColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "ptsec"
End With
With dt.Columns
   .Add(totalColumn)
   .Add(perColumn)
   .Add(pertotColumn)
End With
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing


        Return dt

    End Function


    Private Function GetOVLLoadTableForRun(ByVal RunId As String) As DataTable

        Dim sql As String = "SELECT * FROM Rerank_OVL  WHERE [run_id] = '" + RunId +
"'"

        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

        Dim myCommand As New SqlDataAdapter(sql, myConnection)


        Dim dt As DataTable = New DataTable
```

```
myCommand.Fill(dt)

myCommand = Nothing
myConnection.Close()
myConnection = Nothing

Return dt
End Function

Private Function GetCableById(ByVal CableId As String) As DataTable
Dim sql As String = "SELECT * FROM Rerank_OVL  WHERE [vision_id] = '" + CableId
+ "'"
Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)
Dim myCommand As New SqlDataAdapter(sql, myConnection)
'SectionsSummary.Items.Add(sql)
Dim dt As DataTable = New DataTable
myCommand.Fill(dt)
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing


        Return dt

    End Function


    Private Function GetJointsForCable(ByVal Feeder As String, ByVal Str As String) As
DataTable

        Dim sql As String = "SELECT * FROM Rerank_CAPT_jeoJoints WHERE Feeder = '" +
Feeder + "' and Structure = '" + Str + "'"

        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

        Dim myCommand As New SqlDataAdapter(sql, myConnection)


        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)
```

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing


Return dt

End Function


Private Function GetJointsForCable(ByVal Feeder As String, ByVal Str As String, ByVal Leg As String) As DataTable

Dim sql As String = "SELECT * FROM Rerank_CAPT_jeoJoints WHERE Feeder = '" + Feeder + "' and Structure = '" + Str + "' and Leg = '" + Leg + "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)


Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


myCommand = Nothing
```

```
        myConnection.Close()

        myConnection = Nothing



        Return dt

    End Function



    Private Function GetXLPTable(ByVal Feeder As String) As DataTable

        If Feeder.Length < 5 Then

            Feeder = "0" + Feeder

        End If

        'modified by RTau 01/15

        Dim sql As String = "SELECT RL.feeder, RL.run_id,(case when RL.[Feeder PILC
RANK] is null then 99999 else RL.[Feeder PILC RANK] END) AS Feeder_Cable_RANK,(case when
RL.[load_percentage] is null then 99999 else RL.[load_percentage] END) as load_percentage, (case when
RL.[LoadXrank] is null then 99999 else RL.[Loadxrank] END) as LoadXrank, RLR.length as length,
RL.Description, RL.INSTALLATION_DATE, RL.insulation FROM EdisonML.dbo.Rerank_Load RL,
Edisonml.dbo.Rerank_Load_Runs RLR WHERE RL.run_id = RLR.run_id and RL.Feeder = '" & Feeder & "' AND
(RL.insulation = 'XLP') AND (year(RL.INSTALLATION_DATE) between 1970 and 1975)"

        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("edisonml").ConnectionString)
```

```
Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 120



Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

With totalColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "tsec"

End With

With perColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "psec"

End With

With pertotColumn
```

```
        .DataType = System.Type.GetType("System.Decimal")
        .ColumnName = "ptsec"
    End With
    With dt.Columns
        .Add(totalColumn)
        .Add(perColumn)
        .Add(pertotColumn)
    End With

    myCommand = Nothing
    myConnection.Close()
    myConnection = Nothing

    Return dt
End Function

Private Function GetElastimoldTable(ByVal Feeder As String) As DataTable
```

'Changed by RTau 12/13/07 Dim sql As String = "SELECT * FROM [Rerank_Joints] WHERE [Feeder] = '" + Feeder + "' AND [Stop] = 'stop' and [Type] = 'ELASTIMOLD' and [Config] = '2W-1W'"

' Dim sql As String = "Select distinct Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints WHERE Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by run_id order by min([Feeder Joint Rank]) asc"

'Dim sql As String = "SELECT Runs.[Run_ID],Runs.[rank] as worst_joint_rank,Runs.[elastimold],Runs.[RAYCHEM] ,Runs.[other_stops],runs.[length], Runs.[stops],cast([delta] as decimal(4,1)) as delta FROM [Rerank_PILC_Runs_Joints_Types_2008] AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & cmbSearchFeeder.SelectedValue & "' and stops > 0 order by elastimold desc,raychem desc, other_stops desc"

'Dim sql As String = "Select distinct Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints WHERE Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by run_id order by min([Feeder Joint Rank]) asc"

'Dim sql As String = "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as delta FROM Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

'feeder_withzeroes_fix(Feeder)

If Feeder.Length < 5 Then

Feeder = "0" + Feeder.ToString

End If

'modified on 03/05/2009 for including delta mtbf column

Dim sql As String = "Select distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as delta FROM Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE runs.Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by Runs.run_id order by min([Feeder Joint Rank]) asc"

' Dim sql As String = "Select distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints AS Runs WHERE runs.Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by Runs.run_id order by min([Feeder Joint Rank]) asc"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

'takes 30 secs to do!

myCommand.SelectCommand.CommandTimeout = 60

Dim dt As DataTable = New DataTable

```
myCommand.Fill(dt)



'******  added by RTau 12/12/07  ***************

Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

Dim secColumn As DataColumn = New DataColumn

With totalColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "tsec"

End With

With perColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "psec"

End With

With pertotColumn

   .DataType = System.Type.GetType("System.Decimal")
```

```
    .ColumnName = "ptsec"
End With
With secColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "sec"
End With
With dt.Columns
    .Add(totalColumn)
    .Add(perColumn)
    .Add(pertotColumn)
    .Add(secColumn)
End With
'*****  end added by RTau 12/12/07   ***************

myCommand = Nothing
myConnection.Close()
myConnection = Nothing
```

```
        Return dt

    End Function



    Private Function GetStopTable(ByVal Feeder As String) As DataTable

        Dim sql As String = "SELECT * FROM [Rerank_Joints] WHERE [Feeder] = '" + Feeder +
"' AND [Stop] = 'stop'"

        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

        Dim myCommand As New SqlDataAdapter(sql, myConnection)

        'takes 30 secs to do!

        myCommand.SelectCommand.CommandTimeout = 120

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        myCommand = Nothing
```

```
        myConnection.Close()

        myConnection = Nothing



        Return dt

    End Function



    Private Function GetRunJointsTable(ByVal Feeder As String) As DataTable

        If (Feeder.Length < 5) Then

            Feeder = "0" + Feeder.ToString

        End If



        Dim sql As String = "SELECT distinct wei_id as Joint_ID, Run_ID, (case when [Feeder
Joint Rank] is null then 99999 else [feeder joint rank ] END) as Joint_Rank ,structure, leg, type, config, stop, year
FROM Rerank_Runs_For_Joints WHERE Feeder = '" + Feeder + "' order by Joint_Rank asc"

        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

        Dim myCommand As New SqlDataAdapter(sql, myConnection)

        'takes 30 secs to do!

        myCommand.SelectCommand.CommandTimeout = 120
```

```
        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)









        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing



        Return dt

    End Function



    Private Function GetRunStopTable(ByVal Feeder As String) As DataTable

        If (Feeder.Length < 5) Then

            Feeder = "0" + Feeder.ToString

        End If



        'modified on 03/05/2009 for including delta mtbf column edisonml
```

Dim sql As String = "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as delta FROM Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

'Dim sql As String = "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints AS Runs WHERE Runs.Feeder = '" + Feeder + "' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

'dim sql as string= "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as delta FROM EdisonML.dbo.Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '4x40' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

'takes 30 secs to do!

```
myCommand.SelectCommand.CommandTimeout = 120

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim totalColumn As DataColumn = New DataColumn
Dim perColumn As DataColumn = New DataColumn
Dim pertotColumn As DataColumn = New DataColumn
Dim secColumn As DataColumn = New DataColumn
With totalColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "tsec"
'    .Expression = "sum(length)"
End With
With perColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "psec"
```

```
End With

With pertotColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "ptsec"

End With

With secColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "sec"

End With

With dt.Columns

   .Add(totalColumn)

   .Add(perColumn)

   .Add(pertotColumn)

   .Add(secColumn)

End With
```

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing


Return dt

End Function


Protected Sub CalculateCostLoad_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostLoad.Click

CheckCommitLoad.Checked = False


CalculateCostLoad_CommitChanges()


If SectionsChangedLoad.Text = "0" Then


  ovruns.Visible = True

  ovsec.Visible = True
```

```
        UpdateAttributesLoad.Visible = False
        Exit Sub
    End If



End Sub
Private Sub CalculateCostLoad_CommitChanges()
    '--modified by shringika on dec 21'2007 included if case
    If cmbSearchFeeder.Items.Count > 0 Then
        BindAttributes()
        BindRankingLPW()
        BindFOT()
        ' Reset_New_Attributes()
        'modification by shringika on 03/11/2008 for LPW tier
        ' BindAttributes1()
        'modification ends here 03/11/2008
        Dim sec_changed As Integer
```

```
Dim dv, dv1 As DataView

'I don't use this strategy now
If StrategyLoadDropDown.SelectedIndex = 1 Then

   dv1 = New DataView(GetsecTable(Feeder.Text, True))

   sec_changed = Replace_Overloaded_Sections(dv1)

End If

If StrategyLoadDropDown.SelectedIndex > 1 Then

   dv = New DataView(GetOvrunTable(Feeder.Text))
   'dv.Sort = "load_percentage DESC"
   'GVovruns.DataSource = dv1
```

```
'GVovruns.DataBind()



'want to detail view to have all sections not just overloaded

dv1 = New DataView(GetsecTable(Feeder.Text, True))

'dv.Sort = "percentage_load DESC"

'GVovsec.DataSource = dv

'GVovsec.DataBind()



Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("OVLRuns")

Dim dt2 As DataTable = New DataTable("OVLSections")

dt1 = dv.ToTable()

dt2 = dv1.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column
```

```
dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("run_id")

Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

GVovlruns.DataSource = dss

GVovlruns.DataMember = dss.Tables(0).TableName()

GVovlruns.DataBind()

'GVpilcruns.RowExpanded(0) = True



Dim m_deltaMTBF As String

m_deltaMTBF =
Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

   GVovlruns.Columns(5).Visible = False

End If
```

```
Dim dv2 As DataView = New DataView(GetOvsecTable(Feeder.Text, False))

'dv2.Sort = "percentage_load DESC"

GVovlnonsec.DataSource = dv2

GVovlnonsec.DataBind()

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVovlruns.Items.Count) - 1

  If Not (Equals(VisitedGlbRuns.Items.FindByText(GVovlruns.Items(visited_run_grid).Cells(2).Text), Nothing)) Then

    GVovlruns.Items(visited_run_grid).ForeColor = Drawing.Color.Red
```

```
            End If

        Next

        sec_changed = Replace_Overloaded_Sections(dv2) + Replace_Overloaded_Runs(dv)

        SectionsChangedLoad.Text = Convert.ToString(sec_changed)

        Total_Cost_Load()

    End If

    UpdateAttributesLoad.Visible = True

    CommitChangesLoad.Visible = True

    CommitChangesLoad.Enabled = True

    ovruns.Visible = True

    ovsec.Visible = True

End If

'included end if for if case on dec 21 2007
```

```
SelectedStrategyLoadPercentage.Text = StrategyLoadDropDown.SelectedValue

loadStrategyInfo.Visible = True

End Sub

Private Function GetSectionInRunTable(ByVal RunID As Integer, ByVal Feeder As String)
As DataTable

Dim sql As String



If (Feeder.Length < 5) Then

  Feeder = "0" + Feeder.ToString

End If




" If PILC Then

' commented to change to edisonml 01/13/2009

sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND Run_id = '" & RunID & " ' AND  isnull(insulation,'UNK') = 'paper'"

'sql  =  "SELECT  *  from  EdisonML.dbo.Rerank_OVL  WHERE  (case  when
left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "'  AND isnull(insulation,'UNK') = 'paper'"

' Else
```

```
' sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND Run_id = '" & RunID & " ' AND isnull(insulation,'UNK') != 'paper'"

'sql = "SELECT * from EdisonML.dbo.Rerank_OVL WHERE (case when
left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND isnull(insulation,'UNK') != 'paper'"

' End If

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)



Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



myCommand = Nothing

myConnection.Close()

myConnection = Nothing



Return dt

End Function
```

```
Protected Sub GVovlruns_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GVovlruns.SelectedIndexChanged

    Dim dt As DataTable

    Dim Run As Integer

    Dim feeder As String

    feeder = cmbSearchFeeder.SelectedValue

    Run = Int32.Parse(GVovlruns.SelectedItem.Cells(2).Text)



    dt = GetSectionInRunTable(Run, feeder)



    Dim struct As String

    struct = dt.Rows(0).Item("from_struct")

    Session("struct") = struct



    Dim URL As String
```

```
URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"

Dim MyScript As New System.Text.StringBuilder
With MyScript
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)
```

```
'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub


Private Sub GVovlruns_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles GVovlruns.TemplateSelection

e.TemplateFilename = "Templates\\OVLSections.ascx"

End Sub


Protected Sub btnovlrunExcel_click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnovlrunExcel.Click


Dim dt As DataTable = GetOvsecTable_EXCEL(Feeder.Text, True)

dt.Merge(GetOvsecTable_EXCEL(Feeder.Text, False))


If dt Is Nothing Then

  Return

End If
```

```
dt.DefaultView.Sort = "Run_Id ASC, load_percentage DESC"

Dim cols As Integer = dt.Columns.Count

'Output excel file

Response.Clear()

Response.ContentType = "application/xls"

Response.AddHeader("content-disposition", "attachment;
filename=""CapitalPlanning_PILC_Rank.xls""")

Response.Write("<table border='1'>")

Dim i As Integer

'Print Row Headings

Response.Write("<tr>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Borough</td>")
```

```
                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Network</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Feeder</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>From Struct</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>To
Struct</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Insulation</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>%
load</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>run id</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>vision id</td>")

                Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Year</td>")

                Response.Write("</tr>")



                'Print Rows
```

```
    For Each row As DataRow In dt.Rows
      Response.Write("<tr>")
      For i = 0 To row.ItemArray.Length() - 1
        Response.Write("<td>" & row(i) & "</td>")
      Next
      Response.Write("</tr>")
    Next
    Response.Write("</table>")
    Response.End()
  End Sub


  Private Sub CalculateCostPILC_CommitChanges()
    visited_sections_count = 0

    '--modified by shringika on dec 21'2007 included if case
    If cmbSearchFeeder.Items.Count > 0 Then
      If RblTotalCostPercentagePILC.SelectedIndex > 0 Then
```

```
    If PercentPILC.Text.Equals("") Or CostPerSectionPILC.Text.Equals("") Then

      Exit Sub



    End If

  Else

    If TotalCostPILC.Text.Equals("") Then

      TargetCostCheck.Visible = True



      Exit Sub



    End If

    If CostPerSectionPILC.Text.Equals("") Then Exit Sub









  End If



  Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder.Text))
```

```
Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))



BindAttributes()

BindRankingLPW()

BindFOT()

'Reset_New_Attributes()

'modification by shringika on 003/11/2008 for LPW tier

' BindAttributes1()

'modification ends here 03/11/2008

' temporary default sort

'dv.Sort = "Run_ID ASC"

'dv1.Sort = "rank DESC"



If StrategyPILCDropDown.SelectedIndex = 1 Then



  'dv.Sort = "Run_ID ASC"

  dv.Sort = "[Feeder PILC Rank] ASC"

  dv1.Sort = "rank ASC"
```

```
    'Replace_paper_percent uses the sort order from above
End If

If StrategyPILCDropDown.SelectedIndex = 2 Then
    'dv.Sort = "Run_ID ASC"
    dv.Sort = "load_percentage DESC"
    dv1.Sort = "percentage_load DESC"
End If

If StrategyPILCDropDown.SelectedIndex = 3 Then
    'dv.Sort = "Run_ID ASC"
    dv.Sort = "LoadXrank DESC"
    dv1.Sort = "LoadXrank DESC"
End If

If StrategyPILCDropDown.SelectedIndex = 4 Then
```

```
StopJointsCalculation()

Runs.Visible = False

GVpilcruns.Visible = False

PILCCablesWithStopJoints.Visible = True

downloadtoExcelPanel.Visible = True

GVPILCStopJoints.Visible = True

UpdateAttributesPILC.Visible = True

CommitChangesPILC.Visible = True

'CommitChangesPILC.Enabled = True

If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"

Exit Sub

End If



'worst stop joint replace logic

If StrategyPILCDropDown.SelectedIndex = 5 Then
```

```
    WorstStopJointsCalculation()

    Runs.Visible = False

    GVpilcruns.Visible = False

    PILCCablesWorstStopJoints.Visible = True

    downloadtoExcelPanel.Visible = True

    GVPILCStopJointTypes.Visible = True

    UpdateAttributesPILC.Visible = True

    CommitChangesPILC.Visible = True

    'CommitChangesPILC.Enabled = True

    If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"

    Exit Sub

End If



'added by shringika for including total cost case on 02/14/2008

If RblTotalCostPercentagePILC.SelectedIndex > 0 Then
```

```
                If CheckCommitPILC.Checked Then

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - duplicate_sections_list)

                    'Updating Section cost text boxes



                Else

                    'SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - visited_sections_count)

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1))

                End If

                Total_Cost_PILC()

            Else



                If CheckCommitPILC.Checked Then

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - duplicate_sections_list)

                Else
```

```
                    'SectionsChangedPILC.Text                                    =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - visited_sections_count)

                    SectionsChangedPILC.Text                                     =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1))

                End If

                Total_Cost_PILC()



            End If

            If SectionsChangedPILC.Text = "0" Then

                actualPercentPILC.Text = "0"

                GVpilcruns.Visible = False

                UpdateAttributesPILC.Visible = False

                downloadtoExcelPanel.Visible = False

                Exit Sub

            End If
```

```
'GVpilcsec.DataSource = dv

'GVpilcsec.DataBind()



'GVpilcruns.DataSource = dv1

'GVpilcruns.DataBind()



' Code for hierarchgrid



Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PICLSections")

dt1 = dv1.ToTable()

dt2 = dv.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)
```

```
Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

Runs.Visible = True

GVpilcruns.Visible = True

UpdateAttributesPILC.Visible = True

CommitChangesPILC.Visible = True

downloadtoExcelPanel.Visible = False

'CommitChangesPILC.Enabled = True

GVpilcruns.DataSource = dss

GVpilcruns.DataMember = dss.Tables(0).TableName()

GVpilcruns.DataBind()
```

```
Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

   GVpilcruns.Columns(10).Visible = False

End If

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVpilcruns.Items.Count) - 1

   If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVpilcruns.Items(visited_run_grid).Cells(2).Text), Nothing)) Then



      GVpilcruns.Items(visited_run_grid).ForeColor = Drawing.Color.Red



   End If
```

```
Next

'GVpilcruns.RowExpanded(0) = True

'lbdebug.Text = dss.Tables(0).TableName()

'CalculateCostPILCAll(StrategyPILCDropDown.SelectedIndex)

End If

'--modification by shringika ends here20-dec-2007
```

```
End Sub

Private Sub WorstStopJointsCalculation()

  Dim dv1 As DataView = New DataView(GetRunsWithWorstStopTable(Feeder.Text))
  Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))
  'runstop.Visible = True
  If RblTotalCostPercentagePILC.SelectedIndex > 0 Then
    If CheckCommitPILC.Checked Then
      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - duplicate_sections_list)
    Else
      'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - visited_sections_count)
      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2))
    End If
    Total_Cost_PILC()
```

```
Else


    If CheckCommitPILC.Checked Then

        SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - duplicate_sections_list)

    Else

        'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - visited_sections_count)

        SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2))

    End If

    Total_Cost_PILC()


End If

If SectionsChangedPILC.Text = "0" Then UpdateAttributesPILC.Visible = False


' Code for hierarchgrid


Dim dss As DataSet = New DataSet()
```

```
Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PILCJoint_Sections")

dt1 = dv1.ToTable()

dt2 = dv2.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("PILCJoint_Sections", dcc1, dcc2, False)

dss.Relations.Add(drr)



'bind the DataSet to the HierarGrid



GVPILCStopJointTypes.DataSource = dss

GVPILCStopJointTypes.DataMember = dss.Tables(0).TableName()
```

```
GVPILCStopJointTypes.DataBind()

Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

  GVPILCStopJointTypes.Columns(5).Visible = False

End If

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVPILCStopJointTypes.Items.Count) - 1

  If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVPILCStopJointTypes.Items(visited_run_grid).Cells(1).Text),
Nothing)) Then


    GVPILCStopJointTypes.Items(visited_run_grid).ForeColor = Drawing.Color.Red



  End If

Next
```

```
        PILCCablesWorstStopJoints.Visible = True

        GVPILCStopJointTypes.Visible = True



    End Sub


    Protected Sub GVpilcruns_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GVpilcruns.SelectedIndexChanged

        Dim dt As DataTable

        Dim Run As Integer

        Dim feeder As String

        feeder = cmbSearchFeeder.SelectedValue

        Run = Int32.Parse(GVpilcruns.SelectedItem.Cells(2).Text)



        dt = GetSectionInRunTable(Run, feeder)
```

```
Dim struct As String

struct = dt.Rows(0).Item("from_struct")

Session("struct") = struct



Dim URL As String



URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"



Dim MyScript1 As New System.Text.StringBuilder

With MyScript1

   .Append("<script type='text/javascript'>")

   .Append("window.open('" & URL & "');")

   .Append("</script>")

End With
```

```
'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript1.ToString, False)

'RegisterStartupScript(Me, Me.GetType(), "FeederMap1", MyScript1.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub

Private Sub GVpilcruns_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles GVpilcruns.TemplateSelection

e.TemplateFilename = "Templates\\PILCSections.ascx"

End Sub

Private Sub CalculateCostPILC_SelectRuns()
```

```
Visited_sections_count = 0

'--modified by shringika on dec 21'2007 included if case
If cmbSearchFeeder.Items.Count > 0 Then

  Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder.Text))
  Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))

  BindAttributes()
  BindRankingLPW()
  BindFOT()
  If StrategyPILCDropDown.SelectedIndex = 1 Then

    'dv.Sort = "Run_ID ASC"
    dv.Sort = "[Feeder PILC Rank] ASC"
    dv1.Sort = "rank ASC"

    'dv1.Sort = "is_basket DESC, is_spur ASC, delta DESC"
```

```
    dv1.Sort = "is_basket DESC, is_spur ASC, rank ASC"

    ViewState("sortdirection") = "ASC"

    'Replace_paper_percent uses the sort order from above

End If



If StrategyPILCDropDown.SelectedIndex = 2 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "load_percentage DESC"

    dv1.Sort = "percentage_load DESC"

    dv1.Sort = "is_basket DESC, is_spur ASC, percentage_load DESC"

    ViewState("sortdirection") = "DESC"

End If



If StrategyPILCDropDown.SelectedIndex = 3 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "LoadXrank DESC"

    dv1.Sort = "LoadXrank DESC"

    dv1.Sort = "is_basket DESC, is_spur ASC, LoadXrank DESC"
```

```
    ViewState("sortdirection") = "DESC"
End If



'If StrategyPILCDropDown.SelectedIndex = 4 Then


'   StopJointsCalculation()
'   Runs.Visible = False
'   GVpilcruns.Visible = False
'   PILCCablesWithStopJoints.Visible = True
'   downloadtoExcelPanel.Visible = True
'   GVPILCStopJoints.Visible = True
'   UpdateAttributesPILC.Visible = True
'   CommitChangesPILC.Visible = True
'   'CommitChangesPILC.Enabled = True
'   If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"
'   Exit Sub
```

```
'End If



"worst stop joint replace logic

'If StrategyPILCDropDown.SelectedIndex = 5 Then



'   WorstStopJointsCalculation()

'   Runs.Visible = False

'   GVpilcruns.Visible = False

'   PILCCablesWorstStopJoints.Visible = True

'   downloadtoExcelPanel.Visible = True

'   GVPILCStopJointTypes.Visible = True

'   UpdateAttributesPILC.Visible = True

'   CommitChangesPILC.Visible = True

'   'CommitChangesPILC.Enabled = True

'   If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"

'   Exit Sub

'End If
```

```
'added by shringika for including total cost case on 02/14/2008

" " "If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

" " "   If CheckCommitPILC.Checked Then

"     "     "                                   SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - Duplicate_sections_list)

" " "     'Updating Section cost text boxes



" " "   Else

"     "     "                                   'SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - visited_sections_count)

"     "     "                                   SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1))

" " "   End If

" " "   Total_Cost_PILC()

" " "Else
```

```
" " "   If CheckCommitPILC.Checked Then

"     "     "                                   SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - Duplicate_sections_list)

" " "   Else

"     "     "                                   'SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - visited_sections_count)

"     "     "                                   SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1))

" " "   End If

" " "   Total_Cost_PILC()



" " "End If

" " "If SectionsChangedPILC.Text = "0" Then

" " "   actualPercentPILC.Text = "0"

" " "   GVpilcruns.Visible = False

" " "   UpdateAttributesPILC.Visible = False

" " "   downloadtoExcelPanel.Visible = False
```

```
""" Exit Sub

""" "End If



'GVpilcsec.DataSource = dv

'GVpilcsec.DataBind()



'GVpilcruns.DataSource = dv1

'GVpilcruns.DataBind()



' Code for hierarchgrid
```

```
Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PICLSections")

dt1 = dv1.ToTable()

dt2 = dv.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

Runs_Select.Visible = True

GVPILCRuns_Select.Visible = True

UpdateAttributesPILC.Visible = False
```

```
'CommitChangesPILC.Visible = False

downloadtoExcelPanel.Visible = False

'CommitChangesPILC.Enabled = True

GVPILCRuns_Select.DataSource = dss

GVPILCRuns_Select.DataMember = dss.Tables(0).TableName()

GVPILCRuns_Select.DataBind()

GVpilcruns.Visible = False

Runs.Visible = False



Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") And Not RblTotalCostPercentagePILC.SelectedIndex =
2 Then

   GVPILCRuns_Select.Columns(7).Visible = False

End If

Dim visited_run_grid As Integer
```

```
For visited_run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1

If Not (Equals(VisitedGlbRuns.Items.FindByText(GVPILCRuns_Select.Items(visited_run_grid).Cells(2).Text), Nothing)) Then

GVPILCRuns_Select.Items(visited_run_grid).ForeColor = Drawing.Color.Red

End If

Next

'GVpilcruns.RowExpanded(0) = True
```

```
'lbdebug.Text = dss.Tables(0).TableName()



'CalculateCostPILCAll(StrategyPILCDropDown.SelectedIndex)









    End If

    '--modification by shringika ends here20-dec-2007

End Sub



Private Sub CalculateCostPILC_SelectRunsCommitChanges()



    'Dim run_grid As Integer

    'Dim chk1 As CheckBox

    Dim count As Integer
```

```
count = 0

'For run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1



'   chk1 = GVPILCRuns_Select.Items(run_grid).Cells(7).FindControl("chkselectrun")

'   If chk1.Checked Then

'      count += 1

'      TextBox3.Text = count.ToString

'      Dim MyScript1 As New System.Text.StringBuilder

'      With MyScript1

'         .Append("<script type='text/javascript'>")

'         .Append("alert ('checkbox checked');")

'         .Append("</script>")

'      End With

'                        Page.ClientScript.RegisterStartupScript(GetType(String),    "check",
MyScript1.ToString)



'      'Else

'      '   Dim MyScript1 As New System.Text.StringBuilder
```

```
'        ' With MyScript1

'        '     .Append("<script type='text/javascript'>")

'        '     .Append("alert ('no checkbox');")

'        '     .Append("</script>")

'        ' End With

'                ' Page.ClientScript.RegisterStartupScript(GetType(String), "check",
MyScript1.ToString)



'  End If



'Next

Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))

Dim m_deltaMTBF As String

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);

'm_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))
```

```
'If Equals(m_deltaMTBF, "no") Then

'    GVPILCRuns_Select.Columns(7).Visible = False

'End If



    SectionsChangedPILC.Text = Replace_PILCRunsSelect(dv1).ToString

    Total_Cost_PILC()

End Sub

Private Function Replace_PILCRunsSelect(ByVal runs As DataView) As Integer



    Dim total_sections, pilc_sections, running_total As Integer

    Dim runId As Integer

    Dim run_grid As Integer



    total_sections = Int32.Parse(Sec_CNT.Text)

    Dim select_runs As ArrayList

    select_runs = New ArrayList

    select_runs.Add(0)

    running_total = 0
```

```
Dim chk2 As CheckBox

If (runs.Table.Rows.Count > 0) Then

  pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))



  'running_percent = 0

  'running_total = 0

  Dim run As String



  For run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1



    run = GVPILCRuns_Select.Items(run_grid).Cells(2).Text



    chk2 = GVPILCRuns_Select.Items(run_grid).Cells(7).FindControl("chkselectrun")

    If chk2.Checked Then

      If Not Equals(VisitedGlbRuns.Items.FindByText(run.ToString), Nothing) Then

        Continue For
```

```
        Else
            select_runs.Add(run.ToString)
            If CheckCommitPILC.Checked Then
                Update_Attributes(run, False)
                VisitedGlbRuns.Items.Add(run)
            End If

        End If
    End If

Next

Dim j, k As Integer
For j = 1 To select_runs.Count - 1
```

```
For k = 0 To runs.Table.Rows.Count - 1

runId = runs.Item(k).Item("Run_ID")

If Not Equals(select_runs(j).ToString, runId.ToString) Then

Continue For

Else

running_total = running_total + runs.Item(k).Item("length")

End If

Next

Next

'commented on 02/18/2010

'  running_percent = running_total / pilc_sections * 100.0

'  running_tot_percent = running_total / total_sections * 100.0

'  If CheckCommitPILC.Checked Then
```

```
'      Update_Attributes(runId, False)

'          If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)

'  End If

'  visited_sections_check(runId, False)



'   'end of modification by shringika on 06/16/2008 for calling update attributes only
when commit changes box is checked

'  actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))




Else

'  'when no rows returned from table



  actualPercentPILC.Text = 0

End If

actualPercentPILC.Text = Convert.ToString(Math.Round(running_total / pilc_sections *
100))
```

```
'ovl_cnt = running_total

'Return ovl_cnt

Return running_total



End Function



Protected Sub CalculateCostRel_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostRel.Click

'--modified by shringika on 20-dec-2007 included if case

' /this one is for xlp tier

CheckCommitRel.Checked = False

CalculateCostRel_CommitChanges()



End Sub

Private Sub CalculateCostRel_CommitChanges()

visited_sections_count = 0

If cmbSearchFeeder.Items.Count > 0 Then
```

```
BindAttributes()

BindRankingLPW()

BindFOT()

' Reset_New_Attributes()

UpdateAttributesRel.Visible = True

CommitChangesRel.Visible = True

XLP.Visible = True

GridViewXLP.Visible = True

Dim MenuChoice As Integer

'added on 01/21/2009 for sectionschanged issue in xlp- dt1 and sectionschanged

Dim dt1 As DataTable

Dim sectionschanged As Integer



MenuChoice = StrategyRelDropDown.SelectedIndex



Select Case MenuChoice

  Case 1 To 3

    'XLP  modified by RTau 1/17/08
```

```
Dim dv1 As DataView = New DataView(GetXLPTable(Feeder.Text))

'added on 01/21/2009 for sectionschanged issue in xlp- dt1 and sectionschanged

dt1 = GetXLPTable(Feeder.Text)

sectionschanged = dt1.Rows.Count

Select Case MenuChoice

  Case 1

    'Sort by Cable rank

    dv1.Sort = "Feeder_Cable_RANK ASC"

  Case 2

    'Sort by Load

    dv1.Sort = "load_percentage DESC"

  Case 3

    'Sort by LoadxRank

    dv1.Sort = "LoadXrank DESC"

End Select
```

```
                If CheckCommitRel.Checked Then

                    SectionsChangedRel.Text =
Convert.ToString(Replace_XLP_percent(RelPercent.Text, dv1) - duplicate_sections_list)

                Else

                    'SectionsChangedRel.Text =
Convert.ToString(Replace_XLP_percent(RelPercent.Text, dv1) - visited_sections_count)

                    SectionsChangedRel.Text =
Convert.ToString(Replace_XLP_percent(RelPercent.Text, dv1))

                End If



                If SectionsChangedRel.Text = "0" Then

                    UpdateAttributesRel.Visible = False

                    'XLP.Visible = False

                End If



                CommitChangesRel.Visible = True

                Total_Cost_Rel()



                ' Code for GridView
```

```
            GridViewXLP.DataSource = dv1

            GridViewXLP.DataBind()



        End Select



        Dim visited_run_grid As Integer

        For visited_run_grid = 0 To (GridViewXLP.Rows.Count) - 1

          If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GridViewXLP.Rows(visited_run_grid).Cells(0).Text), Nothing)) Then

            GridViewXLP.Rows(visited_run_grid).ForeColor = Drawing.Color.Red

          End If









        Next

        'added on 01/21/2009 for sectionschanged issue in xlp

        SectionsChangedRel.Text = sectionschanged
```

```
End If

'end if added on 21-dec-2007 by shringika



End Sub



Protected Sub StrategyDropDownLoad_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles StrategyLoadDropDown.SelectedIndexChanged

'--modified by shringika on 21-dec-2007 included if case

If cmbSearchFeeder.Items.Count > 0 Then


  BindAttributes()

  BindRankingLPW()
```

```
BindFOT()

'Reset_New_Attributes()

'added by shringika on 03/11/2008 for LPW tier

'BindAttributes1()



'modification ends here 03/11/2008



If StrategyLoadDropDown.SelectedIndex = 1 Then



   ovsec.Visible = True

   ovruns.Visible = True



End If



If StrategyLoadDropDown.SelectedIndex = 2 Then



   ovsec.Visible = True
```

```
                ovruns.Visible = True


                End If

            End If

            'iend if added by shringika on 21-dec-2007

        End Sub


        Protected Sub StrategyDropDownPILC_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles StrategyPILCDropDown.SelectedIndexChanged

            TargetCostCheck.Visible = False

            actualTotalCostPILC.Text = ""

            actualPercentPILC.Text = ""

            SectionsChangedPILC.Text = ""

            TotalCostPILC.Text = ""

            RblTotalCostPercentagePILC.Items(0).Enabled = True

            downloadtoExcelPanel.Visible = False

            RblTotalCostPercentagePILC.SelectedIndex = 0
```

```
Runs.Visible = False

Runs_Select.Visible = False

PILCCablesWithStopJoints.Visible = False

UpdateAttributesPILC.Visible = False

CommitChangesPILC.Visible = False

PILCCablesWorstStopJoints.Visible = False

downloadtoExcelPanel.Visible = False

PercentPILC.SelectedIndex = 0

Reset_MTBF_Load()

Reset_MTBF_PILC()

Reset_MTBF_Rel()



If StrategyPILCDropDown.SelectedIndex = 4 Then
```

```
' RblTotalCostPercentagePILC.Items(0).Enabled = False

'RblTotalCostPercentagePILC.Items(1).Selected = True



'TotalCostPILC.Enabled = False

'PercentPILC.Enabled = True



Runs.Visible = False

downloadtoExcelPanel.Visible = False

PILCCablesWithStopJoints.Visible = False

UpdateAttributesPILC.Visible = False

CommitChangesPILC.Visible = True



PercentPILC.SelectedIndex = 0









End If
```

```
End Sub

Protected Sub StrategyDropDownRel_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles StrategyRelDropDown.SelectedIndexChanged

BindAttributes()

BindRankingLPW()

BindFOT()

UpdateAttributesRel.Visible = False



RelXLPCalculationReset()




' Reset_New_Attributes()

'added by shringika on 03/11/2008 for LPW tier

'BindAttributes1()

'modification ends here 03/11/2008

If StrategyRelDropDown.SelectedIndex = 1 Then
```

```
'elastimolds



'ELAS.Visible = True

XLP.Visible = False

'runstop.Visible = False

'may need ot have its own view...

' runstop.Visible = True




End If


If StrategyRelDropDown.SelectedIndex = 2 Then


'Replace_Stops()


'ELAS.Visible = False
```

```
                XLP.Visible = False

                'runstop.Visible = True




            End If

            If (StrategyRelDropDown.SelectedIndex >= 3) And (StrategyRelDropDown.SelectedIndex
<= 5) Then

                'Replace_XLP_percent()


                'ELAS.Visible = False

                XLP.Visible = True

                ' runstop.Visible = False



            End If
```

```
End Sub

Private Function Replace_Overloaded_Sections(ByVal secs As DataView) As Integer

  Dim i As Integer

  Dim ovl_cnt As Int32

  Dim Id As Integer

  ovl_cnt = 0

  If secs.Table.Rows.Count = 0 Then

    Return 0

  End If
```

```
Dim dr As DataRow = secs.Table.Rows(0)

'   ovl_cnt = Convert.ToString(Int32.Parse(drcheck(dr, "ovl_cnt")))



SectionsChangedLoad.Text = Convert.ToString(ovl_cnt)

Total_Cost_Load()



For i = 0 To (secs.Table.Rows.Count - 1)

   Id = secs.Table.Rows.Item(i).Item("vision_id")



   Dim visited_visionIds As ArrayList

   If Session("visited_sections_nonPILC") Is Nothing Then

      visited_visionIds = New ArrayList

   Else

      visited_visionIds = CType(Session("visited_sections_nonPILC"), ArrayList)

   End If
```

```
If visited_visionIds.Count < 1 Then visited_visionIds.Add(0)



If visited_visionIds.Contains(Id) Then Continue For

ovl_cnt += 1

If CheckCommitLoad.Checked Then

   visited_visionIds.Add(Id)

   Session("visited_sections_nonPILC") = visited_visionIds



   Update_Attributes(Id, True)

End If

Next



'vs2220: Commented out for centralized filling of SummarySections

'If Not (Equals(SectionsSummary.Items.FindByText("No Sections"), Nothing)) Then
SectionsSummary.Items.Remove("No Sections")

'added by shringika on 07/29/2008 to fix populating secions changed details on commiting
changes again

'If CheckCommitLoad.Checked Then Return ovl_cnt
```

```
'For i = 0 To (secs.Table.Rows.Count - 1)

'Id = secs.Table.Rows.Item(i).Item("vision_id")

'SectionsSummary.Items.Add(Convert.ToString(Id))



'Next



Return ovl_cnt

End Function

Private Function Replace_Overloaded_Runs(ByVal runs As DataView) As Integer



Dim ovl_cnt As Int32

Dim i As Integer

Dim runId As Integer

'coomented by shringika on 01/19/2009 for getting total no. of sections changed on 1st tier

ovl_cnt = 0
```

```
If (runs.Table.Rows.Count > 0) Then

  'commented on 01/19/2009 , included this tatement within loop with i



  'ovl_cnt = ovl_cnt + Int32.Parse(runs.Item(0).Item("ovl_cnt"))



  For i = 0 To (runs.Table.Rows.Count - 1)

    runId = runs.Item(i).Item("Run_ID")

    If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For

    ovl_cnt = ovl_cnt + Int32.Parse(runs.Item(i).Item("ovl_cnt"))

    ' Update_Attributes(runId, False)

    If CheckCommitLoad.Checked Then

      Update_Attributes(runId, False)

      VisitedGlbRuns.Items.Add(runId)



    End If
```

```
    Next

  Else : Return 0



  End If



  Return ovl_cnt




End Function

'not to be used anymore -- agb 7/30/08

Private Function Replace_Paper() As Integer

  If CheckCommitPILC.Checked Then

    mon_Paper_GT40.Text = "0"

    mon_Paper_LE40.Text = "0"

    mon_Solid_LT16.Text = Convert.ToString(Int32.Parse(Solid_LT16.Text) +
Int32.Parse(Paper_LE40.Text) + Int32.Parse(Paper_GT40.Text))

    mon_Percent_Paper_GT40.Text = "0"

    mon_Percent_Paper_LE40.Text = "0"
```

```
mon_Percent_Solid_LT16.Text = Convert.ToString((Int32.Parse(Solid_LT16.Text) +
Int32.Parse(Paper_LE40.Text) + Int32.Parse(Paper_GT40.Text)) / Int32.Parse(Sec_CNT.Text) * 100.0)

mon_AllPaper.Text = "0"

mon_Percent_AllPaper.Text = "0"

mon_E21GT93.Text = "0"

mon_E21LE93.Text = "0"

mon_R31GT94.Text = "0"

mon_R31LE93.Text = "0"

mon_OtherStop_LT7.Text = "0"

mon_OtherStop7_12.Text = "0"

mon_OtherStop_GT12.Text = "0"

mon_AllStop.Text = "0"

mon_Percent_ALLStop.Text = "0"

mon_PJ_LT8.Text = "0"

mon_PJ8_16.Text = "0"

mon_PJ16_24.Text = "0"

' mon_PJ_GT24.Text = "0"

mon_PJ24_32.Text = "0"
```

```
mon_PJ32_40.Text = "0"

mon_PJGT40.Text = "0"

mon_SJ_LT8.Text = Convert.ToString(Int32.Parse(SJ_LT8.Text) +
Int32.Parse(AllStop.Text) + Int32.Parse(AllPaper.Text))

mon_Percent_E21GT93.Text = "0"

mon_Percent_E21LE93.Text = "0"

mon_Percent_R31GT94.Text = "0"

mon_Percent_R31LE93.Text = "0"

mon_Percent_OtherStop_LT7.Text = "0"

mon_Percent_OtherStop7_12.Text = "0"

mon_Percent_OtherStop_GT12.Text = "0"

mon_Percent_PJ_LT8.Text = "0"

mon_Percent_PJ8_16.Text = "0"

mon_Percent_PJ16_24.Text = "0"

'mon_Percent_PJ_GT24.Text = "0"

mon_Percent_PJ24_32.Text = "0"

mon_Percent_PJ32_40.Text = "0"

mon_Percent_PJGT40.Text = "0"
```

```
                mon_Percent_SJ_LT8.Text =
Convert.ToString(Convert.ToDouble(Percent_SJ_LT8.Text) + Convert.ToDouble(Percent_ALLStop.Text) +
Convert.ToDouble(Percent_AllPaper.Text))

                mon_AllSolid.Text = Convert.ToString(Int32.Parse(mon_AllSolid.Text) +
Int32.Parse(AllStop.Text) + Int32.Parse(AllPaper.Text))

                mon_Percent_AllSolid.Text =
Convert.ToString(Convert.ToDouble(Percent_AllSolid.Text) + Convert.ToDouble(Percent_ALLStop.Text) +
Convert.ToDouble(Percent_AllPaper.Text))

                'Might want to move this to a separate sub

                'SectionsChangedPILC.Text = Convert.ToString(Int32.Parse(Paper_LE40.Text) +
Int32.Parse(Paper_GT40.Text))

                'Total_Cost_PILC()

            End If

            Return Int32.Parse(Paper_LE40.Text) + Int32.Parse(Paper_GT40.Text)

        End Function

        Private Function Replace_Paper_percent(ByVal percent As String, ByVal runs As
DataView) As Integer
```

```
Dim ovl_cnt As Int32



Dim i As Integer

Dim total_sections, pilc_sections, running_total As Integer

Dim per As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer



'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)

'If percent.Length = 0 Then percent = "100"

per = Int32.Parse(percent)



'If (per = 100) Then

'section count is from attributes in function so can be different, bummer!

'Replace_Paper()

'changed by shringika and made ==100 for 100 percent
```

```
'actualPercentPILC.Text = "100"

'If (runs.Table.Rows.Count > 0) Then



'Return Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'Else

'actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

'Return 0

'End If

'End If




If (runs.Table.Rows.Count > 0) Then

   pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))



   'running_percent = 0

   'running_total = 0
```

```
For i = 0 To (runs.Table.Rows.Count - 1)

  runId = runs.Item(i).Item("Run_ID")

  'modified on 08/15/2008 for eliminATING ACCUMULATED PERCENT CALCULATION FOR VISITED RUNS

  If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then Continue For

  running_total = running_total + runs.Item(i).Item("length")

  running_percent = running_total / pilc_sections * 100.0

  running_tot_percent = running_total / total_sections * 100.0

  If (running_percent > per) Then

    running_total = running_total - runs.Item(i).Item("length")
```

```
running_percent = running_total / pilc_sections * 100.0

running_tot_percent = running_total / total_sections * 100.0

runId = runs.Item(i).Item("Run_ID")



actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

Exit For

End If



runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

'shringika, add commit logic here and other places this is called

If CheckCommitPILC.Checked Then



Update_Attributes(runId, False)

If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)
```

```
                End If

                visited_sections_check(runId, False)



                'end of modification by shringika on 06/16/2008 for calling update attributes only when commit changes box is checked

                actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))



            Next

        Else

            'when no rows returned from table



            actualPercentPILC.Text = 0

        End If



        ovl_cnt = running_total

        Return ovl_cnt
```

```
End Function

Private Function Replace_Paper_TotalCost(ByVal totalcost As String, ByVal runs As
DataView) As Integer

    Dim ovl_cnt As Int32

    Dim i As Integer

    Dim total_sections, pilc_sections, running_total As Integer

    Dim total_cost As Integer

    Dim running_percent, running_tot_percent As Decimal

    Dim runId As Integer

    Dim cost_per_section As Integer

    'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))

    total_sections = Int32.Parse(Sec_CNT.Text)

    total_cost = Int32.Parse(totalcost)
```

```
cost_per_section = Int32.Parse(CostPerSectionPILC.Text)

Dim total_replaced_sections As Integer
total_replaced_sections = Math.Floor(total_cost / cost_per_section)

If (runs.Table.Rows.Count > 0) Then
   pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))
   lblPILCTargetPercent.Text = pilc_sections
   For i = 0 To (runs.Table.Rows.Count - 1)

      runId = runs.Item(i).Item("Run_ID")
      If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For
      running_total = running_total + runs.Item(i).Item("length")
```

```
If (running_total > total_replaced_sections) Then
   running_total = running_total - runs.Item(i).Item("length")
   actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))
   Exit For
End If

running_percent = running_total / pilc_sections * 100.0
running_tot_percent = running_total / total_sections * 100.0

runs.Item(i).Item("tsec") = running_total
runs.Item(i).Item("psec") = Math.Round(running_percent)
runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)
'modified on 08/13/08 for visited section elimination
If CheckCommitPILC.Checked Then
   If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)
```

```
            Update_Attributes(runId, False)

        End If

        visited_sections_check(runId, False)



        actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

        If (running_total.Equals(total_replaced_sections)) Then Exit For



    Next

End If



ovl_cnt = running_total

'actualPercentPILC.Text = Math.Round(running_total / pilc_sections) * 100.0



Return ovl_cnt



'Might want to move this to a separate sub

'SectionsChangedPILC.Text = Convert.ToString(ovl_cnt)
```

```
'Total_Cost_PILC()

End Function

' end of function added by shringika on 02/14/2008



Private Function Replace_Elastimold_percent(ByVal percent As String, ByVal runs As DataView, ByVal joints As DataView) As Integer

'*** added by RTau 12/11

Dim ovl_cnt As Int32



Dim i, j As Integer

Dim total_joints, total_sections, total_estops As Integer

Dim running_total, running_sec_total As Integer

Dim per As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer

Dim visitedRuns As New ArrayList

Dim visitedJoints As New ArrayList
```

```
Dim dupcount As Integer



total_sections = Int32.Parse(Sec_CNT.Text)

'If percent.Length = 0 Then percent = "100"

per = Int32.Parse(percent)

running_sec_total = 0



If (runs.Table.Rows.Count > 0) Then

   total_estops = Int32.Parse(mon_E21LE93.Text) + Int32.Parse(mon_E21GT93.Text)

   " If total_estops = 0 Then Return 0

   total_joints = Int32.Parse(mon_Joint_CNT.Text)

   running_total = 0



   For i = 0 To (runs.Table.Rows.Count - 1)

      runId = runs.Item(i).Item("Run_ID")

      If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For
```

```
dupcount = 0

'Joints can be in more than one run! Have to check that

'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND type =
'ELASTIMOLD' AND config = '2W-1W' AND stop = 'S'"

For j = 0 To (joints.Count - 1)

  If visitedJoints.Contains(joints(j)("Joint_ID")) Then

    dupcount = dupcount + 1

  Else

    visitedJoints.Add(joints(j)("Joint_ID"))

  End If



Next



joints.RowFilter = Nothing



running_total = running_total + runs.Item(i).Item("Stops") - dupcount

running_sec_total = running_sec_total + runs.Item(i).Item("length")
```

```
running_percent = (running_total / total_estops) * 100.0

running_tot_percent = (running_total / total_joints) * 100.0



If (running_percent > per) Then

   running_total = running_total - runs.Item(i).Item("length")

   running_percent = (running_total / total_estops) * 100.0

   running_tot_percent = running_total / total_joints * 100.0

   runId = runs.Item(i).Item("Run_ID")



   'actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

   Exit For

End If



runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)
```

```
runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

'I dont think this check does anything! agb

If (Not (visitedRuns.Contains(runId))) Then

  visitedRuns.Add(runId)



  If CheckCommitElast.Checked Then



    Update_Attributes(runId, False)

    If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)

  End If



End If



visited_sections_check(runId, False)
```

```
        ' If (running_percent >= per) Then Exit For


      Next

    End If


    ovl_cnt = running_sec_total


    Return ovl_cnt


End Function
```

```
Private Function Replace_Stops_percent(ByVal percent As String, ByVal runs As DataView,
ByVal joints As DataView) As Integer

    Dim ovl_cnt As Int32

    Dim i, j As Integer

    Dim total_joints, total_sections, total_stops As Integer

    Dim running_total, running_sec_total As Integer

    Dim per As Integer

    Dim running_percent, running_tot_percent As Decimal

    Dim runId As Integer

    Dim visitedRuns As New ArrayList

    Dim visitedJoints As New ArrayList

    Dim dupcount As Integer
```

```
'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)



per = Int32.Parse(percent)

running_sec_total = 0



'If (per = 100) Then

'section count is from attributes in funtion so can be different, bummer!

'sec_changed = Replace_Paper()

'If (runs.Table.Rows.Count > 0) Then

'Return Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'Return sec_changed

'Else : Return 0

'End If

'End If



If (runs.Table.Rows.Count > 0) Then
```

```
'pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'for now do it this way

'get this form base not running AGB

total_stops = Int32.Parse(AllStop.Text)

total_joints = Int32.Parse(Joint_CNT.Text)

running_total = 0



For i = 0 To (runs.Table.Rows.Count - 1)

  runId = runs.Item(i).Item("Run_ID")

  If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For

  dupcount = 0

  'Joints can be in more than one run! Have to check that

  'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

  joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND stop = 'S'"

  For j = 0 To joints.Count - 1

    If visitedJoints.Contains(joints(j)("Joint_ID")) Then
```

```
            dupcount = dupcount + 1
        Else
            visitedJoints.Add(joints(j)("Joint_ID"))
        End If

    Next

    joints.RowFilter = Nothing
    running_total = running_total + runs.Item(i).Item("Stops") - dupcount
    running_sec_total = running_sec_total + runs.Item(i).Item("length")
    running_percent = running_total / total_stops * 100.0
    running_tot_percent = running_total / total_joints * 100.0

    'do this early now -- agb
    ' If (running_percent >= per) Then Exit For
```

```
If (running_percent >= per) Then

    running_sec_total = running_sec_total - runs.Item(i).Item("length")

    running_total = running_total - (runs.Item(i).Item("Stops") - dupcount)

    running_percent = running_total / total_stops * 100.0

    running_tot_percent = running_total / total_joints * 100.0

    runId = runs.Item(i).Item("Run_ID")



    ' actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

    Exit For


End If
```

```
runId = runs.Item(i).Item("Run_ID")
```

```
runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

'use this to debug dups

'runs.Item(i).Item("ptsec") = dupcount



'I dont think this check does anything... AGB

If (Not (visitedRuns.Contains(runId))) Then

   visitedRuns.Add(runId)

   If CheckCommitStops.Checked Then

      Update_Attributes(runId, False)

      If  Equals(VisitedGlbRuns.Items.FindByText(runId.ToString),  Nothing)  Then
VisitedGlbRuns.Items.Add(runId)

   End If

   visited_sections_check(runId, False)
```

```
End If

'If (running_percent.Equals(per)) Then Exit For

'If (running_percent >= per) Then Exit For

  Next

End If

ovl_cnt = running_sec_total
```

```
    Return ovl_cnt

End Function

Private Function Replace_XLP_percent(ByVal percent As String, ByVal runs As DataView)
As Integer

    Dim i As Integer
    Dim total_sections, XLP_sections, running_total As Integer
    Dim per As Integer
    Dim running_percent, running_tot_percent As Decimal
    Dim runId As Integer

    total_sections = Int32.Parse(Sec_CNT.Text)
    'If percent.Length = 0 Then percent = "100"
```

```
per = Int32.Parse(percent)

' Sum total length of XLP sections from individual sections
For i = 0 To (runs.Table.Rows.Count - 1)
   XLP_sections = XLP_sections + runs.Item(i).Item("length")

Next

If (runs.Table.Rows.Count > 0) Then
   For i = 0 To (runs.Table.Rows.Count - 1)
      runId = runs.Item(i).Item("Run_ID")

      If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For

      running_total = running_total + runs.Item(i).Item("length")
      running_percent = running_total / XLP_sections * 100.0
      running_tot_percent = running_total / total_sections * 100.0
```

```
If (running_percent > per) Then
   running_total = running_total - runs.Item(i).Item("length")
   running_percent = running_total / XLP_sections * 100.0
   running_tot_percent = running_total / total_sections * 100.0
   runId = runs.Item(i).Item("Run_ID")

   actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))
   Exit For
End If

runs.Item(i).Item("tsec") = running_total
runs.Item(i).Item("psec") = Math.Round(running_percent)
```

```
                runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

                If CheckCommitRel.Checked Then

                  Update_Attributes(runId, False)

                  If  Equals(VisitedGlbRuns.Items.FindByText(runId.ToString),  Nothing)  Then
VisitedGlbRuns.Items.Add(runId)

                End If

                visited_sections_check(runId, False)



              Next

            End If



            Return running_total



          End Function
```

```
Private Function GetCblCnt(ByVal Feeder As String, ByVal Str As String) As Integer

'AGB added this to emulate Vaibhav's logic in update_attributes stroed proc

Dim sql As String

sql = "SELECT cableCount = count(*) FROM EdisonML.dbo.Rerank_OVL WHERE (from_struct = '" + Str + "' OR to_struct = '" + Str + "') and feeder= '" + Feeder + "' AND 1 = (SELECT count(*) FROM EdisonML.dbo.Rerank_OVL WHERE (from_struct = '" + Str + "' or to_struct = '" + Str + "') AND insulation = 'paper' and feeder= '" + Feeder + "' )"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Dim dr As DataRow = dt.Rows(0)
```

```
    Return Int32.Parse(dr("cableCount"))

End Function



Private Sub Update_Attributes(ByVal id As String, ByVal isSection As Boolean)

    Dim m_negativeValues As String

    m_negativeValues =
Convert.ToString(ConfigurationManager.AppSettings("negativeAttributes"))

    'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



    If Equals(m_negativeValues, "no") Then



      Update_Attributes_Fix_For_Negative_Attribute_Values(id, isSection)

    Else

      Update_Attributes__Old_With_Negative_Values(id, isSection)

    End If

End Sub

Private Sub Update_Attributes__Old_With_Negative_Values(ByVal id As String, ByVal
isSection As Boolean)
```

```
Dim totalJoints As Integer

Dim dr As DataRow

Dim fromtostructs As DataTable

'  Static Uattr As UpdateAttributes = New UpdateAttributes

Dim feeder_id As String

feeder_id = cmbSearchFeeder.SelectedValue

If feeder_id.Length < 5 Then feeder_id = "0" + feeder_id.ToString




If (isSection = False) Then

   'this function is more generic than the nam ovl lets on to

   'attributes = upd.GetAttributeVectors(id, False, feeder_id)



   dr = Uattr.CountMe(id, False)

   fromtostructs = Uattr.GetFromToStruct(id, False)

Else
```

```
'attributes = upd.GetAttributeVectors(id, True, feeder_id)
'
dr = Uattr.CountMe(id, True)
fromtostructs = Uattr.GetFromToStruct(id, True)

End If

Dim k As Integer
Dim dtr As DataRow

'commented on 08/16/2009 for new update attributes code

For k = 0 To (fromtostructs.Rows.Count - 1)
dtr = fromtostructs.Rows(k)
If IsDBNull(dtr("feeder")) = True And IsDBNull(dtr("from_struct")) = True And IsDBNull(dtr("to_struct")) = True Then
'Exit Sub
Continue For
```

```
                End If


                If (Equals(SectionsSummary.Items.FindByText(dtr("feeder") + "." + dtr("from_struct")
+ "." + dtr("to_struct")), Nothing)) Then

                    SectionsSummary.Items.Add(dtr("feeder") + "." + dtr("from_struct") + "." +
dtr("to_struct"))

                    sections_area.InnerText += System.Environment.NewLine + dtr("feeder") + "." +
dtr("from_struct") + "." + dtr("to_struct")

                    'Else

                    'commented on 01/27/2009

                    'duplicate_sections_list = duplicate_sections_list + 1

                    ' duplicate_sections_list = 0


                End If

            Next


            'mean imputation stats fon canonical feeders

            'Insulationcnt avg_date

            'NULL                    256     1979
```

```
'XLP                    45099   1983

'PAPER          24976           1959

'EPR                    58125   1999

'commented on 08/16/2009 for new update attributes code

'Dim k As Integer

'Dim dr As DataRow



'For k = 0 To (attributes.Rows.Count - 1)

'dr = attributes.Rows(k)

'If IsDBNull(dr("feeder")) = True And IsDBNull(dr("from_struct")) = True And IsDBNull(dr("to_struct")) = True Then Exit Sub

'  Continue For

' End If



' If (Equals(SectionsSummary.Items.FindByText(dr("feeder") + "." + dr("from_struct") + "." + dr("to_struct")), Nothing)) Then

'SectionsSummary.Items.Add(dr("feeder") + "." + dr("from_struct") + "." + dr("to_struct"))

'sections_area.InnerText += System.Environment.NewLine + dr("feeder") + "." + dr("from_struct") + "." + dr("to_struct")
```

```
' Else

'commented on 01/27/2009

'duplicate_sections_list = duplicate_sections_list + 1

'  duplicate_sections_list = 0



'  End If

'modifiction for update_attribute ends here



'Updating Counts for cable sections

mon_Paper_GT40.Text = Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) -
Int32.Parse(dr("iPaperGT40")))

mon_Paper_LE40.Text = Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) -
Int32.Parse(dr("iPaperLE40")))

'dont need to add and subtract iSolidLT16 agb

mon_Solid_LT16.Text = Convert.ToString(Int32.Parse(mon_Solid_LT16.Text) +
Int32.Parse(dr("iPaperGT40")) + Int32.Parse(dr("iPaperLE40")) + Int32.Parse(dr("isolidGE16")) +
Int32.Parse(dr("iSolid70_75")))

mon_Solid_GT16.Text = Convert.ToString(Int32.Parse(mon_Solid_GT16.Text) -
Int32.Parse(dr("isolidGE16")))
```

```
mon_Solid_70_75.Text = Convert.ToString(Int32.Parse(mon_Solid_70_75.Text) -
Int32.Parse(dr("iSolid70_75")))

'Updating Counts for joints

'Elastimold Stop - just remove, don't replace

mon_E21GT93.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) -
Int32.Parse(dr("elasGT1993")))

mon_E21LE93.Text = Convert.ToString(Int32.Parse(mon_E21LE93.Text) -
Int32.Parse(dr("elasLE1993")))

'Raychem Stop - just remove, don't replace

mon_R31GT94.Text = Convert.ToString(Int32.Parse(mon_R31GT94.Text) -
Int32.Parse(dr("rayGT1993")))

mon_R31LE93.Text = Convert.ToString(Int32.Parse(mon_R31LE93.Text) -
Int32.Parse(dr("rayLE1993")))

'Other Stop - just remove, don't replace

mon_OtherStop_LT7.Text = Convert.ToString(Int32.Parse(mon_OtherStop_LT7.Text) -
Int32.Parse(dr("oStopLT7")))
```

```
mon_OtherStop7_12.Text = Convert.ToString(Int32.Parse(mon_OtherStop7_12.Text) -
Int32.Parse(dr("oStopBet7A12")))

mon_OtherStop_GT12.Text = Convert.ToString(Int32.Parse(mon_OtherStop_GT12.Text)
- Int32.Parse(dr("oStopGT12")))



'Updating total stops

mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) +
Int32.Parse(mon_E21LE93.Text) + Int32.Parse(mon_R31GT94.Text) + Int32.Parse(mon_R31LE93.Text) +
Int32.Parse(mon_OtherStop_LT7.Text) + Int32.Parse(mon_OtherStop7_12.Text) +
Int32.Parse(mon_OtherStop_GT12.Text))



'Paper - replace

mon_PJ_LT8.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) -
Int32.Parse(dr("papLT8")))

mon_PJ8_16.Text = Convert.ToString(Int32.Parse(mon_PJ8_16.Text) -
Int32.Parse(dr("papBet8A16")))

mon_PJ16_24.Text = Convert.ToString(Int32.Parse(mon_PJ16_24.Text) -
Int32.Parse(dr("papBet16A24")))

mon_PJ24_32.Text = Convert.ToString(Int32.Parse(mon_PJ24_32.Text) -
Int32.Parse(dr("papBet24A32")))
```

```
mon_PJ32_40.Text = Convert.ToString(Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40")))

mon_PJGT40.Text = Convert.ToString(Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40")))

'Updating total paper

mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) + Int32.Parse(mon_PJ8_16.Text) + Int32.Parse(mon_PJ16_24.Text) + Int32.Parse(mon_PJ24_32.Text) + Int32.Parse(mon_PJ32_40.Text) + Int32.Parse(mon_PJGT40.Text))

'Solid - update date

mon_SJ8_16.Text = Convert.ToString(Int32.Parse(mon_SJ8_16.Text) - Int32.Parse(dr("solBet8A16")))

mon_SJ16_24.Text = Convert.ToString(Int32.Parse(mon_SJ16_24.Text) - Int32.Parse(dr("solBet16A24")))

mon_SJ_GT24.Text = Convert.ToString(Int32.Parse(mon_SJ_GT24.Text) - Int32.Parse(dr("solGT24")))

'Finally, adding all to be replaced and date updated to solid
```

```
mon_SJ_LT8.Text = Convert.ToString(Int32.Parse(mon_SJ_LT8.Text) + Int32.Parse(dr("papLT8")) + Int32.Parse(dr("papBet8A16")) + Int32.Parse(dr("papBet16A24")) + Int32.Parse(dr("papBet24A32")) + Int32.Parse(dr("papBet32A40")) + Int32.Parse(dr("papGE40")) + Int32.Parse(dr("solBet8A16")) + Int32.Parse(dr("solBet16A24")) + Int32.Parse(dr("solGT24")) + Int32.Parse(dr("leadWipeCount")) + Int32.Parse(dr("raychem1w1w")) + Int32.Parse(dr("raychem3w1w")) + Int32.Parse(dr("other1w1w")))

' Next

'Updating Section cost text boxes

SectionsChangedSum.Text = SectionsSummary.Items.Count

CostPerSectionSum.Text = "18000"

TotalCostSum.Text = SectionsSummary.Items.Count * 18000

'Updating percentages for cable sections

mon_Percent_Paper_GT40.Text = Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) / Int32.Parse(Sec_CNT.Text) * 100.0)

mon_Percent_Paper_LE40.Text = Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) / Int32.Parse(Sec_CNT.Text) * 100.0)
```

```
mon_Percent_Solid_LT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_LT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_GT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_GT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_70_75.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_70_75.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))



'Updating total solids

mon_AllSolid.Text = Convert.ToString(Int32.Parse(mon_SJ8_16.Text) +
Int32.Parse(mon_SJ16_24.Text) + Int32.Parse(mon_SJ_GT24.Text) + Int32.Parse(mon_SJ_LT8.Text))



'updating total joints

mon_Joint_CNT.Text = Convert.ToString(Int32.Parse(mon_AllStop.Text) +
Int32.Parse(mon_AllPaper.Text) + Int32.Parse(mon_AllSolid.Text))



totalJoints = mon_Joint_CNT.Text
```

```
'Updating percentages

'Elastimold Stop Percentages

mon_Percent_E21GT93.Text = Convert.ToString((Int32.Parse(mon_E21GT93.Text) /
totalJoints) * 100)

mon_Percent_E21LE93.Text = Convert.ToString((Int32.Parse(mon_E21LE93.Text) /
totalJoints) * 100)

'Raychem Stop percentages

mon_Percent_R31GT94.Text = Convert.ToString((Int32.Parse(mon_R31GT94.Text) /
totalJoints) * 100)

mon_Percent_R31LE93.Text = Convert.ToString((Int32.Parse(mon_R31LE93.Text) /
totalJoints) * 100)

'Other Stop percentages

mon_Percent_OtherStop_LT7.Text =
Convert.ToString((Int32.Parse(mon_OtherStop_LT7.Text) / totalJoints) * 100)

mon_Percent_OtherStop7_12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop7_12.Text) / totalJoints) * 100)
```

```
                mon_Percent_OtherStop_GT12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop_GT12.Text) / totalJoints) * 100)


                'Updating total stop percentages

                mon_Percent_ALLStop.Text = Convert.ToString((Int32.Parse(mon_AllStop.Text) /
totalJoints) * 100)


                'Paper percentages

                mon_Percent_PJ_LT8.Text = Convert.ToString((Int32.Parse(mon_PJ_LT8.Text) /
totalJoints) * 100)

                mon_Percent_PJ8_16.Text = Convert.ToString((Int32.Parse(mon_PJ8_16.Text) /
totalJoints) * 100)

                mon_Percent_PJ16_24.Text = Convert.ToString((Int32.Parse(mon_PJ16_24.Text) /
totalJoints) * 100)

                mon_Percent_PJ24_32.Text = Convert.ToString((Int32.Parse(mon_PJ24_32.Text) /
totalJoints) * 100)

                mon_Percent_PJ32_40.Text = Convert.ToString((Int32.Parse(mon_PJ32_40.Text) /
totalJoints) * 100)

                mon_Percent_PJGT40.Text = Convert.ToString((Int32.Parse(mon_PJGT40.Text) /
totalJoints) * 100)
```

```
'Updating total paper percentages

mon_Percent_AllPaper.Text = Convert.ToString((Int32.Parse(mon_AllPaper.Text) /
totalJoints) * 100)

'Solid percentages

mon_Percent_SJ_LT8.Text = Convert.ToString((Int32.Parse(mon_SJ_LT8.Text) /
totalJoints) * 100)

mon_Percent_SJ8_16.Text = Convert.ToString((Int32.Parse(mon_SJ8_16.Text) /
totalJoints) * 100)

mon_Percent_SJ16_24.Text = Convert.ToString((Int32.Parse(mon_SJ16_24.Text) /
totalJoints) * 100)

mon_Percent_SJ_GT24.Text = Convert.ToString((Int32.Parse(mon_SJ_GT24.Text) /
totalJoints) * 100)

'Updating total solids

mon_Percent_AllSolid.Text = Convert.ToString((Int32.Parse(mon_AllSolid.Text) /
totalJoints) * 100)

End Sub
```

```
Private Sub Reset_MTBFs()

    StrategyLoadDropDown.SelectedIndex = 0

    StrategyPILCDropDown.SelectedIndex = 0

    StrategyRelDropDown.SelectedIndex = 0

    RelPercentStops.SelectedIndex = 0

    RelPercentElastimold.SelectedIndex = 0

    RelPercent.SelectedIndex = 0

    SectionsChangedLoad.Text = ""

    SectionsChangedPILC.Text = ""

    SectionsChangedRelStops.Text = ""

    SectionsChangedRelElastimold.Text = ""

    SectionsChangedRel.Text = ""

    actualPercentPILC.Text = ""

    actualTotalCostPILC.Text = ""



    Reset_MTBF_Load()
```

```
    Reset_MTBF_PILC()

    Reset_MTBF_Rel()

        Reset_MTBF_LPW()

End Sub



Private Sub Reset_MTBF_Load()

    RankBaseLoad.Text = ""

    FOTBaseLoad.Text = ""

    MTBFBaseLoad.Text = ""

    RankNewLoad.Text = ""

    RankChangeLoad.Text = ""

    MTBFNewLoad.Text = ""

    MTBFChangeLoad.Text = ""

    FOTNewLoad.Text = ""

    FOTChangeLoad.Text = ""

    TotalCostLoad.Text = ""

    loadStrategyInfo.Visible = False
```

```
End Sub

Private Sub Reset_MTBF_PILC()

    RankBasePILC.Text = ""

    FOTBasePILC.Text = ""

    MTBFBasePILC.Text = ""

    RankNewPILC.Text = ""

    RankChangePILC.Text = ""

    MTBFNewPILC.Text = ""

    MTBFChangePILC.Text = ""

    FOTNewPILC.Text = ""

    FOTChangePILC.Text = ""

    'PercentPILC.SelectedIndex = 0

    'TotalCostPILC.Text = ""

    PILCMTBFInfo.Visible = False

    Panel9.Visible = False

    PILCMTBFTotalCostInfo.Visible = False
```

```
    LoadandPILCPanel.Visible = False

    sectionschangedPILCPanel.Visible = False




End Sub



Private Sub Reset_MTBF_Rel()

    'for xlp tier

    RankBaseRel.Text = ""

    FOTBaseRel.Text = ""

    MTBFBaseRel.Text = ""

    RankNewRel.Text = ""

    RankChangeRel.Text = ""

    MTBFNewRel.Text = ""

    MTBFChangeRel.Text = ""

    FOTNewRel.Text = ""
```

```
FOTChangeRel.Text = ""

'for stopjoints tier

RankBaseRelStops.Text = ""

FOTBaseRelStops.Text = ""

MTBFBaseRelStops.Text = ""

RankNewRelStops.Text = ""

RankChangeRelStops.Text = ""

MTBFNewRelStops.Text = ""

MTBFChangeRelStops.Text = ""

FOTNewRelStops.Text = ""

FOTChangeRelStops.Text = ""

'for elastimold iter

RankBaseRelElast.Text = ""

FOTBaseRelElast.Text = ""

MTBFBaseRelElast.Text = ""

RankNewRelElast.Text = ""

RankChangeRelElast.Text = ""

MTBFNewRelElast.Text = ""
```

```
    MTBFChangeRelElast.Text = ""

    FOTNewRelElast.Text = ""

    FOTChangeRelElast.Text = ""




End Sub



Private Sub Reset_MTBF_LPW()

    RankBaseLPW.Text = ""

    FOTBaseLPW.Text = ""

    MTBFBaseLPW.Text = ""

    RankNewLPW.Text = ""

    RankChangeLPW.Text = ""

    MTBFNewLPW.Text = ""

    MTBFChangeLPW.Text = ""

    FOTNewLPW.Text = ""

    FOTChangeLPW.Text = ""
```

```
End Sub

Public Function rand(ByVal low As Int32, ByVal high As Int32) As Integer

   Static RandomNumGen As New System.Random

   Return RandomNumGen.Next(low, high + 1)

End Function




Private Sub Calculate_New_Load_Runs()

   'modified by shringika on dec 21'2007 included if case

   If cmbSearchFeeder.Items.Count > 0 Then



      If SectionsChangedLoad.Text = "" Then



         'MTBF_NA.Visible = True

         Return

      End If
```

```
loadStrategyInfo.Visible = True

Dim feeder As String

feeder = cmbSearchFeeder.SelectedValue

If feeder.Length < 5 Then feeder = "0" + feeder

Dim pct As Integer

If StrategyLoadDropDown.SelectedIndex = 2 Then

  pct = 100

ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then

  pct = 105

Else

  pct = 110

End If



'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '"
& cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score_CAPTV10_RV1_2008]
WHERE [Feeder] = '" & feeder & "' and load_relief_percent = 100 and pilc_percent = 0"
```

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score_CAPTV10_RV1_2008] WHERE [Feeder] = '" & feeder & "' and load_relief_percent =" & pct & " and pilc_percent = 0"

Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [Feeder] = '" & feeder & "' and load_relief_percent =" & pct & " and pilc_percent = 0"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and load_relief_percent = 100 and pilc_percent = 0"

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

```
Dim Rank As Integer

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_LR_NA.Visible = True

   Reset_MTBF_Load()

   Return

End If




Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary




RankBaseLoad.Text = dr("OldRank")

Rank = Int32.Parse(dr("OldRank"))
```

```
'conversion  of rank to MTBF using regression (New 6-08)

' MTBFBaseLoad.Text = (Rank * 0.3386 + 165.96).ToString("F0")

MTBFBaseLoad.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

'MTBFBaseLoad.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseLoad.Text = Convert.ToString(Math.Round(Math.Max((Rank * -0.0095 + 6),
0.0), 0))



'TEMP!!!!!!!!!!!!!

'RankNewLoad.Text = Convert.ToString(rand(Rank, Rank + 100))

RankNewLoad.Text = dr("rank")

'RankNewPILC.Text = "930"

RankChangeLoad.Text = Convert.ToString(Int32.Parse(RankNewLoad.Text) - Rank)

'conversion  of rank to MTBF using regression

MTBFNewLoad.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")
```

```
                MTBFChangeLoad.Text = Convert.ToDouble(dr("delta")).ToString("F0")

                'MTBFNewLoad.Text = (Int32.Parse(RankNewLoad.Text) * 0.3386 +
165.96).ToString("F0")

                ' MTBFChangeLoad.Text =
Convert.ToString(Convert.ToDouble(MTBFNewLoad.Text) - Convert.ToDouble(MTBFBaseLoad.Text))

                FOTNewLoad.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewLoad.Text) * -0.0095 + 6), 0.0), 0))

                FOTChangeLoad.Text = Convert.ToString(Convert.ToDouble(FOTBaseLoad.Text) -
Convert.ToDouble(FOTNewLoad.Text))

                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If

        End Sub
```

```
Private Sub Calculate_New_Load()

'not used now since we do RUNS not sections



'modified by shringika on dec 21'2007 included if case

If cmbSearchFeeder.Items.Count > 0 Then



Dim sql As String = "SELECT * FROM ediprodb2.FeederSuscept.dbo.WhatIf_Ranking
WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("DB2FeederSuscept").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)



Dim dt As DataTable = New DataTable



myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)
```

```
'ABOVE is only temporary

RankBaseLoad.Text = dr("real_rank")

'delete line below when webservice is working



'conversion  of rank to MTBF using regression

MTBFBaseLoad.Text = (Int32.Parse(RankBaseLoad.Text) * 0.311 +
150.0).ToString("F0")

FOTBaseLoad.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankBaseLoad.Text) * -0.0095 + 6), 0.0), 0))



RankNewLoad.Text = "890"

RankChangeLoad.Text = Convert.ToString(Int32.Parse(RankNewLoad.Text) -
Int32.Parse(RankBaseLoad.Text))

'conversion  of rank to MTBF using regression

MTBFNewLoad.Text = (Int32.Parse(RankNewLoad.Text) * 0.311 +
150.0).ToString("F0")



'MTBFNewLoad.Text = Convert.ToString(Int32.Parse(RankNewLoad.Text) * 0.311 +
150.0)
```

```
                MTBFChangeLoad.Text = Convert.ToString(Convert.ToDouble(MTBFNewLoad.Text)
- Convert.ToDouble(MTBFBaseLoad.Text))

                FOTNewLoad.Text = Convert.ToString(Math.Max((Int32.Parse(RankNewLoad.Text) *
-0.0095 + 6), 0.0))

                FOTChangeLoad.Text = Convert.ToString(Convert.ToDouble(FOTBaseLoad.Text) -
Convert.ToDouble(FOTNewLoad.Text))



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If

        End Sub



        Private Sub Calculate_New_PILC()



            sectionschangedPILCPanel.Visible = True



            If RblTotalCostPercentagePILC.SelectedIndex > 1 Then

                sectionschangedPILCPanel.Visible = False
```

```
Dim feeder_Id As String

feeder_Id = New String("X")

feeder_Id = cmbSearchFeeder.SelectedValue

If feeder_Id.Length < 5 Then

   feeder_Id = "0" + feeder_Id



End If



Dim data_new_rank, data_base_rank, base_rank As String

data_base_rank = "feeder=" + feeder_Id.ToString + ";"

data_base_rank += "Static_PaperCable_GT_40 =" + Paper_GT40.Text + ";" +
"Static_PaperCable_LE_40 =" + Paper_LE40.Text + ";" + "Static_SolidCable_LT_16 =" + Solid_LT16.Text +
";"

data_base_rank += "Static_SolidCable_70_75 =" + Solid_70_75.Text + ";" +
"Static_SolidCable_GE_16 =" + Solid_GT16.Text + ";" + "Static_Sections_Total=" + Sec_CNT.Text + ";"

data_base_rank += "Static_E2_1_GT_1993=" + E21GT93.Text + ";" +
"Static_E2_1_LE_1993 =" + E21LE93.Text + ";" + "Static_R3_1_GT_1993=" + R31GT94.Text + ";"
```

```
data_base_rank += "Static_R3_1_LE_1993 =" + R31LE93.Text + ";" + "Static_Other_Stops_LT_7 =" + OtherStop_LT7.Text + ";" + "Static_Other_Stops_7_12 =" + OtherStop7_12.Text + ";"

data_base_rank += "Static_Other_Stops_GT_12 =" + OtherStop_GT12.Text + ";" + "Static_Stops_Total =" + AllStop.Text + ";" + "Static_PaperJoints_LT_8=" + PJ_LT8.Text + ";"

data_base_rank += "Static_PaperJoints_8_16=" + PJ8_16.Text + ";" + "Static_PaperJoints_16_24 =" + PJ16_24.Text + ";" + "Static_PaperJoints_24_32 =" + PJ24_32.Text + ";"

data_base_rank += "Static_PaperJoints_32_40 =" + PJ32_40.Text + ";" + "Static_PaperJoints_GE_40 =" + PJGT40.Text + ";" + "Static_PaperJoints_Total =" + AllPaper.Text + ";"

data_base_rank += "Static_SolidJoints_LT_8 =" + SJ_LT8.Text + ";" + "Static_SolidJoints_8_16 =" + SJ8_16.Text + ";" + "Static_SolidJoints_16_24 =" + SJ16_24.Text + ";" + "Static_SolidJoints_GT_24 =" + SJ_GT24.Text + ";" + "Static_SolidJoints_Total=" + AllSolid.Text


base_rank = WebserviceClient.WebserviceClient.proxy_rerank(data_base_rank.ToString)

Dim i As Integer

i = base_rank.IndexOf(",")


RankBasePILC.Text = base_rank.Substring(i + 1)
```

```
Dim score As Double

score = Double.Parse(base_rank.Substring(0, i - 1))



' MTBFBasePILC.Text = -(112.33 * score) + 278.71

'MTBFBasePILC.Text = (1.1638 * score) + 1.6793



Dim pilcmtbf_base As Integer



pilcmtbf_base = Math.Round(1.1733 * score) + 1.6807



If pilcmtbf_base = 0 Then

   MTBFBasePILC.Text = 4999

Else

   MTBFBasePILC.Text = 1 / Math.Round(1.1733 * score + 1.6807, 3) * 365



End If

' MTBFBasePILC.Text = (1.1733 * score) + 1.6807
```

```
data_new_rank = "feeder=" + feeder_Id.ToString + ";"

data_new_rank += "Static_PaperCable_GT_40 =" + mon_Paper_GT40.Text + ";" +
"Static_PaperCable_LE_40 =" + mon_Paper_LE40.Text + ";" + "Static_SolidCable_LT_16 =" +
mon_Solid_LT16.Text + ";"

data_new_rank += "Static_SolidCable_70_75 =" + mon_Solid_70_75.Text + ";"
+ "Static_SolidCable_GE_16 =" + mon_Solid_GT16.Text + ";" + "Static_Sections_Total=" + mon_Sec_CNT.Text
+ ";"

data_new_rank += "Static_E2_1_GT_1993 =" + mon_E21GT93.Text + ";" +
"Static_E2_1_LE_1993 =" + mon_E21LE93.Text + ";" + "Static_R3_1_GT_1993=" + mon_R31GT94.Text + ";"

data_new_rank += "Static_R3_1_LE_1993 =" + mon_R31LE93.Text + ";" +
"Static_Other_Stops_LT_7 =" + mon_OtherStop_LT7.Text + ";" + "Static_Other_Stops_7_12 =" +
mon_OtherStop7_12.Text + ";"

data_new_rank += "Static_Other_Stops_GT_12 =" + mon_OtherStop_GT12.Text + ";"
+ "Static_Stops_Total =" + mon_AllStop.Text + ";" + "Static_PaperJoints_LT_8=" + mon_PJ_LT8.Text + ";"

data_new_rank += "Static_PaperJoints_8_16=" + mon_PJ8_16.Text + ";" +
"Static_PaperJoints_16_24 =" + mon_PJ16_24.Text + ";" + "Static_PaperJoints_24_32 =" + mon_PJ24_32.Text +
";"

data_new_rank += "Static_PaperJoints_32_40 =" + mon_PJ32_40.Text + ";" +
"Static_PaperJoints_GE_40 =" + mon_PJGT40.Text + ";" + "Static_PaperJoints_Total =" + mon_AllPaper.Text +
";"
```

```
data_new_rank += "Static_SolidJoints_LT_8 =" + mon_SJ_LT8.Text + ";" +
"Static_SolidJoints_8_16 =" + mon_SJ8_16.Text + ";" + "Static_SolidJoints_16_24 =" + mon_SJ16_24.Text + ";" +
"Static_SolidJoints_GT_24 =" + mon_SJ_GT24.Text + ";" + "Static_SolidJoints_Total=" + mon_AllSolid.Text

Dim new_rank As String

new_rank =
WebserviceClient.WebserviceClient.proxy_rerank(data_new_rank.ToString)

Dim j As Integer

j = new_rank.IndexOf(",")

Dim score_new As Double

score_new = Double.Parse(new_rank.Substring(0, j - 1))

RankNewPILC.Text = new_rank.Substring(j + 1)

RankChangePILC.Text = Convert.ToString(Int32.Parse(RankNewPILC.Text) -
Int32.Parse(RankBasePILC.Text))
```

```
' MTBFNewPILC.Text = -(112.33 * score_new) + 278.71

'MTBFNewPILC.Text = (1.1638 * score_new) + 1.6793

'MTBFNewPILC.Text = (1.1733 * score_new) + 1.6807



Dim pilcmtbf_new As Integer



pilcmtbf_new = Math.Round(1.1733 * score_new) + 1.6807



If pilcmtbf_new = 0 Then

   MTBFNewPILC.Text = 4999

Else

   MTBFNewPILC.Text = 1 / Math.Round(1.1733 * score_new + 1.6807, 3) * 365



End If
```

```
MTBFNewPILC.Text = Math.Round(Convert.ToDouble(MTBFNewPILC.Text))

MTBFChangePILC.Text = Convert.ToString(Convert.ToDouble(MTBFNewPILC.Text)
- Convert.ToDouble(MTBFBasePILC.Text))

MTBFBasePILC.Text = Math.Round(Convert.ToDouble(MTBFBasePILC.Text))

MTBFChangePILC.Text = Math.Round(Convert.ToDouble(MTBFChangePILC.Text))

FOTBasePILC.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankBasePILC.Text) * -0.0095 + 6), 0.0), 0))

FOTNewPILC.Text = Convert.ToString(Math.Max((Int32.Parse(RankNewPILC.Text) *
-0.0095 + 6), 0.0))

FOTChangePILC.Text = Convert.ToString(Convert.ToDouble(FOTBasePILC.Text) -
Convert.ToDouble(FOTNewPILC.Text))


FOTBasePILC.Text = Math.Round(Convert.ToDouble(FOTBasePILC.Text))

FOTChangePILC.Text = Math.Round(Convert.ToDouble(FOTChangePILC.Text))

FOTNewPILC.Text = Math.Round(Convert.ToDouble(FOTNewPILC.Text))


selectedStrategy.Text = StrategyPILCDropDown.SelectedItem.Text

sectionsChangedValue.Text = SectionsChangedPILC.Text

totalCostValue.Text = actualTotalCostPILC.Text
```

```
    Panel9.Visible = True




    Exit Sub



End If




Dim cost_percent As Integer

Dim sql As String

lblCostPercentage.Text = PercentPILC.Text

lblCostPercentageValue.Text = actualPercentPILC.Text

'modified by shringika on dec 21'2007 included if case

If cmbSearchFeeder.Items.Count > 0 Then

    MTBF_NA.Visible = False

    If actualPercentPILC.Text = "" Then

        MTBF_NA.Visible = True
```

```
        Return

    End If

    Dim load_strategy As String


    If Session("strategies_load") Is Nothing Then Session("strategies_load") = "0"

    load_strategy = CType(Session("strategies_load"), String)

    sectionschangedPILCPanel.Visible = True


    lblSectionsChangedPILCMTBF.Text = SectionsChangedPILC.Text

    'feeder  network pilc_strategy pilc_percent cost delta NewMTBF       OldMTBF
OldScore OldRank score rank       versionf

    If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

      ' lblTarget.Text = "For Target Percentage "

      lblCostPercentage.Text = PercentPILC.Text


      ' lblStrategy.Text = "and strategy = "

      lblSelectedStrategy.Text = StrategyPILCDropDown.SelectedItem.Text

      ' lblAchieved.Text = " achieved percentage was :"
```

```
PILCMTBFInfo.Visible = True

lblCostPercentageValue.Text = actualPercentPILC.Text

Dim pct As Integer

If StrategyLoadDropDown.SelectedIndex = 2 Then

  pct = 100

ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then

  pct = 105

Else

  pct = 110

End If



If load_strategy.Length > 2 Then

  sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent ='" & PercentPILC.Text & "' and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent = " & pct & ""

  LoadandPILCPanel.Visible = True



  lblLoadPercentagePILCMTBF.Text = StrategyLoadDropDown.SelectedValue
```

```
Else

    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent ='" & PercentPILC.Text & "' and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent= 0"

    LoadandPILCPanel.Visible = False

End If


Else

    PILCMTBFTotalCostInfo.Visible = True

    If Int32.Parse(actualPercentPILC.Text) < 5 Then

      cost_percent = 5

    ElseIf Int32.Parse(actualPercentPILC.Text) < 10 Then

      cost_percent = 10

    ElseIf Int32.Parse(actualPercentPILC.Text) < 15 Then

      cost_percent = 15

    ElseIf Int32.Parse(actualPercentPILC.Text) < 20 Then
```

```
    cost_percent = 20
ElseIf Int32.Parse(actualPercentPILC.Text) < 25 Then
    cost_percent = 25
ElseIf Int32.Parse(actualPercentPILC.Text) < 50 Then
    cost_percent = 50
ElseIf Int32.Parse(actualPercentPILC.Text) < 75 Then
    cost_percent = 75
ElseIf Int32.Parse(actualPercentPILC.Text) < 90 Then
    cost_percent = 90
ElseIf Int32.Parse(actualPercentPILC.Text) < 95 Then
    cost_percent = 95
Else
    cost_percent = 100
End If
If load_strategy.Length > 0 Then
    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent =" & cost_percent & " and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent= 100"
```

```
                LoadandPILCPanel.Visible = True




            Else

                LoadandPILCPanel.Visible = False

                sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent ='" & cost_percent & "' and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent= 0"

            End If




            'modified on 07/14/2008 for load_relief_percent=100

            ' Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] =
'" & cmbSearchFeeder.SelectedValue & "' and load_relief_percent = 100 and pilc_percent ='" & PercentPILC.Text
& "'"



            lblTotalCostPILC.Text = TotalCostPILC.Text

            lblactualCostPILC.Text = actualTotalCostPILC.Text
```

```
lblSelectedStrategyTotalCost.Text = StrategyPILCDropDown.SelectedItem.Text

lblPILCActualPercent.Text = actualPercentPILC.Text



Dim total_sections, total_cost, cost_per_section As Integer

total_sections = Int32.Parse(Sec_CNT.Text)

total_cost = Int32.Parse(TotalCostPILC.Text)

cost_per_section = Int32.Parse(CostPerSectionPILC.Text)



Dim total_replaced_sections, target_percent, pilc_sections As Integer

pilc_sections = lblPILCTargetPercent.Text

total_replaced_sections = Math.Floor(total_cost / cost_per_section)

target_percent = Math.Floor((total_replaced_sections / pilc_sections) * 100)

lblPILCTargetPercent.Text = target_percent.ToString



End If
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

Dim rank As Integer

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_NA.Visible = True

   Return

End If
```

```
Dim dr As DataRow = dt.Rows(0)

RankBasePILC.Text = dr("OldRank")

rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression

MTBFBasePILC.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBasePILC.Text = Convert.ToString(Math.Round(Math.Max((rank * -0.0095 + 6),
0.0), 0))

RankNewPILC.Text = dr("rank")

'RankNewPILC.Text = "930"

RankChangePILC.Text = Convert.ToString(Int32.Parse(RankNewPILC.Text) - rank)

'conversion  of rank to MTBF using regression

MTBFNewPILC.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")
```

```
                    MTBFChangePILC.Text = Convert.ToDouble(dr("delta")).ToString("F0")


                    FOTNewPILC.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewPILC.Text) * -0.0095 + 6), 0.0), 0))

                    FOTChangePILC.Text = Convert.ToString(Convert.ToDouble(FOTBasePILC.Text) -
Convert.ToDouble(FOTNewPILC.Text))

                    lblCostPercentage.Visible = True

                    lblCostPercentageValue.Visible = True

                    ' If RankBasePILC.Text = "" Or MTBFBasePILC.Text = "" Or FOTBasePILC.Text = ""
Or RankNewPILC.Text = "" Or MTBFNewPILC.Text = "" Or FOTNewPILC.Text = 0 Then

                    'MTBF_NA.Visible = True

                    'RankBasePILC.Text = ""

                    ' MTBFBasePILC.Text = ""

                    ' FOTBasePILC.Text = ""

                    ' RankNewPILC.Text = ""

                    ' MTBFNewPILC.Text = ""

                    ' FOTNewPILC.Text = ""
```

```
        'End If



        If MTBFChangePILC.Text.Length = 0 Then

          MTBF_NA.Visible = True

        Else

          MTBF_NA.Visible = False



        End If

        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

      End If

    End Sub
```

```
Private Sub Calculate_New_Rel()

    'modified by shringika on dec 21'2007 included if case

    If cmbSearchFeeder.Items.Count > 0 Then

        If RelPercent.Text = "" Then

            MTBF_Rel_NA.Visible = True

            Return

        End If



        Dim dt As DataTable = New DataTable

        Dim Rank As Integer

        'have to work on this query more



        'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '"
& cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"
```

```
Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and rel_percent = '" & RelPercent.Text & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_Rel_NA.Visible = True

   Return

End If



Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary



RankBaseRel.Text = dr("OldRank")
```

```
Rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression (New 6-08)

'modified on 01/04/2009 for picking up rank from rerank_cost_score

'MTBFBaseRel.Text = (Rank * 0.3386 + 165.96).ToString("F0")



MTBFBaseRel.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseRel.Text  =  Convert.ToString(Math.Round(Math.Max((Rank  *  -0.0095  +  6),
0.0), 0))



'TEMP!!!!!!!!!!!!!



' RankNewRel.Text = Convert.ToString(rand(Rank, Rank + 100))

'modified on 01/04/2009 for picking up rank from rerank_cost_score

RankNewRel.Text = dr("rank")

RankChangeRel.Text = Convert.ToString(Int32.Parse(RankNewRel.Text) - Rank)

'conversion  of rank to MTBF using regression

MTBFNewRel.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")
```

```
MTBFChangeRel.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'added on 01/04/2009 for picking up new mtbf and delta from database , not by calculation

' MTBFNewRel.Text = (Int32.Parse(RankNewRel.Text) * 0.3386 + 165.96).ToString("F0")

'MTBFChangeRel.Text = Convert.ToString(Convert.ToDouble(MTBFNewRel.Text) - Convert.ToDouble(MTBFBaseRel.Text))

FOTNewRel.Text = Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewRel.Text) * -0.0095 + 6), 0.0), 0))

FOTChangeRel.Text = Convert.ToString(Convert.ToDouble(FOTBaseRel.Text) - Convert.ToDouble(FOTNewRel.Text))

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

If MTBFChangeRel.Text.Length = 0 Then

  MTBF_Rel_NA.Visible = True
```

```
Else

    MTBF_Rel_NA.Visible = False



End If

End If

'end if added by shringika on 21dec 2007 as part of validation

End Sub

Protected Sub CalculateNewLoadMTBF_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewLoadMTBF.Click

    MTBF_LR_NA.Visible = False
```

```
        Calculate_New_Load_Runs()

        'Calculate_New_Load()



    End Sub

    Protected Sub CalculateNewPILCMTBF_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewPILCMTBF.Click

        MTBF_NA.Visible = False

        Calculate_New_PILC()



    End Sub

    Protected Sub CalculateNewRelMTBF_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewRelMTBF.Click

        MTBF_Rel_NA.Visible = False

        Calculate_New_Rel()
```

```
        End Sub


    #Region " Costs "

        Protected Sub Initial_Costs()

          Initial_Costs_Load()

          Initial_Costs_PILC()

          Initial_Costs_Rel()



        End Sub

        Protected Sub CostPerSection_Load_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionLoad.TextChanged

          'TotalCost.Text = "$$$"

          If SectionsChangedLoad.Text <> "" Then

            Total_Cost_Load()

          End If

        End Sub

        Private Sub Initial_Costs_Load()
```

```
        CostPerSectionLoad.Text = "18000.00"

        TotalCostLoad.Text = ""

        SectionsChangedLoad.Text = ""



    End Sub

    Private Sub Total_Cost_Load()

        TotalCostLoad.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedLoad.Text) *
Convert.ToDouble(CostPerSectionLoad.Text)), 2))

    End Sub



    Protected Sub CostPerSection_PILC_Text_Changed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionPILC.TextChanged

        'TotalCost.Text = "$$$"

        If SectionsChangedPILC.Text <> "" Then

            Total_Cost_PILC()

        End If

    End Sub

    '--modified by shringika on 02/14/2008 for actualtotalcostpilc field
```

```
Private Sub Initial_Costs_PILC()

    CostPerSectionPILC.Text = "18000"

    actualTotalCostPILC.Text = ""

    SectionsChangedPILC.Text = ""

    'TotalCostPILC.Text = "180000"

End Sub

Private Sub Total_Cost_PILC()

    actualTotalCostPILC.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedPILC.Text) *
Convert.ToDouble(CostPerSectionPILC.Text)), 2))

End Sub '*--modification by shringika on 02/14/2008 for actualTotalCostPILC ends

Protected Sub CostPerSection_Rel_Text_Changed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionRel.TextChanged

    'TotalCost.Text = "$$$"
```

```
    If SectionsChangedRel.Text <> "" Then

      Total_Cost_Rel()

    End If

End Sub



Private Sub Initial_Costs_Rel()



    CostPerSectionRel.Text = "18000.00"

    TotalCostRel.Text = ""

    SectionsChangedRel.Text = ""

    CostPerSectionRelElastimold.Text = "18000.00"

    TotalCostRelElastimold.Text = ""

    SectionsChangedRelElastimold.Text = ""



End Sub



Private Sub Total_Cost_Sum()
```

```
                TotalCostSum.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRel.Text) *
Convert.ToDouble(CostPerSectionRel.Text)), 2))

            End Sub


            Protected Sub CostPerSection_Sum_Text_Changed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionSum.TextChanged

              'TotalCost.Text = "$$$"

              If SectionsChangedSum.Text <> "" Then

                Total_Cost_Rel()

              End If

            End Sub


            Private Sub Initial_Costs_Sum()


              CostPerSectionSum.Text = "18000"

              TotalCostSum.Text = ""

              'TotalCostPILC.Text = "180000"

              'SectionsChangedRel.Text = ""
```

```
End Sub

Private Sub Total_Cost_Rel()
  TotalCostRel.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRel.Text) *
Convert.ToDouble(CostPerSectionRel.Text)), 2))
End Sub
#End Region

Private Function getResultsPILCTable() As DataTable
  'gather and validate parameters here
  Dim Feeder As String = cmbSearchFeeder.SelectedValue

  'Get data from the database
  'Dim sql As String = "SELECT [Feeder PILC RANK] AS Feeder_PILC_Rank, [NWK
PILC RANK] AS NWK_PILC_Rank, [PILC Boro Rank] AS Boro_PILC_Rank, description FROM
[Cable_section_rank_pilc_all] WHERE ([Feeder] = '{0}') ORDER BY [Feeder PILC RANK]"
```

```
Dim sql As String = "SELECT * FROM [Rerank_Cable_section_rank_pilc_all] WHERE ([Feeder] = '{0}') ORDER BY [Feeder PILC RANK]"

sql = String.Format(sql, Feeder)

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Return dt

End Function
```

```
Protected Sub btnFOOrunExcel_click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnovlrunExcel.Click

'work on this later

Return

Dim dt As DataTable = getResultsPILCTable()

If dt Is Nothing Then
  Return
End If

Dim cols As Integer = dt.Columns.Count

'Output excel file
```

```
Response.Clear()

Response.ContentType = "application/xls"

Response.AddHeader("content-disposition", "attachment;
filename=""CapitalPlanning_PILC_Rank.xls""")

Response.Write("<table border='1'>")


Dim i As Integer


'Print Row Headings

Response.Write("<tr>")


Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Feeder Rank</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Network Rank</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Boro Rank</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Description</td>")
```

```
    Response.Write("</tr>")


    'Print Rows
    For Each row As DataRow In dt.Rows
      Response.Write("<tr>")
      For i = 0 To row.ItemArray.Length() - 1
        Response.Write("<td>" & row(i) & "</td>")
      Next
      Response.Write("</tr>")
    Next
    Response.Write("</table>")
    Response.End()
  End Sub
```

```
Protected Sub FMBoro_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles FMBoro.Click

    Dim URL As String

    URL = "feedermaps/" & cmbSearchBorough.SelectedValue & ".html"

    Dim MyScript As New System.Text.StringBuilder
    With MyScript
      .Append("<script type='text/javascript'>")
      .Append("window.open('" & URL & "');")
      .Append("</script>")
    End With

    'Response.Write
    Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
```

'Page.ClientScript.RegisterStartupScript(GetType(String), "ClientScript", "alert ('hello');",
True)

End Sub

Protected Sub FMNet_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles FMNet.Click

Dim URL As String

URL = "feedermaps/" & cmbSearchNetwork.SelectedValue.Trim & ".load.html"

Dim MyScript As New System.Text.StringBuilder

With MyScript

```
.Append("<script type='text/javascript'>")

.Append("window.open('" & URL & "');")

.Append("</script>")

End With



'Response.Write

Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

End Sub



Protected Sub FMFeeder_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles FMFeeder.Click

Dim URL As String



'URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.html"

URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"



Dim MyScript As New System.Text.StringBuilder
```

```
With MyScript
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Response.Write
Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
End Sub

'Protected Sub CustomValidator1_ServerValidate(ByVal source As Object, ByVal args As
System.Web.UI.WebControls.ServerValidateEventArgs) Handles CustomValidator1.ServerValidate
' args.IsValid = SectionsChangedLoad.Text.Length() > 0
'End Sub

'Helpers ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
Private Function CalculateCostPILC_Feeder(ByVal Feeder As String, ByVal strategy As
Integer) As Integer

    Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder))

    Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder))

    Dim CNT As Integer



    BindAttributes()

    BindRankingLPW()

    BindFOT()

    'Reset_New_Attributes()

    'added by shringika on 03/11/2008 for LPW step

    'BindAttributes1()

    ' modification ends here 03/11/2008

    If strategy = 1 Then
```

```
    'dv.Sort = "Run_ID ASC"

    dv.Sort = "[Feeder PILC Rank] ASC"

    dv1.Sort = "rank ASC"

    'Replace_paper_percent uses the sort order from above

End If



If strategy = 2 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "load_percentage DESC"

    dv1.Sort = "percentage_load DESC"

End If



If strategy = 3 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "LoadXrank DESC"

    dv1.Sort = "LoadXrank DESC"

End If
```

```
CNT = Replace_Paper_percent(PercentPILC.Text, dv1)

Return CNT



'GVpilcsec.DataSource = dv

'GVpilcsec.DataBind()



'GVpilcruns.DataSource = dv1

'GVpilcruns.DataBind()




'PILC.Visible = False



End Function



'--added by shringika on 17-dec-2007

Protected Sub cmbFeederType_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbFeederType.SelectedIndexChanged
```

```
cmbPercentage.Enabled = False

If cmbFeederType.SelectedItem.Value.Contains("PILC") Then
    cmbPercentage.Items.Clear()
    cmbPercentage.Items.Add("10")
    cmbPercentage.Items.Add("20")
    cmbPercentage.Items.Add("25")
    cmbPercentage.Items.Add("30")
    cmbPercentage.Items.Add("40")
    cmbPercentage.Items.Add("50")
    cmbPercentage.Items.Add("60")
    cmbPercentage.Items.Add("70")
    cmbPercentage.Items.Add("75")
    cmbPercentage.Items.Add("80")
    cmbPercentage.Items.Add("90")
    cmbPercentage.Items.Add("100")
    cmbPercentage.Enabled = True
```

```
End If

If cmbFeederType.SelectedItem.Value.Contains("Load") Then
    cmbPercentage.Items.Clear()
    cmbPercentage.Items.Add("70")
    cmbPercentage.Items.Add("80")
    cmbPercentage.Items.Add("90")
    cmbPercentage.Items.Add("100")
    cmbPercentage.Items.Add("110")
    cmbPercentage.Items.Add("120")
    cmbPercentage.Items.Add("130")
    cmbPercentage.Items.Add("140")
    cmbPercentage.SelectedValue = "100"
    cmbPercentage.Enabled = True

End If
```

```
        BindSearchFeederList()

    End Sub

    Protected Sub cmbPercentage_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbPercentage.SelectedIndexChanged

        BindSearchFeederList()

    End Sub

    '--modification by shringika on 17-dec-2007 ends

    '--modification by shringika 21-dec-2007 ResetFeederAttributes

    Private Sub ResetFeederAttributes()

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

        If cmbSearchFeeder.Items.Count < 1 Then
```

```
Label2.Text = ""

Feeder.Text = ""

Feeder1.Text = ""

Feeder2.Text = ""

' Feeder3.Text = ""

Feeder4.Text = ""

'Feeder5.Text = ""

Feeder6.Text = ""

Feeder7.Text = ""

feeder11.Text = ""

feeder20.Text = ""

feeder12.Text = ""

Feeder_Name.Text = " "

'SelectedFeederName.TabIndex = ""

SelectedFeederName.TabIndex = 0
```

FLoad.Text = ""

Normal_Rating.Text = ""

Emergency_Rating.Text = ""

Shift.Text = ""

Delta.Text = ""

Emergency_Load.Text = ""

Feeders_4kv.Text = ""

Paper_GT40.Text = ""

Paper_LE40.Text = ""

Solid_LT16.Text = ""

Solid_70_75.Text = ""

Solid_GT16.Text = ""

Percent_Paper_GT40.Text = ""

Percent_Paper_LE40.Text = ""

Percent_Solid_LT16.Text = ""

Percent_Solid_70_75.Text = ""

Percent_Solid_GT16.Text = ""

Sec_CNT.Text = ""

E21GT93.Text = ""

E21LE93.Text = ""

R31GT94.Text = ""

R31LE93.Text = ""

OtherStop_LT7.Text = ""

OtherStop7_12.Text = ""

OtherStop_GT12.Text = ""

AllStop.Text = ""

Percent_ALLStop.Text = ""

PJ_LT8.Text = ""

PJ8_16.Text = ""

PJ16_24.Text = ""

'PJ_GT24.Text = ""

PJ24_32.Text = ""

PJ32_40.Text = ""

PJGT40.Text = ""

AllPaper.Text = ""

Percent_AllPaper.Text = ""

SJ_LT8.Text = ""

SJ8_16.Text = ""

SJ16_24.Text = ""

SJ_GT24.Text = ""

AllSolid.Text = ""

Percent_AllSolid.Text = ""

Percent_E21GT93.Text = ""

Percent_E21LE93.Text = ""

Percent_R31GT94.Text = ""

Percent_R31LE93.Text = ""

Percent_OtherStop_LT7.Text = ""

Percent_OtherStop7_12.Text = ""

Percent_OtherStop_GT12.Text = ""

Percent_PJ_LT8.Text = ""

Percent_PJ8_16.Text = ""

Percent_PJ16_24.Text = ""

'Percent_PJ_GT24.Text = ""

Percent_PJ24_32.Text = ""

Percent_PJ32_40.Text = ""

Percent_PJGT40.Text = ""

Percent_SJ_LT8.Text = ""

Percent_SJ8_16.Text = ""

Percent_SJ16_24.Text = ""

Percent_SJ_GT24.Text = ""

Joint_CNT.Text = ""

XFMR_GT40.Text = ""

XFMR30_39.Text = ""

XFMR20_29.Text = ""

XFMR10_19.Text = ""

XFMR0_9.Text = ""

W_XFMR_GT40.Text = ""

W_XFMR30_39.Text = ""

W_XFMR20_29.Text = ""

W_XFMR10_19.Text = ""

W_XFMR0_9.Text = ""

Percent_XFMR_GT40.Text = ""

```
Percent_XFMR30_39.Text = ""

Percent_XFMR20_29.Text = ""

Percent_XFMR10_19.Text = ""

Percent_XFMR0_9.Text = ""

Percent_W_XFMR_GT40.Text = ""

Percent_W_XFMR30_39.Text = ""

Percent_W_XFMR20_29.Text = ""

Percent_W_XFMR10_19.Text = ""

Percent_W_XFMR0_9.Text = ""

XFMR_CNT_SUMMED.Text = ""

fJeopardy.Text = ""

fCNT.Text = ""

ACTMtbf.Text = ""

'Other stats fill

'StDevMTBF.Text = dr("StDevMTBF")

CountOA.Text = ""

CountCIOA.Text = ""

CountAllOutages.Text = ""
```

```
LastAutoDate.Text = ""

DaysLastOA.Text = ""

LastScheduledWorkDate.Text = ""

LastSuccessfulHipotDate.Text = ""

'all attributes from resetnewattributes set to blank.

mon_Load.Text = ""

mon_Normal_Rating.Text = ""

mon_Emergency_Rating.Text = ""

mon_Shift.Text = ""

mon_Delta.Text = ""

mon_Emergency_Load.Text = ""

mon_4kv.Text = ""

mon_Paper_GT40.Text = ""

mon_Paper_LE40.Text = ""

mon_Solid_LT16.Text = ""

mon_Solid_70_75.Text = ""

mon_Solid_GT16.Text = ""

mon_Percent_Paper_GT40.Text = ""
```

mon_Percent_Paper_LE40.Text = ""

mon_Percent_Solid_LT16.Text = ""

mon_Percent_Solid_70_75.Text = ""

mon_Percent_Solid_GT16.Text = ""

mon_Sec_CNT.Text = ""

mon_E21GT93.Text = ""

mon_E21LE93.Text = ""

mon_R31GT94.Text = ""

mon_R31LE93.Text = ""

mon_OtherStop_LT7.Text = ""

mon_OtherStop7_12.Text = ""

mon_OtherStop_GT12.Text = ""

mon_AllStop.Text = ""

mon_Percent_ALLStop.Text = ""

mon_PJ_LT8.Text = ""

mon_PJ8_16.Text = ""

mon_PJ16_24.Text = ""

'mon_PJ_GT24.Text = ""

```
mon_PJ24_32.Text = ""

mon_PJ32_40.Text = ""

mon_PJGT40.Text = ""

mon_AllPaper.Text = ""

mon_Percent_AllPaper.Text = ""

mon_SJ_LT8.Text = ""

mon_SJ8_16.Text = ""

mon_SJ16_24.Text = ""

mon_SJ_GT24.Text = ""

mon_AllSolid.Text = ""

mon_Percent_AllSolid.Text = ""

mon_Percent_E21GT93.Text = ""

mon_Percent_E21LE93.Text = ""

mon_Percent_R31GT94.Text = ""

mon_Percent_R31LE93.Text = ""

mon_Percent_OtherStop_LT7.Text = ""

mon_Percent_OtherStop7_12.Text = ""

mon_Percent_OtherStop_GT12.Text = ""
```

mon_Percent_PJ_LT8.Text = ""

mon_Percent_PJ8_16.Text = ""

mon_Percent_PJ16_24.Text = ""

' mon_Percent_PJ_GT24.Text = ""

mon_Percent_PJ24_32.Text = ""

mon_Percent_PJ32_40.Text = ""

mon_Percent_PJGT40.Text = ""

mon_Percent_SJ_LT8.Text = ""

mon_Percent_SJ8_16.Text = ""

mon_Percent_SJ16_24.Text = ""

mon_Percent_SJ_GT24.Text = ""

mon_Joint_CNT.Text = ""

mon_XFMR_GT40.Text = ""

mon_XFMR30_39.Text = ""

mon_XFMR20_29.Text = ""

mon_XFMR10_19.Text = ""

mon_XFMR0_9.Text = ""

mon_W_XFMR_GT40.Text = ""

mon_W_XFMR30_39.Text = ""

mon_W_XFMR20_29.Text = ""

mon_W_XFMR10_19.Text = ""

mon_W_XFMR0_9.Text = ""

mon_Percent_XFMR_GT40.Text = ""

mon_Percent_XFMR30_39.Text = ""

mon_Percent_XFMR20_29.Text = ""

mon_Percent_XFMR10_19.Text = ""

mon_Percent_XFMR0_9.Text = ""

mon_Percent_W_XFMR_GT40.Text = ""

mon_Percent_W_XFMR30_39.Text = ""

mon_Percent_W_XFMR20_29.Text = ""

mon_Percent_W_XFMR10_19.Text = ""

mon_Percent_W_XFMR0_9.Text = ""

mon_XFMR_CNT_SUMMED.Text = ""

Jeopardy.Text = ""

CNT.Text = ""

LPW.Text = ""

```
            AVG_LPW.Text = ""

            fAVG_LPW.Text = ""

            fLPW.Text = ""









            Reset_MTBF_Load()

        End If

    End Sub



    Protected Sub CAPTWizard_ActiveStepChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CAPTWizard.ActiveStepChanged

        PILCCablesWorstStopJoints.Visible = False

        UpdateAttributesPILC.Visible = False

        UpdateAttributesRel.Visible = False

        UpdateAttributesLoad.Visible = False
```

UpdateAttributesElast.Visible = False

UpdateAttributesStops.Visible = False

PILCCablesWithStopJoints.Visible = False

PILCMTBFInfo.Visible = False

Panel9.Visible = False

PILCMTBFTotalCostInfo.Visible = False

loadStrategyInfo.Visible = False

Runs.Visible = False

downloadtoExcelPanel.Visible = False

ovruns.Visible = False

ovsec.Visible = False

GVRelElastPanel.Visible = False

RelGVElastPanel.Visible = False

XLP.Visible = False

GVRelStopsPanel.Visible = False

MTBF_NA.Visible = False

MTBF_LR_NA.Visible = False

MTBF_Rel_NA.Visible = False

```
Reset_MTBF_Load()

Reset_MTBF_PILC()

Reset_MTBF_Rel()

Runs_Select.Visible = False

sectionschangedPILCPanel.Visible = False



'lblTarget.Text = ""

'lblCostPercentage.Text = ""

'lblAchieved.Text = ""

'lblCostPercentageValue.Text = ""




'SectionsChangedPILC.Text = ""

' PILCCalculationReset()



TargetCostCheck.Visible = False

'SectionsChangedPILC.Text = ""

' actualTotalCostPILC.Text = ""
```

```
' actualPercentPILC.Text = ""

' GVPILCStopJoints.Visible = False

' Runs.Visible = False

' UpdateAttributesPILC.Visible = False

'  CommitChangesPILC.Visible = False

'CheckCommitPILC.Checked = False

'PILCCablesWithStopJoints.Visible = False

' GVPILCStopJoints.Visible = False

' End If

CalculateNewLoadMTBF.Enabled = True

' If CAPTWizard.ActiveStepIndex = 2 Then

'If StrategyLoadDropDown.SelectedIndex > 2 Then CalculateNewLoadMTBF.Enabled =
False

'End If

'If CAPTWizard.ActiveStepIndex.Equals(4) Then Exit Sub



'RblTotalCostPercentagePILC.SelectedIndex = 1
```

```
End Sub

'--method added on 2-jan-2008 ends here.



Protected Sub CAPTWizard_SideBarButtonClick(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.WizardNavigationEventArgs) Handles CAPTWizard.SideBarButtonClick

  If cmbSearchFeeder.Items.Count < 1 Then

    e.Cancel = True

  End If

  If e.NextStepIndex.Equals(9) Then

    BindAttributes1()

  End If



End Sub
```

```
'        Protected Sub RblCostPercentage_SelectedIndexChanged(ByVal sender As Object,
ByVal e 'As System.EventArgs) Handles RblCostPercentageRel.SelectedIndexChanged

'        If RblCostPercentageRel.SelectedIndex > 0 Then

'          RelTotalCost.Enabled = False

'          RelPercent.Enabled = True

'     Else

'          RelTotalCost.Enabled = True

'        RelPercent.Enabled = False

'

'

'        End If

'

'     End Sub

'     '*--added by shringika to include totalcost/percent radio button selectioncase
```

```
Protected Sub RblTotalCostPercentagePILC_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles RblTotalCostPercentagePILC.SelectedIndexChanged

    PercentPILC.SelectedIndex = 0

    TotalCostPILC.Text = ""

    actualTotalCostPILC.Text = ""

    SectionsChangedPILC.Text = ""

    actualPercentPILC.Text = ""

    Runs.Visible = False

    downloadtoExcelPanel.Visible = False

    PILCCablesWithStopJoints.Visible = False

    PILCCablesWorstStopJoints.Visible = False

    Runs_Select.Visible = False

    CostPercentPanel.Visible = True

    If RblTotalCostPercentagePILC.SelectedIndex < 1 Then

      'targetcost case

      TotalCostPILC.Enabled = True

      PercentPILC.Enabled = False
```

```
    End If


    If RblTotalCostPercentagePILC.SelectedIndex < 2 Then
      ' percent case
      TargetCostCheck.Visible = False
      TotalCostPILC.Enabled = False
      PercentPILC.Enabled = True


    Else
      'select runs case
      CostPercentPanel.Visible = False
      TargetCostCheck.Visible = False
      TotalCostPILC.Enabled = False
      PercentPILC.Enabled = False
      CalculateCostPILC_SelectRuns()


    End If
  End Sub
```

```
'---modification by shringika to include stop joints calculation in tier2

Private Sub StopJointsCalculation()


  Dim dv1 As DataView = New DataView(GetRunStopTable(Feeder.Text))

  Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))

  'runstop.Visible = True

  If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

    If CheckCommitPILC.Checked Then

      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - duplicate_sections_list)

    Else

      'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - visited_sections_count)

      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2))
```

```
                End If

                Total_Cost_PILC()

            Else


                If CheckCommitPILC.Checked Then

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - duplicate_sections_list)

                Else

                    'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - visited_sections_count)

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2))

                End If

                Total_Cost_PILC()


            End If

            If SectionsChangedPILC.Text = "0" Then UpdateAttributesPILC.Visible = False
```

```
' Code for hierarchgrid


Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PILCJoint_Sections")

dt1 = dv1.ToTable()

dt2 = dv2.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("PILCJoint_Sections", dcc1, dcc2, False)

dss.Relations.Add(drr)


'bind the DataSet to the HierarGrid
```

```
GVPILCStopJoints.DataSource = dss

GVPILCStopJoints.DataMember = dss.Tables(0).TableName()

GVPILCStopJoints.DataBind()



Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

   GVPILCStopJoints.Columns(9).Visible = False

End If

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVPILCStopJoints.Items.Count) - 1

   If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVPILCStopJoints.Items(visited_run_grid).Cells(1).Text),     Nothing))
Then
```

```
                GVPILCStopJoints.Items(visited_run_grid).ForeColor = Drawing.Color.Red


            End If

        Next


        PILCCablesWithStopJoints.Visible = True

        GVPILCStopJoints.Visible = True

        downloadtoExcelPanel.Visible = True




    End Sub


    Private Function Replace_Stops_percent_PILC(ByVal percent As String, ByVal runs As
DataView, ByVal joints As DataView) As Integer


        Dim ovl_cnt As Int32
```

```
Dim i, j As Integer

Dim total_joints, total_sections, total_stops As Integer

Dim running_total, running_sec_total As Integer

Dim per As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer

Dim visitedRuns As New ArrayList

Dim visitedJoints As New ArrayList

Dim dupcount As Integer



'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)

'If percent.Length = 0 Then

'percent = "100"
```

```
' End If

per = Int32.Parse(percent)



running_sec_total = 0

actualPercentPILC.Text = "0"



'If (per = 100) Then

'actualPercentPILC.Text = "100"

'section count is from attributes in funtion so can be different, bummer!

' sec_changed = Replace_Paper()

'

'If (runs.Table.Rows.Count > 0) Then

'Return Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'

'Return sec_changed

'Else : Return 0

'End If

'End If
```

```
If (runs.Table.Rows.Count > 0) Then

  'pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))

  'for now do it this way

  'total_stops = Int32.Parse(mon_AllStop.Text)

  'total_joints = Int32.Parse(mon_Joint_CNT.Text)

  total_stops = Int32.Parse(AllStop.Text)

  total_joints = Int32.Parse(Joint_CNT.Text)

  running_total = 0



  For i = 0 To (runs.Table.Rows.Count - 1)



    runId = runs.Item(i).Item("Run_ID")

    If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For
```

```
dupcount = 0

'Joints can be in more than one run! Have to check that

'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND stop = 'S'"

For j = 0 To joints.Count - 1

   If visitedJoints.Contains(joints(j)("Joint_ID")) Then

      dupcount = dupcount + 1

   Else

      visitedJoints.Add(joints(j)("Joint_ID"))

   End If



   ' If visitedGlbJoints.Contains(joints(j)("Joint_ID")) Then running_total =
running_total - 1



Next
```

```
joints.RowFilter = Nothing

running_total = running_total + runs.Item(i).Item("Stops") - dupcount

running_sec_total = running_sec_total + runs.Item(i).Item("length")

running_percent = running_total / total_stops * 100.0

running_tot_percent = running_total / total_joints * 100.0

actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))



'do it early now....

If (running_percent >= per) Then

   running_sec_total = running_sec_total - runs.Item(i).Item("length")

   running_total = running_total - (runs.Item(i).Item("Stops") - dupcount)

   running_percent = running_total / total_stops * 100.0

   running_tot_percent = running_total / total_joints * 100.0

   runId = runs.Item(i).Item("Run_ID")



   actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

   Exit For
```

```
End If

runId = runs.Item(i).Item("Run_ID")

runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

'use this to debug dups

'runs.Item(i).Item("ptsec") = dupcount

'I dont think this check does anything... AGB

If (Not (visitedRuns.Contains(runId))) Then

   visitedRuns.Add(runId)

   If CheckCommitPILC.Checked Then
```

```
                    Update_Attributes(runId, False)

                    If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)


                  End If

                  visited_sections_check(runId, False)

                End If


                'If (running_percent >= per) Then Exit For


              Next

            End If


            ovl_cnt = running_sec_total
```

```
        Return ovl_cnt



    End Function



    Protected Sub GVPILCStopJoints_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles GVPILCStopJoints.SelectedIndexChanged

        Dim dt As DataTable

        Dim Run As Integer

        Dim feeder As String

        feeder = cmbSearchFeeder.SelectedValue

        Run = Int32.Parse(GVPILCStopJoints.SelectedItem.Cells(1).Text)



        dt = GetSectionInRunTable(Run, feeder)



        Dim struct As String

        struct = dt.Rows(0).Item("from_struct")

        Session("struct") = struct
```

```
Dim URL As String

URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"

Dim MyScript As New System.Text.StringBuilder
With MyScript
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
```

```
ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub



Private Sub GVpilcstopjoints_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVPILCStopJoints.TemplateSelection

e.TemplateFilename = "Templates\\RelJointSections.ascx"

End Sub



Protected Sub GVPILCStopJointTypes_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles GVPILCStopJointTypes.SelectedIndexChanged

Dim dt As DataTable

Dim Run As Integer

Dim feeder As String

feeder = cmbSearchFeeder.SelectedValue

Run = Int32.Parse(GVPILCStopJointTypes.SelectedItem.Cells(2).Text)
```

```
dt = GetSectionInRunTable(Run, feeder)


Dim struct As String

struct = dt.Rows(0).Item("from_struct")

Session("struct") = struct


Dim URL As String


URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"


Dim MyScript As New System.Text.StringBuilder

With MyScript

   .Append("<script type='text/javascript'>")

   .Append("window.open('" & URL & "');")

   .Append("</script>")
```

```
        End With

        'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

        ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

        'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

    End Sub

    Private Sub GVpilcstopjointtypes_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVPILCStopJointTypes.TemplateSelection

        e.TemplateFilename = "Templates\\RelJointSections.ascx"

    End Sub
```

```
Protected Sub PercentPILC_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PercentPILC.SelectedIndexChanged

    PILCCalculationReset()

End Sub



Protected Sub CommitChangesPILC_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesPILC.Click

    TargetCostCheck.Visible = False



    CheckCommitPILC.Checked = True

    If RblTotalCostPercentagePILC.SelectedIndex = 2 Then

      CalculateCostPILC_SelectRunsCommitChanges()

      CheckCommitPILC.Checked = False
```

```
            'CommitChangesPILC.Enabled = False

            UpdateAttributesPILC.Visible = False

            Session("strategies_pilc") = StrategyPILCDropDown.SelectedValue



            Exit Sub

        End If



        CalculateCostPILC_CommitChanges()

        CheckCommitPILC.Checked = False

        'CommitChangesPILC.Enabled = False

        UpdateAttributesPILC.Visible = False

        Session("strategies_pilc") = StrategyPILCDropDown.SelectedValue




    End Sub



    Protected Sub CalculateCostPILC_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostPILC.Click
```

```
        CheckCommitPILC.Checked = False

        TargetCostCheck.Visible = False


        If RblTotalCostPercentagePILC.SelectedIndex > 1 Then

            CalculateCostPILC_SelectRunsCommitChanges()

            CostPercentPanel.Visible = True

            UpdateAttributesPILC.Visible = True

            CommitChangesPILC.Visible = True

            downloadtoExcelPanel.Visible = True

            Exit Sub

        End If


        CalculateCostPILC_CommitChanges()

    End Sub


    Protected Sub TotalCostPILC_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TotalCostPILC.TextChanged

        PILCCalculationReset()
```

```
End Sub

Private Sub PILCCalculationReset()

  If cmbSearchFeeder.Items.Count > 0 Then



    TargetCostCheck.Visible = False

    SectionsChangedPILC.Text = ""

    actualTotalCostPILC.Text = ""

    actualPercentPILC.Text = ""

    GVPILCStopJoints.Visible = False

    Runs.Visible = False

    downloadtoExcelPanel.Visible = False

    UpdateAttributesPILC.Visible = False

    CommitChangesPILC.Visible = False

    CheckCommitPILC.Checked = False

    PILCCablesWithStopJoints.Visible = False

    GVPILCStopJoints.Visible = False
```

```
RankBasePILC.Text = ""

FOTBasePILC.Text = ""

MTBFBasePILC.Text = ""

RankNewPILC.Text = ""

RankChangePILC.Text = ""

MTBFNewPILC.Text = ""

MTBFChangePILC.Text = ""

FOTNewPILC.Text = ""

FOTChangePILC.Text = ""



PILCCablesWorstStopJoints.Visible = False
```

```
        End If

    End Sub


    Protected Sub CommitChangesLoad_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesLoad.Click

        CheckCommitLoad.Checked = True

        CalculateCostLoad_CommitChanges()

        Session("strategies_load") = StrategyLoadDropDown.SelectedValue

        CheckCommitLoad.Checked = False

        'CommitChangesLoad.Enabled = False

        UpdateAttributesLoad.Visible = False


    End Sub
```

'Protected Sub CommitChangesRel_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles CommitChangesPILC.Click

' CheckCommitRel.Checked = True

' CalculateCostRel_CommitChanges()

' CheckCommitRel.Checked = False

'CommitChangesRel.Enabled = False

' UpdateAttributesRel.Visible = False

' End Sub

Protected Sub CommitChangesRel_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles CommitChangesRel.Click

CheckCommitRel.Checked = True

CalculateCostRel_CommitChanges()

CheckCommitRel.Checked = False

```
'CommitChangesRel.Enabled = False

UpdateAttributesRel.Visible = False



End Sub

Private Function Replace_Stops_TotalCost_PILC(ByVal totalcost As String, ByVal runs As
DataView, ByVal joints As DataView) As Integer

'needs changes for PILC sections replacement selection tier's StopJointsStrategy based on
toal Cost

'added on 06/23/2008 shringika

Dim ovl_cnt As Int32



Dim i, j, k As Integer

Dim total_sections, pilc_sections_count, running_total, running_sec_total, total_stops,
total_joints As Integer

Dim total_cost As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer

Dim cost_per_section As Integer

Dim visitedRuns As New ArrayList
```

```
Dim visitedJoints As New ArrayList

Dim dupcount As Integer




total_stops = Int32.Parse(AllStop.Text)

total_joints = Int32.Parse(Joint_CNT.Text)

running_total = 0

'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)

total_cost = Int32.Parse(totalcost)

cost_per_section = Int32.Parse(CostPerSectionPILC.Text)



Dim total_replaced_sections As Integer

total_replaced_sections = Math.Floor(total_cost / cost_per_section)



pilc_sections_count = 0
```

```
If (runs.Table.Rows.Count > 0) Then

    'count the number of pilc sections by running loop earlier so that it can be used for
getting running percent

    'commented by shringika on 06/24/2008

    ' pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))

    'why it come to 154 when it should be 48 for '4x40' need to check

    For k = 0 To (runs.Table.Rows.Count - 1)

      pilc_sections_count = pilc_sections_count + runs.Item(k).Item("length")

    Next



    lblsections.Text = pilc_sections_count



    For i = 0 To (runs.Table.Rows.Count - 1)
```

"""""inclusiomn of modifications by Albert into this code for duplicate joints 06/26/2008

runId = runs.Item(i).Item("Run_ID")

If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then Continue For

dupcount = 0

'Joints can be in more than one run! Have to check that

'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND stop = 'S'"

For j = 0 To joints.Count - 1

If visitedJoints.Contains(joints(j)("Joint_ID")) Then

dupcount = dupcount + 1

Else

visitedJoints.Add(joints(j)("Joint_ID"))

End If

```
Next



joints.RowFilter = Nothing

"""""""""""""""modifcation for joints ends here

running_total = running_total + runs.Item(i).Item("stops") - dupcount

'total # of stops

running_sec_total = running_sec_total + runs.Item(i).Item("length")

'total sections

If (running_sec_total > total_replaced_sections) Then

   'running_total = running_total + runs.Item(i).Item("Stops") - dupcount
```

```
    running_total = running_total - (runs.Item(i).Item("stops") - dupcount)

    running_sec_total = running_sec_total - runs.Item(i).Item("length")

    running_percent = running_total / total_stops * 100.0

    actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

    Exit For

End If



running_percent = running_total / total_stops * 100.0

running_tot_percent = running_sec_total / total_sections * 100.0

runId = runs.Item(i).Item("Run_ID")

runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)



If (Not (visitedRuns.Contains(runId))) Then
```

```
                visitedRuns.Add(runId)

                If CheckCommitPILC.Checked Then

                    Update_Attributes(runId, False)

                    If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)

                End If

                visited_sections_check(runId, False)

            End If




            actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

            If (running_sec_total.Equals(total_replaced_sections)) Then Exit For



        Next

    End If
```

```
        ovl_cnt = running_sec_total

        'actualPercentPILC.Text = Math.Round(running_total / pilc_sections) * 100.0




        Return ovl_cnt



        'Might want to move this to a separate sub

        'SectionsChangedPILC.Text = Convert.ToString(ovl_cnt)

        'Total_Cost_PILC()

    End Function



    'Protected Sub RelPercent_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RelPercent.SelectedIndexChanged

    ' If cmbSearchFeeder.Items.Count > 0 Then




    ' SectionsChangedRel.Text = ""

    ' TotalCostRel.Text = ""
```

```
' ELAS.Visible = False

' runstop.Visible = False

'  XLP.Visible = False



'   UpdateAttributesRel.Visible = False

'   CommitChangesRel.Visible = False

'  CheckCommitRel.Checked = False



'End If

' End Sub




Private Sub Total_Cost_RelElastimold()

   TotalCostRelElastimold.Text                                                    =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRelElastimold.Text)                *
Convert.ToDouble(CostPerSectionRelElastimold.Text)), 2))

End Sub
```

```
Protected Sub CaculateCostRelElastimold_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CaculateCostRelElastimold.Click

    CheckCommitElast.Checked = False

    visited_sections_count = 0

    CalculateCostRelElastimold_CommitChanges()

End Sub

Private Sub CalculateCostRelElastimold_CommitChanges()

    If cmbSearchFeeder.Items.Count > 0 Then



        BindAttributes()

        BindRankingLPW()

        BindFOT()

        ' Reset_New_Attributes()

        UpdateAttributesElast.Visible = True

        CommitChangesElast.Visible = True

        GVRelElastPanel.Visible = True
```

```
RelGVElastPanel.Visible = True

RelGVElast.Visible = True

GVRelElastStops.Visible = True



'ELASTIMOLDS

Dim dv1 As DataView = New DataView(GetElastimoldTable(Feeder.Text))

Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))



' SectionsChangedRelElastimold.Text =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2))

'Total_Cost_RelElastimold()



If CheckCommitElast.Checked Then

  SectionsChangedRelElastimold.Text =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2) - duplicate_sections_list)

  Total_Cost_RelElastimold()
```

```
                Else

                    '                                   SectionsChangedRelElastimold.Text =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2) - visited_sections_count)

                    SectionsChangedRelElastimold.Text                                   =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2))

                    Total_Cost_RelElastimold()

                End If



                If SectionsChangedRelElastimold.Text = "0" Then UpdateAttributesElast.Visible =
False



                ' Code for hierarchgrid



                Dim dss As DataSet = New DataSet()

                Dim dt1 As DataTable = New DataTable("RelRuns")

                Dim dt2 As DataTable = New DataTable("RelJoint_Sections")



                dt1 = dv1.ToTable()

                dt2 = dv2.ToTable()
```

```
dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("RelElastJoint_Sections", dcc1, dcc2,
False)

dss.Relations.Add(drr)



'bind the DataSet to the HierarGrid

GVRelElastStops.DataSource = dss

GVRelElastStops.DataMember = dss.Tables(0).TableName()

GVRelElastStops.DataBind()

Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);
```

```
If Equals(m_deltaMTBF, "no") Then
    GVRelElastStops.Columns(9).Visible = False
End If

Dim visited_run_grid As Integer
For visited_run_grid = 0 To (GVRelElastStops.Items.Count) - 1
    If Not (Equals(VisitedGlbRuns.Items.FindByText(GVRelElastStops.Items(visited_run_grid).Cells(1).Text), Nothing)) Then
        GVRelElastStops.Items(visited_run_grid).ForeColor = Drawing.Color.Red
    End If
Next
```

```
        End If

    End Sub



    Protected Sub GVRelElast_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVRelElastStops.TemplateSelection

        e.TemplateFilename = "Templates\\RelJointSections.ascx"

    End Sub



    Protected Sub CommitChangesElast_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesElast.Click

        CheckCommitElast.Checked = True

        CalculateCostRelElastimold_CommitChanges()

        CheckCommitElast.Checked = False

        'CalculateCostRelElastimold_CommitChanges()



        'CommitChangesRelElast.Enabled = False

        UpdateAttributesElast.Visible = False
```

```
End Sub

Protected Sub CalculateNewRelMTBFElast_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewRelMTBFElast.Click
    MTBF_NA_RelElast.Visible = False
    Calculate_New_RelElast()

End Sub

Private Sub Calculate_New_RelElast()
    'modified by shringika on dec 21'2007 included if case
    If cmbSearchFeeder.Items.Count > 0 Then

        Dim dt As DataTable = New DataTable
        Dim Rank As Integer
        'have to work on this query more
```

```
'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '"
& cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '"
& cmbSearchFeeder.SelectedValue & "'and rel_percent =  '" & RelPercentElastimold.Text & "' "

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

  MTBF_NA_RelElast.Visible = True

  Exit Sub

End If
```

```
Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary



RankBaseRelElast.Text = dr("OldRank")

Rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression (New 6-08)

'MTBFBaseRelElast.Text = (Rank * 0.3386 + 165.96).ToString("F0")

'modified on 01/04/2009 for picking up rank from rerank_cost_score



'MTBFBaseRel.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

MTBFBaseRelElast.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseRelElast.Text = Convert.ToString(Math.Round(Math.Max((Rank * -0.0095 +
6), 0.0), 0))



'TEMP!!!!!!!!!!!!!
```

```
' RankNewRelElast.Text = Convert.ToString(rand(Rank, Rank + 100))

'modified on 01/04/2009 for picking up rank from rerank_cost_score

RankNewRelElast.Text = dr("rank")



'RankNewPILC.Text = "930"



RankChangeRelElast.Text = Convert.ToString(Int32.Parse(RankNewRelElast.Text) - Rank)

'added on 01/04/2009 for picking up new mtbf and delta from database , not by calculation

MTBFNewRelElast.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

MTBFChangeRelElast.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'conversion of rank to MTBF using regression

'MTBFNewRel.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

'MTBFChangeRel.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'commented on 01/04/2009 for picking up values from database

'MTBFNewRelElast.Text = (Int32.Parse(RankNewRelElast.Text) * 0.3386 + 165.96).ToString("F0")
```

```
'MTBFChangeRelElast.Text =
Convert.ToString(Convert.ToDouble(MTBFNewRelElast.Text) - Convert.ToDouble(MTBFBaseRelElast.Text))

FOTNewRelElast.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewRelElast.Text) * -0.0095 + 6), 0.0), 0))

FOTChangeRelElast.Text =
Convert.ToString(Convert.ToDouble(FOTBaseRelElast.Text) - Convert.ToDouble(FOTNewRelElast.Text))


myCommand = Nothing
myConnection.Close()
myConnection = Nothing
If MTBFChangeRelElast.Text.Length = 0 Then
  MTBF_NA_RelElast.Visible = True
Else
  MTBF_NA_RelElast.Visible = False


End If
```

```
        End If

        'end if added by shringika on 21dec 2007 as part of validation

    End Sub


    Protected Sub GVRelStops_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GVRelStops.SelectedIndexChanged

        Dim dt As DataTable

        Dim Run As Integer

        Dim feeder As String

        feeder = cmbSearchFeeder.SelectedValue

        Run = Int32.Parse(GVRelStops.SelectedItem.Cells(1).Text)



        dt = GetSectionInRunTable(Run, feeder)



        Dim struct As String

        struct = dt.Rows(0).Item("from_struct")

        Session("struct") = struct
```

```
Dim URL As String

URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"

Dim MyScript As New System.Text.StringBuilder
With MyScript
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
```

```
ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub



Protected Sub GVRelStops_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles GVRelStops.TemplateSelection

e.TemplateFilename = "Templates\\RelJointSections.ascx"

End Sub



Protected Sub CalculateCostRelStops_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostRelStops.Click

CheckCommitStops.Checked = False

CalculateCostRelStops_CommitChanges()

End Sub

Private Sub CalculateCostRelStops_CommitChanges()

If cmbSearchFeeder.Items.Count > 0 Then
```

```
BindAttributes()

BindRankingLPW()

BindFOT()

'Reset_New_Attributes()



CommitChangesStops.Visible = True

GVRelStops.Visible = True

UpdateAttributesStops.Visible = True

GVRelStopsPanel.Visible = True

'STOP JOINTS

Dim dv1 As DataView = New DataView(GetRunStopTable(Feeder.Text))

Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))

'runstop.Visible = True




If CheckCommitStops.Checked Then
```

```
                SectionsChangedRelStops.Text =
Convert.ToString(Replace_Stops_percent(RelPercentStops.Text, dv1, dv2) - duplicate_sections_list)

            Else

                ' SectionsChangedRelStops.Text =
Convert.ToString(Replace_Stops_percent(RelPercentStops.Text, dv1, dv2) - visited_sections_count)

                SectionsChangedRelStops.Text =
Convert.ToString(Replace_Stops_percent(RelPercentStops.Text, dv1, dv2))

            End If

            If SectionsChangedRelStops.Text = "0" Then UpdateAttributesStops.Visible = False



            Total_Cost_RelStops()



            ' Code for hierarchgrid



            Dim dss As DataSet = New DataSet()

            Dim dt1 As DataTable = New DataTable("RelRuns")

            Dim dt2 As DataTable = New DataTable("RelJoint_Sections")

            dt1 = dv1.ToTable()

            dt2 = dv2.ToTable()
```

```
dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("RelJoint_Sections", dcc1, dcc2, False)

dss.Relations.Add(drr)



'bind the DataSet to the HierarGrid

GVRelStops.DataSource = dss

GVRelStops.DataMember = dss.Tables(0).TableName()

GVRelStops.DataBind()

Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);
```

```
If Equals(m_deltaMTBF, "no") Then

  GVRelStops.Columns(9).Visible = False

End If


Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVRelStops.Items.Count) - 1

  If Not (Equals(VisitedGlbRuns.Items.FindByText(GVRelStops.Items(visited_run_grid).Cells(1).Text), Nothing)) Then

    GVRelStops.Items(visited_run_grid).ForeColor = Drawing.Color.Red

  End If

Next

End If
```

```
'end if added on 21-dec-2007 by shringika

End Sub

Private Sub Total_Cost_RelStops()

TotalCostRelStops.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRelStops.Text) *
Convert.ToDouble(CostperSectionRelStops.Text)), 2))

End Sub


Protected Sub CommitChangesStops_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesStops.Click

UpdateAttributesStops.Visible = False



CommitChangesStops.Visible = False

CheckCommitStops.Checked = True

GVRelStopsPanel.Visible = False

CalculateCostRelStops_CommitChanges()

'CommitChangesRel.Enabled = False

UpdateAttributesStops.Visible = False
```

```
End Sub

Protected Sub CalculateNewRelMTBFStops_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewRelMTBFStops.Click
    MTBF_NA_RelStops.Visible = False
    Calculate_New_RelStops()
End Sub
Private Sub Calculate_New_RelStops()

    'modified by shringika on dec 21'2007 included if case
    If cmbSearchFeeder.Items.Count > 0 Then

      MTBF_NA_RelStops.Visible = False
      If SectionsChangedRelStops.Text = "" Then
        MTBF_NA_RelStops.Visible = True
        Return
```

```
End If

Dim feeder_id As String

If cmbSearchFeeder.SelectedValue.Length < 5 Then

   feeder_id = "0" + cmbSearchFeeder.SelectedValue

Else

   feeder_id = cmbSearchFeeder.SelectedValue

End If



If Session("strategies_pilc") Is Nothing Then Session("strategies_pilc") = "0"

Dim pilc_strategy As String = CType(Session("strategies_pilc"), String)



If Session("strategies_load") Is Nothing Then Session("strategies_load") = "0"

Dim load_strategy As String = CType(Session("strategies_load"), String)



Dim dt As DataTable = New DataTable

Dim Rank As Integer

Dim sql As String

'have to work on this query more
```

```
'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

If load_strategy.Length < 2 Then

  If pilc_strategy.Length < 2 Then

    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "' and rel_percent = '" & RelPercentStops.Text & "'"


  Else

    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "' AND pilc_strategy='" & pilc_strategy & "' AND pilc_percent= '" & PercentPILC.SelectedValue & "' AND rel_percent = '" & RelPercentStops.SelectedValue & "'"

  End If

Else


  If pilc_strategy.Length < 2 Then

    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and rel_percent = '" & RelPercentStops.Text & "' AND load_relief_percent='" & StrategyLoadDropDown.SelectedValue & "'"
```

```
                Else

                    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' AND pilc_strategy='" & pilc_strategy & "' AND pilc_percent= '" &
PercentPILC.SelectedValue & "' AND rel_percent = '" & RelPercentStops.SelectedValue & "' AND
load_relief_percent='100'"

                    ' & StrategyLoadDropDown.SelectedValue & """

                End If



            End If



            ' TextBox1.Text = sql



            Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

            Dim myCommand As New SqlDataAdapter(sql, myConnection)

            myCommand.SelectCommand.CommandTimeout = 180

            myCommand.Fill(dt)

            'return if there is no MTBF data computed
```

```
'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_NA_RelStops.Visible = True

   Return

End If



Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary



RankBaseRelStops.Text = dr("OldRank")

Rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression (New 6-08)

'MTBFBaseRelStops.Text = (Rank * 0.3386 + 165.96).ToString("F0")

MTBFBaseRelStops.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")



'MTBFBaseRel.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseRelStops.Text = Convert.ToString(Math.Round(Math.Max((Rank * -0.0095 +
6), 0.0), 0))
```

'modified on 01/04/2009 for picking up rank from rerank_cost_score

RankNewRelStops.Text = dr("rank")

'RankNewRelStops.Text = Convert.ToString(rand(Rank, Rank + 100))

RankChangeRelStops.Text = Convert.ToString(Int32.Parse(RankNewRelStops.Text) - Rank)

'conversion of rank to MTBF using regression

'MTBFNewRel.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

'MTBFChangeRel.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'added on 01/04/2009 for picking up new mtbf and delta from database , not by calculation

```
MTBFNewRelStops.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

MTBFChangeRelStops.Text = Convert.ToDouble(dr("delta")).ToString("F0")



'MTBFNewRelStops.Text = (Int32.Parse(RankNewRelStops.Text) * 0.3386 +
165.96).ToString("F0")

'MTBFChangeRelStops.Text =
Convert.ToString(Convert.ToDouble(MTBFNewRelStops.Text) - Convert.ToDouble(MTBFBaseRelStops.Text))

FOTNewRelStops.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewRelStops.Text) * -0.0095 + 6), 0.0), 0))

FOTChangeRelStops.Text =
Convert.ToString(Convert.ToDouble(FOTBaseRelStops.Text) - Convert.ToDouble(FOTNewRelStops.Text))



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

If MTBFChangeRelStops.Text.Length = 0 Then

  MTBF_NA_RelStops.Visible = True

Else

  MTBF_NA_RelStops.Visible = False
```

```
End If

End If

'end if added by shringika on 21dec 2007 as part of validation

End Sub

Private Sub BindAttributes1()

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

' Dim myCommand As New SqlCommand

' myCommand.CommandType = CommandType.StoredProcedure

' myCommand.CommandText = " Exec UpdateBanksOff_Current"
```

```
'    myCommand.Connection = myConnection



'   myCommand = Nothing

'   myConnection.Close()

'  myConnection = Nothing
```

OpenMainsLTE2_Feeder()

OpenMainsGT2_Feeder()

OpenMainsLTE2_NBY_FCTR_GTE10()

OpenMainsLTE2_NBY_FCTR_GTE5LT10()

OpenMainsGT2_NBY_FCTR_GTE10()

OpenMainsGT2_NBY_FCTR_GTE5LT10()

BanksOff_NBY_FCTR_GTE5LT10()

BanksOff_Feeder()

BanksOff_NBY_FCTR_GTE10()

NetRms_Fuse_Vault_Scan_Feeder()

NetRms_Fuse_NBY_FCTR_GTE10()

NetRms_Fuse_NBY_FCTR_GTE5LT10()

NetRms_SwitchesOpen_Vault_Scan_Feeder()

NetRms_SwitchesOpen_NBY_FCTR_GTE5LT10()

NetRms_SwitchesOpen_NBY_FCTR_GTE10()

NetRms_ClosedDNU_Vault_Scan_Feeder()

NetRms_ClosedDNU_GTE10()

NetRms_ClosedDNU_GTE5LT10()

'PlannedBanksOff_Feeder()

' PlannedBanksOff_Feeder_GTE10()

'PlannedBanksOff_Feeder_GTE5LT10()

NetRms_Missing_Vault_Scan_Feeder()

NetRms_Missing_Vault_GTE5LT10()

NetRms_Missing_Vault_GTE10()

' PlannedOpenMainsLTE2_Feeder()

'PlannedOpenMainsGT2_Feeder()

'PlannedOpenMainsLTE2_GTE10()

'PlannedOpenMainsGT2_GTE10()

' PlannedOpenMainsGT2_GTE5LT10()

' PlannedOpenMainsLTE2_GTE5LT10()

NetRms_ClosedOld_Vault_Scan_Feeder()

NetRms_ClosedUNR_Vault_Scan_Feeder()

NetRms_ClosedUNR_Vault_Scan_GTE5LT10()

NetRms_ClosedUNR_Vault_Scan_GTE10()

NetRms_ClosedOld_Vault_Scan_GTE5LT10()

NetRms_ClosedOld_Vault_Scan_GTE10()

Total_LPW_Calc()

TargetInitialValues_LPW()

'Target_Total_LPW_Calc()

'LPW_Cost_Calculation()

```
'populating the vault details



'code from show vault detail button



VaultDetailsPanel.Visible = True

Check_Target_Values()




Target_Total_LPW_Calc()

LPW_Cost_Calculation()

' LPW_grid()









GridviewdataBindBanksOff()
```

GridViewdataBindOpenSwitches()

GridViewdataBindFuse()

GridViewDataBindClosedOld()

GridViewDataBindClosedUNR()

GridViewDataBindClosedDNU()

GridViewDataBindOpenMainsLTE2()

GridViewDataBindOpenMainsGT2()

GridviewdataBindMissing()

GridviewdataBindBanksOff_NbyFactorGTE10()

GridViewdataBindOpenSwitches_NbyFactorGTE10()

GridViewdataBindFuse_NbyFactorGTE10()

GridViewDataBindClosedOld_NbyFactorGTE10()

GridViewDataBindClosedUNR_NbyFactorGTE10()

GridViewDataBindClosedDNU_NbyFactorGTE10()

GridViewDataBindOpenMainsLTE2_NbyFactorGTE10()

GridViewDataBindOpenMainsGT2_NbyFactorGTE10()

GridviewdataBindMissing_NbyFactorGTE10()

GridviewdataBindBanksOff_NbyFactorGTE5LT10()

GridViewdataBindOpenSwitches_NbyFactorGTE5LT10()

GridViewdataBindFuse_NbyFactorGTE5LT10()

GridViewDataBindClosedOld_NbyFactorGTE5LT10()

GridViewDataBindClosedUNR_NbyFactorGTE5LT10()

GridViewDataBindClosedDNU_NbyFactorGTE5LT10()

GridViewDataBindMissing_NbyFactorGTE5LT10()

GridViewDataBindOpenMainsLTE2_NbyFactorGTE5LT10()

GridViewDataBindOpenMainsGT2_NbyFactorGTE5LT10()

IRlpw.Text = fLPW.Text

IR_Scandatetime()

End Sub

```
Private Sub lpwreset()

    Total_LPW_Calc()




    TargetInitialValues_LPW()

    'Target_Total_LPW_Calc()

    'LPW_Cost_Calculation()




    'populating the vault details



    'code from show vault detail button



    VaultDetailsPanel.Visible = True
```

' Check_Target_Values()

Target_Total_LPW_Calc()

LPW_Cost_Calculation()

' LPW_grid()

GridviewdataBindBanksOff()

GridViewdataBindOpenSwitches()

GridViewdataBindFuse()

GridViewDataBindClosedOld()

GridViewDataBindClosedUNR()

GridViewDataBindClosedDNU()

GridViewDataBindOpenMainsLTE2()

GridViewDataBindOpenMainsGT2()

GridviewdataBindMissing()

GridviewdataBindBanksOff_NbyFactorGTE10()

GridViewdataBindOpenSwitches_NbyFactorGTE10()

GridViewdataBindFuse_NbyFactorGTE10()

GridViewDataBindClosedOld_NbyFactorGTE10()

GridViewDataBindClosedUNR_NbyFactorGTE10()

GridViewDataBindClosedDNU_NbyFactorGTE10()

GridViewDataBindOpenMainsLTE2_NbyFactorGTE10()

GridViewDataBindOpenMainsGT2_NbyFactorGTE10()

GridViewDataBindMissing_NbyFactorGTE10()

GridviewdataBindBanksOff_NbyFactorGTE5LT10()

GridViewdataBindOpenSwitches_NbyFactorGTE5LT10()

```
    GridViewdataBindFuse_NbyFactorGTE5LT10()

    GridViewDataBindClosedOld_NbyFactorGTE5LT10()

    GridViewDataBindClosedUNR_NbyFactorGTE5LT10()

    GridViewDataBindClosedDNU_NbyFactorGTE5LT10()

    GridViewDataBindOpenMainsLTE2_NbyFactorGTE5LT10()

    GridViewDataBindOpenMainsGT2_NbyFactorGTE5LT10()

    GridViewDataBindMissing_NbyFactorGTE5LT10()

    IRlpw.Text = fLPW.Text



    IR_Scandatetime()

End Sub



'*--modification on 02/14/2008 ends here



'*--added by shringika for LPW calculation



Private Sub BanksOff_NBY_FCTR_GTE10()
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

BanksOff_GTE10.Text = drcheck(dr, "bo_count")
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub BanksOff_NBY_FCTR_GTE5LT10()



    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String
```

```
Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then
   Feeder_name = "0" + Feeder_name
End If


'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
sql = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER = '" & Feeder_name & "'AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"


Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
        Dim myCommand As New SqlDataAdapter(sql, myConnection)
        myCommand.SelectCommand.CommandTimeout = 120
        Dim dt As DataTable = New DataTable
        myCommand.Fill(dt)


        Dim dr As DataRow = dt.Rows(0)
        'NB: Grid views monitor Feeder to re-query!!!
        BanksOff_GTE5LT10.Text = drcheck(dr, "bo_count")




        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If


End Sub
```

```
Private Sub BanksOff_Feeder()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



  'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH] where Feeder_ID = '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

BanksOffFeederCurrent.Text = drcheck(dr, "bo_count")
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsLTE2_Feeder()

    'transformer on feeder

    Dim sql As String



    If cmbSearchFeeder.Items.Count > 0 Then



        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue



        If Feeder_name.Length < 5 Then
```

```
Feeder_name = "0" + Feeder_name

End If

' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name & "'"

'commented on 10/12/2009 based on lpw discussion with Albert for including all vault entries

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name & "'"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name & "'"
```

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSXW_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name & "'"

' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSM_3MONTH_LTE2] where LVSD_Feeder='" & Feeder_name & "'"

' End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

```
        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        OpenMainsLTE2Current.Text = drcheck(dr, "om_count")









        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsGT2_Feeder()

    'transformer on feeder

    Dim sql As String
```

```
If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    ' If cmbSearchFeeder.SelectedValue.Contains("B") Or
cmbSearchFeeder.SelectedValue.Contains("Q") Then

    'Dim sql As String = "SELECT count(*) as bo_count FROM
[Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" &
cmbSearchFeeder.SelectedValue & "'"

    'commented on 10/12/2009 for including new open mains definition
```

'sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

' sql = "SELECT (case when vault_entries is not null then vault_entries else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSXW_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSM_3MONTH_GT2] where LVSD_Feeder='" & Feeder_name & "'"

```
'End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

OpenMainsGT2Current.Text = drcheck(dr, "om_count")



myCommand = Nothing

myConnection.Close()
```

```
        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsGT2_NBY_FCTR_GTE5LT10()

    'transformer on feeder

    Dim sql As String



    If cmbSearchFeeder.Items.Count > 0 Then



        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue



        If Feeder_name.Length < 5 Then

            Feeder_name = "0" + Feeder_name

        End If
```

'If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

'commented on 10/12/2009 for new openmains count criteria

'sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id = '" & Feeder_name & "')"

' sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id = '" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id = '" & Feeder_name & "')"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

```
' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSXW_3MONTH_GT2OM_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSM_3MONTH_GT2OM_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "'"

' End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)
```

```
        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        OMGT2_GTE5LT10.Text = drcheck(dr, "om_count")




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub

Private Sub OpenMainsGT2_NBY_FCTR_GTE10()

    'transformer on feeder

    Dim sql As String



    If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If


'commented on 10/12/2009 for new om count criteria

'sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] WHERE LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in( Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

' sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end ) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] WHERE LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in( Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] WHERE LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in( Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)

                'NB: Grid views monitor Feeder to re-query!!!

                OMGT2_GTE10.Text = drcheck(dr, "om_count")




                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If
```

```
End Sub



Private Sub OpenMainsLTE2_NBY_FCTR_GTE5LT10()
  'transformer on feeder
  Dim sql As String

  If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String
    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then
      Feeder_name = "0" + Feeder_name
    End If
```

'commenetd on 10/12/2009 for new om criteria

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id= '" & Feeder_name & "')"

' sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id= '" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id= '" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

```
        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        OMLTE2_GTE5LT10.Text = drcheck(dr, "om_count")









        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsLTE2_NBY_FCTR_GTE10()
```

```
'transformer on feeder

Dim sql As String


If cmbSearchFeeder.Items.Count > 0 Then



  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue


  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If


  'commenetd on 10/12/2009 for new om criteria

  'sql = "SELECT count(*) as om_count as om_count FROM
[Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" &
Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM
[Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

'sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

OMLTE2_GTE10.Text = drcheck(dr, "om_count")

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If

End Sub

Private Sub NetRms_Fuse_Vault_Scan_Feeder()
    'transformer on feeder
    Dim sql As String
    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
    If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If


sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE LVS_VAULT_CONDITION like '%D%' AND LVS_FEEDER_NO='" & Feeder_name & "'"


Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180
```

```
        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)



        FuseFeederCurrent.Text = drcheck(dr, "fuse_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub NetRms_Fuse_NBY_FCTR_GTE10()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)



                Fuse_GTE10.Text = drcheck(dr, "fuse_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub NetRms_Fuse_NBY_FCTR_GTE5LT10()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



      Dim Feeder_name As String

      Feeder_name = cmbSearchFeeder.SelectedValue



      If Feeder_name.Length < 5 Then

        Feeder_name = "0" + Feeder_name

      End If



      ' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then
```

sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

'sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_Fuse_NBY_FCTR_GTE5LT10] WHERE LVS_FEEDER_NO='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSM_Fuse_NBY_FCTR_GTE5LT10] WHERE LVS_FEEDER_NO='" & Feeder_name & "'"

'End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)



        Fuse_GTE5LT10.Text = drcheck(dr, "fuse_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub NetRms_SwitchesOpen_Vault_Scan_Feeder()

    'transformer on feeder
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    ' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

    'sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE LVS_NWP_STATUS_FLAG like '%B%' AND LVS_FEEDER_NO = '" & Feeder_name & "'"
```

```
sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE ([Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%B%' OR [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%A%' ) AND LVS_FEEDER_NO = '" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

'sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_2WEEK_VAULTSCAN] WHERE LVS_NWP_STATUS_FLAG like '%B%' AND LVS_FEEDER_NO='" & Feeder_name & "'"

' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[NETRMSM_2WEEK_VAULTSCAN] WHERE LVS_NWP_STATUS_FLAG like '%B%' AND LVS_FEEDER_NO='" & Feeder_name & "'"

' End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
        Dim myCommand As New SqlDataAdapter(sql, myConnection)
        myCommand.SelectCommand.CommandTimeout = 180
        Dim dt As DataTable = New DataTable
        myCommand.Fill(dt)


        Dim dr As DataRow = dt.Rows(0)


        OpenSwitchesFeederCurrent.Text = drcheck(dr, "open_count")


        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If


End Sub


Private Sub NetRms_SwitchesOpen_NBY_FCTR_GTE10()
    'transformer on feeder
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    sql = "SELECT count(*) AS opens_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE10] WHERE LVSD_FEEDER= '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



OpenSwitches_GTE10.Text = drcheck(dr, "opens_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub

Private Sub NetRms_SwitchesOpen_NBY_FCTR_GTE5LT10()
```

```
'transformer on feeder

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then



  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue



  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If



  sql = "SELECT count(*) AS opens_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE5LT10] WHERE
```

LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)


OpenSwitches_GTE5LT10.Text = drcheck(dr, "opens_count")


myCommand = Nothing

myConnection.Close()

myConnection = Nothing
```

```
End If


End Sub


Private Sub NetRms_ClosedDNU_Vault_Scan_Feeder()
  'transformer on feeder
  Dim sql As String
  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
  If cmbSearchFeeder.Items.Count > 0 Then


    Dim Feeder_name As String
    Feeder_name = cmbSearchFeeder.SelectedValue


    If Feeder_name.Length < 5 Then
      Feeder_name = "0" + Feeder_name
```

```
End If

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_Failed_Flags]='U' AND LVS_VAULT_CONDITION NOT LIKE '%B%' AND LVS_VAULT_CONDITION NOT LIKE '%A%'AND LVS_FEEDER_NO='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)
```

```
            ClosedDNUCurrent.Text = drcheck(dr, "closed_count")


            myCommand = Nothing

            myConnection.Close()

            myConnection = Nothing

        End If


    End Sub


    Private Sub NetRms_ClosedDNU_GTE10()

        'transformer on feeder

        Dim sql As String

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

        If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If


sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"


Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)
```

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)



        ClosedDNU_GTE10.Text = drcheck(dr, "closed_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub

Private Sub NetRms_ClosedDNU_GTE5LT10()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "'AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



ClosedDNU_GTE5LT10.Text = drcheck(dr, "closed_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub
```

```
Private Sub PlannedBanksOff_Feeder()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



      Dim Feeder_name As String

      Feeder_name = cmbSearchFeeder.SelectedValue



      If Feeder_name.Length < 5 Then

        Feeder_name = "0" + Feeder_name

      End If
```

```
sql = "SELECT count(*) AS pbo_count FROM [Transformer_OM_BO].[dbo].[PlannedBanksOff] where Feeder_ID='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

'PlannedBanksOff.Text = drcheck(dr, "pbo_count")


myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If
```

```
End Sub

Private Sub PlannedBanksOff_Feeder_GTE10()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If
```

```
sql = "SELECT count(*) AS pbo_count FROM [Transformer_OM_BO].[dbo].[PlannedBanksOff_GTE10] where Feeder_ID='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

'  PlannedBanksOff_GTE10.Text = drcheck(dr, "pbo_count")




myCommand = Nothing

myConnection.Close()
```

```
        myConnection = Nothing

      End If


    End Sub


    Private Sub PlannedBanksOff_Feeder_GTE5LT10()

      'transformer on feeder

      Dim sql As String

      '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

      If cmbSearchFeeder.Items.Count > 0 Then



        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue



        If Feeder_name.Length < 5 Then

          Feeder_name = "0" + Feeder_name
```

```
End If

sql = "SELECT count(*) AS pbo_count FROM [Transformer_OM_BO].[dbo].[PlannedBanksOff_GTE5LT10] where Feeder_ID='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

' PlannedBanksOff_GTE5LT10.Text = drcheck(dr, "pbo_count")
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

      End If



    End Sub



    Private Sub NetRms_Missing_Vault_Scan_Feeder()

      'transformer on feeder

      Dim sql As String

      '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

      If cmbSearchFeeder.Items.Count > 0 Then



        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue
```

```
If Feeder_name.Length < 5 Then
  Feeder_name = "0" + Feeder_name
End If

' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE (year([LVS_vault_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_2WEEK_VAULTSCAN] WHERE (year([LVS_NWK_SCAN_TIME_DATE])< 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then
```

```
' sql = "SELECT count(*) AS missing_count FROM
[Transformer_OM_BO].[dbo].[NETRMSM_2WEEK_VAULTSCAN] WHERE
(year([LVS_NWK_SCAN_TIME_DATE])< 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

' End If


Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

MissingFeederCurrent.Text = drcheck(dr, "missing_count")


myCommand = Nothing

myConnection.Close()

myConnection = Nothing
```

```
End If

End Sub

Private Sub NetRms_Missing_Vault_GTE5LT10()

'transformer on feeder

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name
```

End If

sql = "SELECT count(distinct([LVSD_VAULT_NUMBER])) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

'sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSBQ_2WEEK_VAULTSCAN] LVS INNER JOIN [Transformer_OM_BO].[dbo].[LVSD_STATIC_DATA_CURRENT_GTE5LT10] LVSD ON LVSD.[LVSD_FEEDER]=LVS.[LVS_FEEDER_NO] AND LVSD.[LVSD_VAULT_NUMBER]=LVS.[LVS_VAULT_NUMBER] AND (year([LVS_VAULT_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

"ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_2WEEK_VAULTSCAN] LVS INNER JOIN [Transformer_OM_BO].[dbo].[LVSD_STATIC_DATA_CURRENT_GTE5LT10] LVSD ON LVSD.[LVSD_FEEDER]=LVS.[LVS_FEEDER_NO] AND

```
LVSD.[LVSD_VAULT_NUMBER]=LVS.[LVS_VAULT_NUMBER] AND
(year([LVS_VAULT_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

                ' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

                ' sql = "SELECT count(*) AS missing_count FROM
[Transformer_OM_BO].[dbo].[NETRMSM_2WEEK_VAULTSCAN] LVS INNER JOIN
[Transformer_OM_BO].[dbo].[LVSD_STATIC_DATA_CURRENT_GTE5LT10] LVSD ON
LVSD.[LVSD_FEEDER]=LVS.[LVS_FEEDER_NO] AND
LVSD.[LVSD_VAULT_NUMBER]=LVS.[LVS_VAULT_NUMBER] AND
(year([LVS_VAULT_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

                ' End If

                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable
```

```
        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        Missing_GTE5LT10.Text = drcheck(dr, "missing_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub NetRms_Missing_Vault_GTE10()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found
```

```
If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    sql = "SELECT count(distinct([LVSD_VAULT_NUMBER])) AS missing_count
FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE10] WHERE LVSD_FEEDER =
'" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number  FROM
[Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)

    myCommand.SelectCommand.CommandTimeout = 180
```

```
        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        Missing_GTE10.Text = drcheck(dr, "missing_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub PlannedOpenMainsGT2_Feeder()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POM_NETRMALL_3MONTH_GT2] where LVSD_Feeder='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        'PlannedOMGT2.Text = drcheck(dr, "pom_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub

Private Sub PlannedOpenMainsLTE2_Feeder()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POM_NETRMALL_3MONTH_LTE2] where LVSD_Feeder='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        'PlannedOMLTE2.Text = drcheck(dr, "pom_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub PlannedOpenMainsLTE2_GTE10()

    'transformer on feeder
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue



If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If



sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "'"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
        Dim myCommand As New SqlDataAdapter(sql, myConnection)

        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        'PlannedOMLTE2_GTE10.Text = drcheck(dr, "pom_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub PlannedOpenMainsGT2_GTE10()
```

```
'transformer on feeder

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then



  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue



  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If



  sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMGT2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

' PlannedOMGT2_GTE10.Text = drcheck(dr, "pom_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub
```

```
Private Sub PlannedOpenMainsGT2_GTE5LT10()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



      Dim Feeder_name As String

      Feeder_name = cmbSearchFeeder.SelectedValue



      If Feeder_name.Length < 5 Then

        Feeder_name = "0" + Feeder_name

      End If


      sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "'"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)

                'NB: Grid views monitor Feeder to re-query!!!

                ' PlannedOMGT2_GTE5LT10.Text = drcheck(dr, "pom_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub PlannedOpenMainsLTE2_GTE5LT10()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "'"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)

                'NB: Grid views monitor Feeder to re-query!!!

                'PlannedOMLTE2_GTE5LT10.Text = drcheck(dr, "pom_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub NetRms_ClosedOld_Vault_Scan_Feeder()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_FEEDER_NO]='" & Feeder_name & "'"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)



                ClosedOld.Text = drcheck(dr, "closed_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub NetRms_ClosedUNR_Vault_Scan_Feeder()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_FEEDER_NO]='" & Feeder_name & "'"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)



                ClosedUNR.Text = drcheck(dr, "closed_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub NetRms_ClosedOld_Vault_Scan_GTE10()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER]
```

= '" & Feeder_name & "'AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



ClosedOld_GTE10.Text = drcheck(dr, "closed_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If
```

```
End Sub


Private Sub NetRms_ClosedOld_Vault_Scan_GTE5LT10()
  'transformer on feeder
  Dim sql As String
  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
  If cmbSearchFeeder.Items.Count > 0 Then


    Dim Feeder_name As String
    Feeder_name = cmbSearchFeeder.SelectedValue


    If Feeder_name.Length < 5 Then
      Feeder_name = "0" + Feeder_name
    End If
```

```
sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

ClosedOld_GTE5LT10.Text = drcheck(dr, "closed_count")

myCommand = Nothing

myConnection.Close()
```

```
            myConnection = Nothing

        End If



    End Sub

    Private Sub NetRms_ClosedUNR_Vault_Scan_GTE10()

        'transformer on feeder

        Dim sql As String

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

        If cmbSearchFeeder.Items.Count > 0 Then



            Dim Feeder_name As String

            Feeder_name = cmbSearchFeeder.SelectedValue



            If Feeder_name.Length < 5 Then

                Feeder_name = "0" + Feeder_name

            End If
```

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

ClosedUNR_GTE10.Text = drcheck(dr, "closed_count")

myCommand = Nothing

myConnection.Close()

```
        myConnection = Nothing

      End If


    End Sub


    Private Sub NetRms_ClosedUNR_Vault_Scan_GTE5LT10()

      'transformer on feeder

      Dim sql As String

      '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

      If cmbSearchFeeder.Items.Count > 0 Then


        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue


        If Feeder_name.Length < 5 Then

          Feeder_name = "0" + Feeder_name
```

```
End If

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER]='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

ClosedUNR_GTE5LT10.Text = drcheck(dr, "closed_count")
```

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If

End Sub

Private Sub TargetInitialValues_LPW()
    Target_OpenSwitchesFeeder.Text = OpenSwitchesFeederCurrent.Text
    Target_OpenSwitches_GTE5LT10.Text = OpenSwitches_GTE5LT10.Text
    Target_OpenSwitches_GTE10.Text = OpenSwitches_GTE10.Text
    Target_BanksOffFeederCurrent.Text = BanksOffFeederCurrent.Text
    Target_BanksOff_GTE10.Text = BanksOff_GTE10.Text
    Target_BanksOff_GTE5LT10.Text = BanksOff_GTE5LT10.Text
    Target_FuseFeederCurrent.Text = FuseFeederCurrent.Text
    Target_Fuse_GTE10.Text = Fuse_GTE10.Text
    Target_Fuse_GTE5LT10.Text = Fuse_GTE5LT10.Text
    Target_MissingFeederCurrent.Text = MissingFeederCurrent.Text
```

```
Target_Missing_GTE10.Text = Missing_GTE10.Text

Target_Missing_GTE5LT10.Text = Missing_GTE5LT10.Text

Target_ClosedOld.Text = ClosedOld.Text

Target_ClosedOld_GTE10.Text = ClosedOld_GTE10.Text

Target_ClosedOld_GTE5LT10.Text = ClosedOld_GTE5LT10.Text

Target_ClosedUNR.Text = ClosedUNR.Text

Target_ClosedUNR_GTE10.Text = ClosedUNR_GTE10.Text

Target_ClosedUNR_GTE5LT10.Text = ClosedUNR_GTE5LT10.Text

Target_ClosedDNUCurrent.Text = ClosedDNUCurrent.Text

Target_ClosedDNU_GTE10.Text = ClosedDNU_GTE10.Text

Target_ClosedDNU_GTE5LT10.Text = ClosedDNU_GTE5LT10.Text

Target_OpenMainsLTE2Current.Text = OpenMainsLTE2Current.Text

Target_OMLTE2_GTE10.Text = OMLTE2_GTE10.Text

Target_OMLTE2_GTE5LT10.Text = OMLTE2_GTE5LT10.Text

Target_OpenMainsGT2Current.Text = OpenMainsGT2Current.Text

Target_OMGT2_GTE10.Text = OMGT2_GTE10.Text

Target_OMGT2_GTE5LT10.Text = OMGT2_GTE5LT10.Text

' Target_PlannedOMLTE2.Text = PlannedOMLTE2.Text
```

```
' Target_PlannedOMLTE2_GTE10.Text = PlannedOMLTE2_GTE10.Text

' Target_PlannedOMLTE2_GTE5LT10.Text = PlannedOMLTE2_GTE5LT10.Text

' Target_PlannedOMGT2.Text = PlannedOMGT2.Text

' Target_PlannedOMGT2_GTE10.Text = PlannedOMGT2_GTE10.Text

' Target_PlannedOMGT2_GTE5LT10.Text = PlannedOMGT2_GTE5LT10.Text

' Target_PlannedBanksOff.Text = PlannedBanksOff.Text

' Target_PlannedBanksOff_GTE10.Text = PlannedBanksOff_GTE10.Text

' Target_PlannedBanksOff_GTE5LT10.Text = PlannedBanksOff_GTE5LT10.Text


End Sub


Private Sub Total_LPW_Calc()


Sum_SwitchesOpen.Text = 5 * Int32.Parse(OpenSwitchesFeederCurrent.Text) + 5 *
Int32.Parse(OpenSwitches_GTE10.Text) + 3 * Int32.Parse(OpenSwitches_GTE5LT10.Text)


Sum_BanksOff.Text = 5 * Int32.Parse(BanksOffFeederCurrent.Text) + 5 *
Int32.Parse(BanksOff_GTE10.Text) + 3 * Int32.Parse(BanksOff_GTE5LT10.Text)
```

Sum_Fuse.Text = 5 * Int32.Parse(FuseFeederCurrent.Text) + 5 * Int32.Parse(Fuse_GTE10.Text) + 3 * Int32.Parse(Fuse_GTE5LT10.Text)

Sum_Missing.Text = Int32.Parse(MissingFeederCurrent.Text) + Int32.Parse(Missing_GTE10.Text) + Int32.Parse(Missing_GTE5LT10.Text)

Sum_ClosedOld.Text = Int32.Parse(ClosedOld.Text) + Int32.Parse(ClosedOld_GTE10.Text) + Int32.Parse(ClosedOld_GTE5LT10.Text)

Sum_ClosedUNR.Text = Int32.Parse(ClosedUNR.Text) + Int32.Parse(ClosedUNR_GTE10.Text) + Int32.Parse(ClosedUNR_GTE5LT10.Text)

Sum_ClosedDNU.Text = Int32.Parse(ClosedDNUCurrent.Text) + Int32.Parse(ClosedDNU_GTE10.Text) + Int32.Parse(ClosedDNU_GTE5LT10.Text)

Sum_OpenMainsLTE2.Text = Int32.Parse(OpenMainsLTE2Current.Text) + Int32.Parse(OMLTE2_GTE10.Text) + Int32.Parse(OMLTE2_GTE5LT10.Text)

Sum_OpenMainsGT2.Text = 2 * (Int32.Parse(OpenMainsGT2Current.Text) + Int32.Parse(OMGT2_GTE10.Text) + Int32.Parse(OMGT2_GTE5LT10.Text))

' Sum_PlannedOpenMainsLTE2.Text = Int32.Parse(PlannedOMLTE2.Text) + Int32.Parse(PlannedOMLTE2_GTE10.Text) + Int32.Parse(PlannedOMLTE2_GTE5LT10.Text)

'Sum_PlannedOpenMainsGT2.Text = 2 * (Int32.Parse(PlannedOMGT2.Text) + Int32.Parse(PlannedOMGT2_GTE10.Text) + Int32.Parse(PlannedOMGT2_GTE5LT10.Text))

' Sum_PlannedBanksOff.Text = 3 * (Int32.Parse(PlannedBanksOff.Text) + Int32.Parse(PlannedBanksOff_GTE10.Text) + Int32.Parse(PlannedBanksOff_GTE5LT10.Text))

TotalLPWCurrent.Text = Int32.Parse(Sum_SwitchesOpen.Text) + Int32.Parse(Sum_BanksOff.Text) + Int32.Parse(Sum_Fuse.Text) + Int32.Parse(Sum_Missing.Text) + Int32.Parse(Sum_ClosedOld.Text) + Int32.Parse(Sum_ClosedUNR.Text) + Int32.Parse(Sum_ClosedDNU.Text) + Int32.Parse(Sum_OpenMainsLTE2.Text) + Int32.Parse(Sum_OpenMainsGT2.Text)

' + Int32.Parse(Sum_PlannedOpenMainsLTE2.Text) + Int32.Parse(Sum_PlannedOpenMainsGT2.Text) + Int32.Parse(Sum_PlannedBanksOff.Text)

Transformer_Current.Text = 5 * (Int32.Parse(OpenSwitchesFeederCurrent.Text) + Int32.Parse(BanksOffFeederCurrent.Text) + Int32.Parse(FuseFeederCurrent.Text)) + Int32.Parse(MissingFeederCurrent.Text) + Int32.Parse(ClosedOld.Text) + Int32.Parse(ClosedUNR.Text) + Int32.Parse(ClosedDNUCurrent.Text) + Int32.Parse(OpenMainsLTE2Current.Text) + 2 * Int32.Parse(OpenMainsGT2Current.Text)

'+ Int32.Parse(PlannedOMLTE2.Text) + 2 * Int32.Parse(PlannedOMGT2.Text) + 3 * Int32.Parse(PlannedBanksOff.Text)

```
                Transformer_NBY_GTE10.Text = 5 * (Int32.Parse(OpenSwitches_GTE10.Text) +
Int32.Parse(BanksOff_GTE10.Text) + Int32.Parse(Fuse_GTE10.Text)) + Int32.Parse(Missing_GTE10.Text) +
Int32.Parse(ClosedOld_GTE10.Text) + Int32.Parse(ClosedUNR_GTE10.Text) +
Int32.Parse(ClosedDNU_GTE10.Text) + Int32.Parse(OMLTE2_GTE10.Text) + 2 *
Int32.Parse(OMGT2_GTE10.Text)

                ' + Int32.Parse(PlannedOMLTE2_GTE10.Text) + 2 *
Int32.Parse(PlannedOMGT2_GTE10.Text) + 3 * Int32.Parse(PlannedBanksOff_GTE10.Text)

                Transformer_NBY_GTE5LT10.Text = 3 * (Int32.Parse(OpenSwitches_GTE5LT10.Text)
+ Int32.Parse(BanksOff_GTE5LT10.Text) + Int32.Parse(Fuse_GTE5LT10.Text)) +
Int32.Parse(Missing_GTE5LT10.Text) + Int32.Parse(ClosedOld_GTE5LT10.Text) +
Int32.Parse(ClosedUNR_GTE5LT10.Text) + Int32.Parse(ClosedDNU_GTE5LT10.Text) +
Int32.Parse(OMLTE2_GTE5LT10.Text) + 2 * Int32.Parse(OMGT2_GTE5LT10.Text)

                ' + Int32.Parse(PlannedOMLTE2_GTE5LT10.Text) + 2 *
Int32.Parse(PlannedOMGT2_GTE5LT10.Text) + 3 * Int32.Parse(PlannedBanksOff_GTE5LT10.Text)

            End Sub
```

```
Private Sub Target_Total_LPW_Calc()

Target_Sum_SwitchesOpen.Text = 5 * Int32.Parse(Target_OpenSwitchesFeeder.Text) + 5 * Int32.Parse(Target_OpenSwitches_GTE10.Text) + 3 * Int32.Parse(Target_OpenSwitches_GTE5LT10.Text)

Target_Sum_BanksOff.Text = 5 * Int32.Parse(Target_BanksOffFeederCurrent.Text) + 5 * Int32.Parse(Target_BanksOff_GTE10.Text) + 3 * Int32.Parse(Target_BanksOff_GTE5LT10.Text)

Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 * Int32.Parse(Target_Fuse_GTE10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)

Target_Sum_Missing.Text = Int32.Parse(Target_MissingFeederCurrent.Text) + Int32.Parse(Target_Missing_GTE10.Text) + Int32.Parse(Target_Missing_GTE5LT10.Text)

Target_Sum_ClosedOld.Text = Int32.Parse(Target_ClosedOld.Text) + Int32.Parse(Target_ClosedOld_GTE10.Text) + Int32.Parse(Target_ClosedOld_GTE5LT10.Text)

Target_Sum_ClosedUNR.Text = Int32.Parse(Target_ClosedUNR.Text) + Int32.Parse(Target_ClosedUNR_GTE10.Text) + Int32.Parse(Target_ClosedUNR_GTE5LT10.Text)
```

Target_Sum_ClosedDNU.Text = Int32.Parse(Target_ClosedDNUCurrent.Text) + Int32.Parse(Target_ClosedDNU_GTE10.Text) + Int32.Parse(Target_ClosedDNU_GTE5LT10.Text)

Target_Sum_OpenMainsLTE2.Text = Int32.Parse(Target_OpenMainsLTE2Current.Text) + Int32.Parse(Target_OMLTE2_GTE10.Text) + Int32.Parse(Target_OMLTE2_GTE5LT10.Text)

Target_Sum_OpenMainsGT2.Text = 2 * (Int32.Parse(Target_OpenMainsGT2Current.Text) + Int32.Parse(Target_OMGT2_GTE10.Text) + Int32.Parse(Target_OMGT2_GTE5LT10.Text))

'Target_Sum_PlannedOpenMainsLTE2.Text = Int32.Parse(Target_PlannedOMLTE2.Text) + Int32.Parse(Target_PlannedOMLTE2_GTE10.Text) + Int32.Parse(Target_PlannedOMLTE2_GTE5LT10.Text)

' Target_Sum_PlannedOpenMainsGT2.Text = 2 * (Int32.Parse(Target_PlannedOMGT2.Text) + Int32.Parse(Target_PlannedOMGT2_GTE10.Text) + Int32.Parse(Target_PlannedOMGT2_GTE5LT10.Text))

' Target_Sum_PlannedBanksOff.Text = 3 * (Int32.Parse(Target_PlannedBanksOff.Text) + Int32.Parse(Target_PlannedBanksOff_GTE10.Text) + Int32.Parse(Target_PlannedBanksOff_GTE5LT10.Text))

Target_TotalLPWCurrent.Text = Int32.Parse(Target_Sum_SwitchesOpen.Text) + Int32.Parse(Target_Sum_BanksOff.Text) + Int32.Parse(Target_Sum_Fuse.Text) + Int32.Parse(Target_Sum_Missing.Text) + Int32.Parse(Target_Sum_ClosedOld.Text) + Int32.Parse(Target_Sum_ClosedUNR.Text) + Int32.Parse(Target_Sum_ClosedDNU.Text) + Int32.Parse(Target_Sum_OpenMainsLTE2.Text) + Int32.Parse(Target_Sum_OpenMainsGT2.Text)

'+ Int32.Parse(Target_Sum_PlannedOpenMainsLTE2.Text) +
Int32.Parse(Target_Sum_PlannedOpenMainsGT2.Text) + Int32.Parse(Target_Sum_PlannedBanksOff.Text)

Target_Transformer_Current.Text = 5 * (Int32.Parse(Target_OpenSwitchesFeeder.Text) +
Int32.Parse(Target_BanksOffFeederCurrent.Text) + Int32.Parse(Target_FuseFeederCurrent.Text)) +
Int32.Parse(Target_MissingFeederCurrent.Text) + Int32.Parse(Target_ClosedOld.Text) +
Int32.Parse(Target_ClosedUNR.Text) + Int32.Parse(Target_ClosedDNUCurrent.Text) +
Int32.Parse(Target_OpenMainsLTE2Current.Text) + 2 * Int32.Parse(Target_OpenMainsGT2Current.Text)

' + Int32.Parse(Target_PlannedOMLTE2.Text) + 2 *
Int32.Parse(Target_PlannedOMGT2.Text) + 3 * Int32.Parse(Target_PlannedBanksOff.Text)

Target_Transformer_NBY_GTE10.Text = 5 *
(Int32.Parse(Target_OpenSwitches_GTE10.Text) + Int32.Parse(Target_BanksOff_GTE10.Text) +
Int32.Parse(Target_Fuse_GTE10.Text)) + Int32.Parse(Target_Missing_GTE10.Text) +
Int32.Parse(Target_ClosedOld_GTE10.Text) + Int32.Parse(Target_ClosedUNR_GTE10.Text) +
Int32.Parse(Target_ClosedDNU_GTE10.Text) + Int32.Parse(Target_OMLTE2_GTE10.Text) + 2 *
Int32.Parse(Target_OMGT2_GTE10.Text)

' + Int32.Parse(Target_PlannedOMLTE2_GTE10.Text) + 2 *
Int32.Parse(Target_PlannedOMGT2_GTE10.Text) + 3 * Int32.Parse(Target_PlannedBanksOff_GTE10.Text)

target_Transformer_NBY_GTE5LT10.Text = 3 * (Int32.Parse(Target_OpenSwitches_GTE5LT10.Text) + Int32.Parse(Target_BanksOff_GTE5LT10.Text) + Int32.Parse(Target_Fuse_GTE5LT10.Text)) + Int32.Parse(Target_Missing_GTE5LT10.Text) + Int32.Parse(Target_ClosedOld_GTE5LT10.Text) + Int32.Parse(Target_ClosedUNR_GTE5LT10.Text) + Int32.Parse(Target_ClosedDNU_GTE5LT10.Text) + Int32.Parse(Target_OMLTE2_GTE5LT10.Text) + 2 * Int32.Parse(Target_OMGT2_GTE5LT10.Text)

' + Int32.Parse(Target_PlannedOMLTE2_GTE5LT10.Text) + 2 * Int32.Parse(Target_PlannedOMGT2_GTE5LT10.Text) + 3 * Int32.Parse(Target_PlannedBanksOff_GTE5LT10.Text)

End Sub

Private Sub LPW_Cost_Calculation()

closeOpenSwitches.Text = Int32.Parse(OpenSwitchesFeederCurrent.Text) - Int32.Parse(Target_OpenSwitchesFeeder.Text) + Int32.Parse(OpenSwitches_GTE10.Text) - Int32.Parse(Target_OpenSwitches_GTE10.Text) + Int32.Parse(OpenSwitches_GTE5LT10.Text) - Int32.Parse(Target_OpenSwitches_GTE5LT10.Text)

closeOpenSwitches_TotalCost.Text = Int32.Parse(unitCost_CloseOpenSwitches.Text) * Int32.Parse(closeOpenSwitches.Text)

closeBanksOff.Text = Int32.Parse(BanksOffFeederCurrent.Text) - Int32.Parse(Target_BanksOffFeederCurrent.Text) + Int32.Parse(BanksOff_GTE10.Text) -

Int32.Parse(Target_BanksOff_GTE10.Text) + Int32.Parse(BanksOff_GTE5LT10.Text) - Int32.Parse(Target_BanksOff_GTE5LT10.Text)

closeBanksOff_TotalCost.Text = Int32.Parse(unitCost_CloseBanksOff.Text) * Int32.Parse(closeBanksOff.Text)

repairOpenFuse.Text = Int32.Parse(FuseFeederCurrent.Text) - Int32.Parse(Target_FuseFeederCurrent.Text) + Int32.Parse(Fuse_GTE10.Text) - Int32.Parse(Target_Fuse_GTE10.Text) + Int32.Parse(Fuse_GTE5LT10.Text) - Int32.Parse(Target_Fuse_GTE5LT10.Text)

repairOpenFuse_TotalCost.Text = Int32.Parse(unitCost_RepairOpenFuse.Text) * Int32.Parse(repairOpenFuse.Text)

repairOpenMains.Text = Int32.Parse(OpenMainsLTE2Current.Text) - Int32.Parse(Target_OpenMainsLTE2Current.Text) + Int32.Parse(OMLTE2_GTE10.Text) - Int32.Parse(Target_OMLTE2_GTE10.Text) + Int32.Parse(OMLTE2_GTE5LT10.Text) - Int32.Parse(Target_OMLTE2_GTE5LT10.Text) + Int32.Parse(OpenMainsGT2Current.Text) - Int32.Parse(Target_OpenMainsGT2Current.Text) + Int32.Parse(OMGT2_GTE10.Text) - Int32.Parse(Target_OMGT2_GTE10.Text) + Int32.Parse(OMGT2_GTE5LT10.Text) - Int32.Parse(Target_OMGT2_GTE5LT10.Text)

RepairOpenMains_TotalCost.Text = Int32.Parse(repairOpenMains.Text) * Int32.Parse(unitCost_RepairOpenMains.Text)

```
FixReporting.Text = Int32.Parse(ClosedDNUCurrent.Text) -
Int32.Parse(Target_ClosedDNUCurrent.Text) + Int32.Parse(ClosedUNR.Text) -
Int32.Parse(Target_ClosedUNR.Text) + Int32.Parse(ClosedOld.Text) - Int32.Parse(Target_ClosedOld.Text) +
Int32.Parse(MissingFeederCurrent.Text) - Int32.Parse(Target_MissingFeederCurrent.Text) +
Int32.Parse(ClosedDNU_GTE10.Text) - Int32.Parse(Target_ClosedDNU_GTE10.Text) +
Int32.Parse(ClosedUNR_GTE10.Text) - Int32.Parse(Target_ClosedUNR_GTE10.Text) +
Int32.Parse(ClosedOld_GTE10.Text) - Int32.Parse(Target_ClosedOld_GTE10.Text) +
Int32.Parse(Missing_GTE10.Text) - Int32.Parse(Target_Missing_GTE10.Text) +
Int32.Parse(ClosedDNU_GTE5LT10.Text) - Int32.Parse(Target_ClosedDNU_GTE5LT10.Text) +
Int32.Parse(ClosedUNR_GTE5LT10.Text) - Int32.Parse(Target_ClosedUNR_GTE5LT10.Text) +
Int32.Parse(ClosedOld_GTE5LT10.Text) - Int32.Parse(Target_ClosedOld_GTE5LT10.Text) +
Int32.Parse(Missing_GTE5LT10.Text) - Int32.Parse(Target_Missing_GTE5LT10.Text)

FixReporting_TotalCost.Text = Int32.Parse(FixReporting.Text) *
Int32.Parse(unitCost_FixReporting.Text)

totalNumberActions.Text = Int32.Parse(closeOpenSwitches.Text) +
Int32.Parse(closeBanksOff.Text) + Int32.Parse(repairOpenFuse.Text) + Int32.Parse(repairOpenMains.Text) +
Int32.Parse(FixReporting.Text)

upgrade_TotalCost.Text = Int32.Parse(closeOpenSwitches_TotalCost.Text) +
Int32.Parse(closeBanksOff_TotalCost.Text) + Int32.Parse(repairOpenFuse_TotalCost.Text) +
Int32.Parse(RepairOpenMains_TotalCost.Text) + Int32.Parse(FixReporting_TotalCost.Text)

End Sub
```

```
Private Function numericCheck(ByVal item As String) As String

    Dim num As Integer

    Dim result As Boolean = Int32.TryParse(item, num)

    If result Then

        If num < 0 Then Return "0" Else Return num

    Else : Return "0"

    End If


End Function
```

```
Private Sub Check_Target_Values()

    Target_OpenSwitchesFeeder.Text = numericCheck(Target_OpenSwitchesFeeder.Text)

    Target_OpenSwitches_GTE10.Text = numericCheck(Target_OpenSwitches_GTE10.Text)

    Target_OpenSwitches_GTE5LT10.Text = numericCheck(Target_OpenSwitches_GTE5LT10.Text)

    Target_BanksOffFeederCurrent.Text = numericCheck(Target_BanksOffFeederCurrent.Text)

    Target_BanksOff_GTE10.Text = numericCheck(Target_BanksOff_GTE10.Text)

    Target_BanksOff_GTE5LT10.Text = numericCheck(Target_BanksOff_GTE5LT10.Text)

    Target_FuseFeederCurrent.Text = numericCheck(Target_FuseFeederCurrent.Text)

    Target_Fuse_GTE5LT10.Text = numericCheck(Target_Fuse_GTE5LT10.Text)

    Target_Fuse_GTE10.Text = numericCheck(Target_Fuse_GTE10.Text)

    Target_MissingFeederCurrent.Text = numericCheck(Target_MissingFeederCurrent.Text)
```

```
Target_Missing_GTE5LT10.Text = numericCheck(Target_Missing_GTE5LT10.Text)

Target_Missing_GTE10.Text = numericCheck(Target_Missing_GTE10.Text)

Target_OpenMainsGT2Current.Text =
numericCheck(Target_OpenMainsGT2Current.Text)

Target_OMGT2_GTE5LT10.Text = numericCheck(Target_OMGT2_GTE5LT10.Text)

Target_OMGT2_GTE10.Text = numericCheck(Target_OMGT2_GTE10.Text)

Target_OpenMainsLTE2Current.Text =
numericCheck(Target_OpenMainsLTE2Current.Text)

Target_OMLTE2_GTE5LT10.Text = numericCheck(Target_OMLTE2_GTE5LT10.Text)

Target_OMLTE2_GTE10.Text = numericCheck(Target_OMLTE2_GTE10.Text)

Target_ClosedOld.Text = numericCheck(Target_ClosedOld.Text)

Target_ClosedOld_GTE5LT10.Text =
numericCheck(Target_ClosedOld_GTE5LT10.Text)

Target_ClosedOld_GTE10.Text = numericCheck(Target_ClosedOld_GTE10.Text)

Target_ClosedDNUCurrent.Text = numericCheck(Target_ClosedDNUCurrent.Text)

Target_ClosedDNU_GTE5LT10.Text =
numericCheck(Target_ClosedDNU_GTE5LT10.Text)

Target_ClosedDNU_GTE10.Text = numericCheck(Target_ClosedDNU_GTE10.Text)

Target_ClosedUNR.Text = numericCheck(Target_ClosedUNR.Text)
```

```
Target_ClosedUNR_GTE5LT10.Text =
numericCheck(Target_ClosedUNR_GTE5LT10.Text)

Target_ClosedUNR_GTE10.Text = numericCheck(Target_ClosedUNR_GTE10.Text)

End Sub

Protected Sub LPW_Reset_Click(ByVal sender As Object, ByVal e As System.EventArgs)

'Protected Sub LPW_Submit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LPW_Submit.Click

'Check_Target_Values()
```

```
'Target_Total_LPW_Calc()

'LPW_Cost_Calculation()

'' LPW_grid()

'GridviewdataBindBanksOff()

'GridViewdataBindOpenSwitches()

'GridViewdataBindFuse()

'GridViewDataBindClosedOld()

'GridViewDataBindClosedUNR()

'GridViewDataBindClosedDNU()

'GridViewDataBindOpenMainsLTE2()

'GridViewDataBindOpenMainsGT2()

' BindAttributes1()

reset_visitedLPWSessionVariables()

lpwreset()



End Sub

Private Sub GridviewdataBindBanksOff()
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT Vault_id, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH] where Feeder_ID = '" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "BO_NETRMSALL_3MONTH")



GV_BanksOff.DataSource = ds

GV_BanksOff.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewdataBindOpenSwitches()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name
```

```
End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_vault_number,LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE ([Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%B%' OR [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%A%' ) AND LVS_FEEDER_NO = '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet
```

```
myCommand.Fill(ds, "NetRMS_OpenSwitches")



GV_OpenSwitches.DataSource = ds

GV_OpenSwitches.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If









End Sub

Private Sub GridViewdataBindFuse()



  Dim sql As String
```

```
'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE LVS_VAULT_CONDITION like '%D%' AND LVS_FEEDER_NO='" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_Fuse")



GV_Fuse.DataSource = ds

GV_Fuse.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewDataBindClosedOld()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_FEEDER_NO]= '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT")

```
        GV_ClosedOld.DataSource = ds

        GV_ClosedOld.DataBind()




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

        '  End If




End Sub


Private Sub GridViewDataBindClosedUNR()


    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT] where lvs_feeder_no = '" & Feeder_name & "'"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT")




GV_ClosedUNR.DataSource = ds

GV_ClosedUNR.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewDataBindClosedDNU()


  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then


  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT] where lvs_Feeder_no = '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT")

GV_ClosedDNU.DataSource = ds

GV_ClosedDNU.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If



End Sub

Private Sub GridViewDataBindOpenMainsLTE2()



  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in(select nearest_vault_id from [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180
```

```
        Dim ds As DataSet = New DataSet

        myCommand.Fill(ds, "OpenMainsLTE2")

        GV_OpenMainsLTE2.DataSource = ds



        GV_OpenMainsLTE2.DataBind()




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

        ' End If


End Sub


Private Sub GridViewDataBindOpenMainsGT2()
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH] where LVSD_FEEDER='" & Feeder_name & "'
```

```
and nearest_vault_id in (select nearest_vault_id from [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_gt2] where LVSD_FEEDER='" & Feeder_name & "')"

                Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim ds As DataSet = New DataSet

                myCommand.Fill(ds, "OpenMainsGT2")

                GV_OpenMainsGT2.DataSource = ds



                GV_OpenMainsGT2.DataBind()




                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

                ' End If
```

```
End Sub

Private Sub GridviewdataBindMissing()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name
```

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

' sql = "SELECT Vault_id, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH] where Feeder_ID = '" & Feeder_name & "'"

sql = "SELECT lvs_vault_number, lvs_vault_scan_time_date FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE (year([LVS_vault_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "Missing")

```
GV_Missing.DataSource = ds

GV_Missing.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub



Private Sub GridviewdataBindBanksOff_NbyFactorGTE10()



  Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT LVSD_NBY_Vault_Number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "BO_NETRMSALL_3MONTHGTE10")




GV_BanksOff_NbyFactorGTE10.DataSource = ds

GV_BanksOff_NbyFactorGTE10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewdataBindOpenSwitches_NbyFactorGTE10()

    Dim sql As String
    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
    '  If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String
    Feeder_name = cmbSearchFeeder.SelectedValue
    If Feeder_name.Length < 5 Then
      Feeder_name = "0" + Feeder_name
    End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE10] WHERE LVSD_FEEDER= '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_OpenSwitches_GTE10")

GV_OpenSwitches_NbyFactorGTE10.DataSource = ds

```
GV_OpenSwitches_NbyFactorGTE10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub

Private Sub GridViewdataBindFuse_NbyFactorGTE10()



Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE10] WHERE LVSD_FEEDER ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_FuseGTE10")



GV_Fuse_NbyFactorGTE10.DataSource = ds

GV_Fuse_NbyFactorGTE10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If



End Sub
```

```
Private Sub GridViewDataBindClosedOld_NbyFactorGTE10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    ' If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM
[Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER]
= '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM
[Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds,
"CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")

GV_ClosedOld_NbyFactorGTE10.DataSource = ds

GV_ClosedOld_NbyFactorGTE10.DataBind()
```

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If

End Sub

Private Sub GridViewDataBindClosedUNR_NbyFactorGTE10()

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER] ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number  FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim ds As DataSet = New DataSet

                myCommand.Fill(ds,
"CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")




                GV_ClosedUNR_NbyFactorGTE10.DataSource = ds

                GV_ClosedUNR_NbyFactorGTE10.DataBind()




                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

                ' End If
```

```
End Sub

Private Sub GridViewDataBindClosedDNU_NbyFactorGTE10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

    '  If cmbSearchFeeder.Items.Count > 0 Then


    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")

GV_ClosedDNU_NbyFactorGTE10.DataSource = ds

GV_ClosedDNU_NbyFactorGTE10.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If

End Sub

Private Sub GridViewDataBindMissing_NbyFactorGTE10()

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number  FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)
```

```
myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "Missing_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")

GV_Missing_NbyFactorGTE10.DataSource = ds

GV_Missing_NbyFactorGTE10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub
```

```
Private Sub GridViewDataBindOpenMainsLTE2_NbyFactorGTE10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    ' If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "OpenMainsLTE2GTE10")

GV_OpenMainsLTE2_NbyFactorGTE10.DataSource = ds

GV_OpenMainsLTE2_NbyFactorGTE10.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If



End Sub


Private Sub GridViewDataBindOpenMainsGT2_NbyFactorGTE10()


Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If
```

```
'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "OpenMainsGT2GTE10")

GV_OpenMainsGT2_NbyFactorGTE10.DataSource = ds



GV_OpenMainsGT2_NbyFactorGTE10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridviewdataBindBanksOff_NbyFactorGTE5LT10()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  '  If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT LVSD_NBY_Vault_Number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "BO_NETRMSALL_3MONTHGTE5LT10")

GV_BanksOff_NbyFactorGTE5LT10.DataSource = ds

GV_BanksOff_NbyFactorGTE5LT10.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If



End Sub



Private Sub GridViewdataBindOpenSwitches_NbyFactorGTE5LT10()



Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If
```

```
'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER= '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_OpenSwitchesGTE5LT10")



GV_OpenSwitches_NbyFactorGTE5LT10.DataSource = ds

GV_OpenSwitches_NbyFactorGTE5LT10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewdataBindFuse_NbyFactorGTE5LT10()


  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  '  If cmbSearchFeeder.Items.Count > 0 Then


  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If


  'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_FuseGTE5LT10")


GV_Fuse_NbyFactorGTE5LT10.DataSource = ds

GV_Fuse_NbyFactorGTE5LT10.DataBind()


myCommand = Nothing

myConnection.Close()
```

```
myConnection = Nothing

' End If

End Sub

Private Sub GridViewDataBindClosedOld_NbyFactorGTE5LT10()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String
```

```
Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If
```

```
'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)
```

```
                myCommand.SelectCommand.CommandTimeout = 180

                Dim ds As DataSet = New DataSet

                myCommand.Fill(ds,
"CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")




                GV_ClosedOld_NbyFactorGTE5LT10.DataSource = ds

                GV_ClosedOld_NbyFactorGTE5LT10.DataBind()




                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

                ' End If
```

```
End Sub


Private Sub GridViewDataBindClosedUNR_NbyFactorGTE5LT10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    '  If cmbSearchFeeder.Items.Count > 0 Then


    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER] ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")

```
GV_ClosedUNR_NbyFactorGTE5LT10.DataSource = ds

GV_ClosedUNR_NbyFactorGTE5LT10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub

Private Sub GridViewDataBindClosedDNU_NbyFactorGTE5LT10()



  Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no
```

overloaded feeder for selected criteria found

```
' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number,LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")

GV_ClosedDNU_NbyFactorGTE5LT10.DataSource = ds

GV_ClosedDNU_NbyFactorGTE5LT10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewDataBindMissing_NbyFactorGTE5LT10()

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    '  If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name
```

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE5LT10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "MISSING_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")

```
        GV_Missing_NbyFactorGTE5LT10.DataSource = ds

        GV_Missing_NbyFactorGTE5LT10.DataBind()




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

        ' End If




    End Sub

    Private Sub GridViewDataBindOpenMainsLTE2_NbyFactorGTE5LT10()



        Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "OpenMainsLTE2GTE5LT10")

GV_OpenMainsLTE2_NbyFactorGTE5LT10.DataSource = ds

GV_OpenMainsLTE2_NbyFactorGTE5LT10.DataBind()

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

```
' End If

End Sub

Private Sub GridViewDataBindOpenMainsGT2_NbyFactorGTE5LT10()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then
```

```
Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300
```

```
        Dim ds As DataSet = New DataSet

        myCommand.Fill(ds, "OpenMainsGT2GTE5LT10")

        GV_OpenMainsGT2_NbyFactorGTE5LT10.DataSource = ds



        GV_OpenMainsGT2_NbyFactorGTE5LT10.DataBind()




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

        ' End If


    End Sub
```

```
Protected Sub RelPercent_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles RelPercent.SelectedIndexChanged

    RelXLPCalculationReset()

End Sub



Private Sub BindAttributes_Precomp()

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then

        Dim fdrvalue As String

        fdrvalue = cmbSearchFeeder.SelectedValue

        'If cmbSearchFeeder.SelectedValue.Length < 5 Then

        'fdrvalue = "0" + cmbSearchFeeder.SelectedValue

        'End If
```

```
Dim sql As String = "SELECT * FROM Rerank_Attributes_PrecomMap_2008 WHERE
Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

Feeder.Text = cmbSearchFeeder.SelectedValue

Feeder1.Text = cmbSearchFeeder.SelectedValue

Feeder2.Text = cmbSearchFeeder.SelectedValue
```

```
Feeder4.Text = cmbSearchFeeder.SelectedValue

' Feeder5.Text = cmbSearchFeeder.SelectedValue

Feeder6.Text = cmbSearchFeeder.SelectedValue

Feeder7.Text = cmbSearchFeeder.SelectedValue

feeder11.Text = cmbSearchFeeder.SelectedValue

feeder12.Text = cmbSearchFeeder.SelectedValue

feeder20.Text = cmbSearchFeeder.SelectedValue

Feeder_Name.Text = cmbSearchFeeder.SelectedValue

SelectedFeederName.Text = cmbSearchFeeder.SelectedValue




FLoad.Text = drcheck(dr, "mon_Load")

Normal_Rating.Text = drcheck(dr, "mon_Normal_Rating")

Emergency_Rating.Text = drcheck(dr, "mon_Emergency_Rating")

Shift.Text = drcheck(dr, "mon_Shift")

Delta.Text = drcheck(dr, "mon_Delta")

Emergency_Load.Text = drcheck(dr, "mon_Emergency_Load")

Feeders_4kv.Text = drcheck(dr, "mon_4kv")
```

```
Paper_GT40.Text = drcheck(dr, "mon_Paper_GT40")

Paper_LE40.Text = drcheck(dr, "mon_Paper_LE40")

'changed by shringika to exclude mon_solid_LT16 and mon_solid_GT16

Solid_LT16.Text = drcheck(dr, "mon_Solid_LT16")

Solid_70_75.Text = drcheck(dr, "mon_Solid_70_75")

Solid_GT16.Text = drcheck(dr, "mon_Solid_GT16")

Percent_Paper_GT40.Text = drcheck(dr, "mon_percent_Paper_GT40")

Percent_Paper_LE40.Text = drcheck(dr, "mon_percent_Paper_LE40")

Percent_Solid_LT16.Text = drcheck(dr, "mon_percent_Solid_LT16")

Percent_Solid_70_75.Text = drcheck(dr, "mon_percent_Solid_70_75")

Percent_Solid_GT16.Text = drcheck(dr, "mon_percent_Solid_GT16")

Sec_CNT.Text = drcheck(dr, "mon_Sec_CNT")

E21GT93.Text = drcheck(dr, "mon_E21GT93")

E21LE93.Text = drcheck(dr, "mon_E21LE93")

R31GT94.Text = drcheck(dr, "mon_R31GT94")

R31LE93.Text = drcheck(dr, "mon_R31LE93")

OtherStop_LT7.Text = drcheck(dr, "mon_OtherStop_LT7")

OtherStop7_12.Text = drcheck(dr, "mon_OtherStop7_12")
```

```
OtherStop_GT12.Text = drcheck(dr, "mon_OtherStop_GT12")

AllStop.Text = drcheck(dr, "mon_AllStop")

Percent_ALLStop.Text = drcheck(dr, "mon_percent_AllStop")

PJ_LT8.Text = drcheck(dr, "mon_PJ_LT8")

PJ8_16.Text = drcheck(dr, "mon_PJ8_16")

PJ16_24.Text = drcheck(dr, "mon_PJ16_24")

' PJ_GT24.Text = drcheck(dr, "mon_PJ_GT24")

PJ24_32.Text = drcheck(dr, "mon_PJ24-32")

PJ32_40.Text = drcheck(dr, "mon_PJ32-40")

PJGT40.Text = drcheck(dr, "mon_PJGT40")

AllPaper.Text = drcheck(dr, "mon_AllPaper")

Percent_AllPaper.Text = drcheck(dr, "mon_Percent_AllPaper")

SJ_LT8.Text = drcheck(dr, "mon_SJ_LT8")

SJ8_16.Text = drcheck(dr, "mon_SJ8_16")

SJ16_24.Text = drcheck(dr, "mon_SJ16_24")

SJ_GT24.Text = drcheck(dr, "mon_SJ_GT24")

AllSolid.Text = drcheck(dr, "mon_AllSolid")

Percent_AllSolid.Text = drcheck(dr, "mon_percent_AllSolid")
```

Percent_E21GT93.Text = drcheck(dr, "mon_percent_E21GT93")

Percent_E21LE93.Text = drcheck(dr, "mon_percent_E21LE93")

Percent_R31GT94.Text = drcheck(dr, "mon_percent_R31GT94")

Percent_R31LE93.Text = drcheck(dr, "mon_percent_R31LE93")

Percent_OtherStop_LT7.Text = drcheck(dr, "mon_percent_OtherStop_LT7")

Percent_OtherStop7_12.Text = drcheck(dr, "mon_percent_OtherStop7_12")

Percent_OtherStop_GT12.Text = drcheck(dr, "mon_percent_OtherStop_GT12")

Percent_PJ_LT8.Text = drcheck(dr, "mon_percent_PJ_LT8")

Percent_PJ8_16.Text = drcheck(dr, "mon_percent_PJ8_16")

Percent_PJ16_24.Text = drcheck(dr, "mon_percent_PJ16_24")

'Percent_PJ_GT24.Text = drcheck(dr, "mon_percent_PJ_GT24")

Percent_PJ24_32.Text = drcheck(dr, "mon_percent_PJ24-32")

Percent_PJ32_40.Text = drcheck(dr, "mon_percent_PJ32-40")

Percent_PJGT40.Text = drcheck(dr, "mon_percent_PJGT40")

Percent_SJ_LT8.Text = drcheck(dr, "mon_percent_SJ_LT8")

Percent_SJ8_16.Text = drcheck(dr, "mon_percent_SJ8_16")

Percent_SJ16_24.Text = drcheck(dr, "mon_percent_SJ16_24")

Percent_SJ_GT24.Text = drcheck(dr, "mon_percent_SJ_GT24")

```
Joint_CNT.Text = drcheck(dr, "mon_Joint_CNT")

XFMR_GT40.Text = drcheck(dr, "mon_XFMR_GT40")

XFMR30_39.Text = drcheck(dr, "mon_XFMR30_39")

XFMR20_29.Text = drcheck(dr, "mon_XFMR20_29")

XFMR10_19.Text = drcheck(dr, "mon_XFMR10_19")

XFMR0_9.Text = drcheck(dr, "mon_XFMR0_9")

W_XFMR_GT40.Text = drcheck(dr, "mon_W_XFMR_GT40")

W_XFMR30_39.Text = drcheck(dr, "mon_W_XFMR30_39")

W_XFMR20_29.Text = drcheck(dr, "mon_W_XFMR20_29")

W_XFMR10_19.Text = drcheck(dr, "mon_W_XFMR10_19")

W_XFMR0_9.Text = drcheck(dr, "mon_W_XFMR0_9")

Percent_XFMR_GT40.Text = drcheck(dr, "mon_percent_XFMR_GT40")

Percent_XFMR30_39.Text = drcheck(dr, "mon_percent_XFMR30_39")

Percent_XFMR20_29.Text = drcheck(dr, "mon_percent_XFMR20_29")

Percent_XFMR10_19.Text = drcheck(dr, "mon_percent_XFMR10_19")

Percent_XFMR0_9.Text = drcheck(dr, "mon_percent_XFMR0_9")

Percent_W_XFMR_GT40.Text = drcheck(dr, "mon_percent_W_XFMR_GT40")

Percent_W_XFMR30_39.Text = drcheck(dr, "mon_percent_W_XFMR30_39")
```

```
        Percent_W_XFMR20_29.Text = drcheck(dr, "mon_percent_W_XFMR20_29")

        Percent_W_XFMR10_19.Text = drcheck(dr, "mon_percent_W_XFMR10_19")

        Percent_W_XFMR0_9.Text = drcheck(dr, "mon_percent_W_XFMR0_9")

        XFMR_CNT_SUMMED.Text = drcheck(dr, "mon_XFMR_CNT_SUMMED")

        fJeopardy.Text = drcheck(dr, "Jeopardy")

        fCNT.Text = drcheck(dr, "CNT")




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If

End Sub


Private Sub visited_sections_check(ByVal id As String, ByVal isSection As Boolean)


    Dim dvCablesInRun As DataView
```

```
Dim currYear As Integer

Dim j As Integer

Dim cableyear As String

Dim fromStr As String

Dim toStr As String

Dim feederId As String

currYear = 2008



'VS: New added

Dim insulation As String







If (isSection = False) Then

   'this function is more generic than the nam ovl lets on to

   dvCablesInRun = New DataView(GetOVLLoadTableForRun(id))

Else

   dvCablesInRun = New DataView(GetCableById(id))
```

```
End If

'SectionsSummary.Items.Add(dvCablesInRun.Item(0).Item("from_struct"))

For j = 0 To (dvCablesInRun.Table.Rows.Count - 1)
  If (dvCablesInRun.Item(j).Item("INSTALLATION_DATE").Equals(DBNull.Value))
Then

    cableYear = 1966
  Else
    cableYear = dvCablesInRun.Item(j).Item("INSTALLATION_DATE")
  End If

  feederId = dvCablesInRun.Item(j).Item("Feeder")
  ' if (isDBNull(dvCablesInRun.Item(j).Item("insulation")) then insulation 07/30/2008
what should we set if insulation = NULL Shringika
  If dvCablesInRun.Item(j).Item("insulation").Equals(DBNull.Value) Then
```

```
    insulation = "paper"
Else
    insulation = dvCablesInRun.Item(j).Item("insulation")
End If

If IsDBNull(dvCablesInRun.Item(j).Item("from_struct")) Then
    fromStr = 100000000
Else

    fromStr = dvCablesInRun.Item(j).Item("from_struct")
End If

If IsDBNull(dvCablesInRun.Item(j).Item("to_struct")) Then
    toStr = 100000000
Else
```

```
            toStr = dvCablesInRun.Item(j).Item("to_struct")

        End If




        'sections_area set up

        '

        ' If sections_area.InnerText.Contains(feederId + "." + fromStr.ToString + "." +
toStr.ToString) = False Then

        'lbldup.Text = "x"

        ' Else

        ' visited_sections_count = visited_sections_count + 1



        ' End If



        'vs2220:new changes, not yet reflected in vs2-rerank
```

```
                    'commented on 01/27/2009 for excluding visited sections_couint

                    '    If Not (Equals(SectionsSummary.Items.FindByText("No Sections"), Nothing)) Then
SectionsSummary.Items.Remove("No Sections")

                    ' If (Equals(SectionsSummary.Items.FindByText(feederId + "." + fromStr.ToString + "."
+ toStr.ToString), Nothing)) Then

                    'lbldup.Text = "x"

                    ' Else

                    ' visited_sections_count = visited_sections_count + 1



                    ' End If


                Next


            End Sub


            Private Function GetRunsWithWorstStopTable(ByVal Feeder As String) As DataTable


                If Feeder.Length < 5 Then
```

```
Feeder = "0" + Feeder.ToString

End If

' Dim sql As String = "SELECT [Run_ID],[rank] as worst_joint_rank, [elastimold],[RAYCHEM] ,[other_stops],[length], [stops] FROM [Rerank_PILC_Runs_Joints_Types_2008] WHERE feeder = '" & Feeder & "' and stops > 0 order by elastimold desc,raychem desc, other_stops desc "

'commented for deltra mtbf

Dim sql As String = "SELECT Runs.[Run_ID],Runs.[rank] as worst_joint_rank,Runs.[elastimold],Runs.[RAYCHEM] ,Runs.[other_stops],runs.[length], Runs.[stops],cast([delta] as decimal(4,1)) as delta FROM [Rerank_PILC_Runs_Joints_Types_2008] AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & cmbSearchFeeder.SelectedValue & "' and stops > 0 order by elastimold desc,raychem desc, other_stops desc"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

'takes 30 secs to do!

myCommand.SelectCommand.CommandTimeout = 180
```

```
Dim dt As DataTable = New DataTable



myCommand.Fill(dt)



Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

Dim secColumn As DataColumn = New DataColumn

With totalColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "tsec"

'    .Expression = "sum(length)"

End With

With perColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "psec"

End With

With pertotColumn
```

```
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "ptsec"
End With
With secColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "sec"
End With
With dt.Columns
    .Add(totalColumn)
    .Add(perColumn)
    .Add(pertotColumn)
    .Add(secColumn)
End With
myCommand = Nothing
myConnection.Close()
```

```
        myConnection = Nothing


        Return dt

    End Function


    Protected Sub RelPercentElastimold_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles RelPercentElastimold.SelectedIndexChanged

        RelElastimoldCalculationReset()

    End Sub


    Private Sub RelElastimoldCalculationReset()

        If cmbSearchFeeder.Items.Count > 0 Then


            TargetCostCheck.Visible = False
```

```
SectionsChangedRelElastimold.Text = ""

TotalCostRelElastimold.Text = ""

'actualPercentPILC.Text = ""



UpdateAttributesElast.Visible = False

CommitChangesElast.Visible = False

CheckCommitElast.Checked = False

RankBaseRelElast.Text = ""

FOTBaseRelElast.Text = ""

MTBFBaseRelElast.Text = ""

RankNewRelElast.Text = ""

RankChangeRelElast.Text = ""

MTBFNewRelElast.Text = ""

MTBFChangeRelElast.Text = ""

FOTNewRelElast.Text = ""

FOTChangeRelElast.Text = ""



GVRelElastPanel.Visible = False
```

```
        RelGVElastPanel.Visible = False

        GVRelElastStops.Visible = False

        RelGVElast.Visible = False




    End If



End Sub

Private Sub RelXLPCalculationReset()

    If cmbSearchFeeder.Items.Count > 0 Then



        SectionsChangedRel.Text = ""

        TotalCostRel.Text = ""

        'actualPercentPILC.Text = ""



        UpdateAttributesRel.Visible = False
```

```
CommitChangesRel.Visible = False

CheckCommitRel.Checked = False

RankBaseRel.Text = ""

FOTBaseRel.Text = ""

MTBFBaseRel.Text = ""

RankNewRel.Text = ""

RankChangeRel.Text = ""

MTBFNewRel.Text = ""

MTBFChangeRel.Text = ""

FOTNewRel.Text = ""

FOTChangeRel.Text = ""



XLP.Visible = False

GridViewXLP.Visible = False




End If
```

```
End Sub

Private Function feeder_withzeroes_fix(ByVal feeder As String) As String

  If feeder.Length < 5 Then
    feeder = "0" + feeder.ToString
  End If
  Return feeder
End Function

Private Function NewAttributes_Check(ByVal item As String) As String
  If IsDBNull(item) Then Return "9999" Else
  If Int32.Parse(item) < 0 Then Return "0" Else Return item

End Function
```

```
Private Sub Update_Attributes_Fix_For_Negative_Attribute_Values(ByVal id As String,
ByVal isSection As Boolean)

    Dim totalJoints As Integer
    'Dim attributes As DataTable
    Dim fromtostructs As DataTable
    ' Dim dr As DataView
    Dim dr As DataRow
    Dim feeder_id As String
    feeder_id = cmbSearchFeeder.SelectedValue
    If feeder_id.Length < 5 Then feeder_id = "0" + feeder_id
    'Uattr.ChangeFeeder(feeder_id)

    If (isSection = False) Then
      'this function is more generic than the nam ovl lets on to
      'attributes = upd.GetAttributeVectors(id, False, feeder_id)

      dr = Uattr.CountMe(id, False)
```

```
    fromtostructs = Uattr.GetFromToStruct(id, False)

Else

    'attributes = upd.GetAttributeVectors(id, True, feeder_id)


    dr = Uattr.CountMe(id, True)

    fromtostructs = Uattr.GetFromToStruct(id, True)


End If


'mean imputation stats fon canonical feeders

'Insulationcnt avg_date

'NULL                    256     1979

'XLP                     45099   1983

'PAPER           24976           1959

'EPR                     58125   1999


Dim k As Integer

Dim dtr As DataRow
```

```
Dim mon_AllStop_forJointstotal, mon_AllPaper_ForJointsTotal As String

'commented on 08/16/2009 for new update attributes code


For k = 0 To (fromtostructs.Rows.Count - 1)

  dtr = fromtostructs.Rows(k)

  If IsDBNull(dtr("feeder")) = True And IsDBNull(dtr("from_struct")) = True And
IsDBNull(dtr("to_struct")) = True Then

    'Exit Sub

    Continue For

  End If


  If (Equals(SectionsSummary.Items.FindByText(dtr("feeder") + "." + dtr("from_struct")
+ "." + dtr("to_struct")), Nothing)) Then

    SectionsSummary.Items.Add(dtr("feeder") + "." + dtr("from_struct") + "." +
dtr("to_struct"))

    sections_area.InnerText += System.Environment.NewLine + dtr("feeder") + "." +
dtr("from_struct") + "." + dtr("to_struct")

    'Else

    'commented on 01/27/2009
```

```
'duplicate_sections_list = duplicate_sections_list + 1

' duplicate_sections_list = 0


End If

Next

'Updating Counts for cable sections

mon_Paper_GT40.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) - Int32.Parse(dr("iPaperGT40"))))

mon_Paper_LE40.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) - Int32.Parse(dr("iPaperLE40"))))

'dont need to add and subtract iSolidLT16 agb

mon_Solid_LT16.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Solid_LT16.Text) + Int32.Parse(dr("iPaperGT40")) +
Int32.Parse(dr("iPaperLE40")) + Int32.Parse(dr("isolidGE16")) + Int32.Parse(dr("iSolid70_75"))))

mon_Solid_GT16.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Solid_GT16.Text) - Int32.Parse(dr("isolidGE16"))))

mon_Solid_70_75.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Solid_70_75.Text) - Int32.Parse(dr("iSolid70_75"))))
```

'Updating Counts for joints

'first updating total stop and paper joints for getting total joint counts at the end

mon_AllStop_forJointstotal = Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) + Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) + Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) + Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))

mon_AllPaper_ForJointsTotal = Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) + Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) + Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) + Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))

'commented on 04/05/2009

' mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) + Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) + Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) + Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))

'mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) - Int32.Parse(dr("papLT8")) + Int32.Parse(mon_PJ8_16.Text) - Int32.Parse(dr("papBet8A16")) + Int32.Parse(mon_PJ16_24.Text) - Int32.Parse(dr("papBet16A24")) + Int32.Parse(mon_PJ24_32.Text) - Int32.Parse(dr("papBet24A32")) + Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40")) + Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40")))

'modificat6ion on 04/05/2009 ends here

'Elastimold Stop - just remove, don't replace

mon_E21GT93.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993"))))

mon_E21LE93.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993"))))

'Raychem Stop - just remove, don't replace

mon_R31GT94.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993"))))

mon_R31LE93.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993"))))

```
'Other Stop - just remove, don't replace

mon_OtherStop_LT7.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7"))))

mon_OtherStop7_12.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_OtherStop7_12.Text) -
Int32.Parse(dr("oStopBet7A12"))))

mon_OtherStop_GT12.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_OtherStop_GT12.Text) -
Int32.Parse(dr("oStopGT12"))))

'Updating total stops

'enabled on 04/05/2009

mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) +
Int32.Parse(mon_E21LE93.Text) + Int32.Parse(mon_R31GT94.Text) + Int32.Parse(mon_R31LE93.Text) +
Int32.Parse(mon_OtherStop_LT7.Text) + Int32.Parse(mon_OtherStop7_12.Text) +
Int32.Parse(mon_OtherStop_GT12.Text))

'mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) -
Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) +
Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) -
```

Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) + Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) + Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))

'Paper - replace

mon_PJ_LT8.Text = NewAttributes_Check((Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) - Int32.Parse(dr("papLT8")))))

mon_PJ8_16.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ8_16.Text) - Int32.Parse(dr("papBet8A16"))))

mon_PJ16_24.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ16_24.Text) - Int32.Parse(dr("papBet16A24"))))

mon_PJ24_32.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ24_32.Text) - Int32.Parse(dr("papBet24A32"))))

mon_PJ32_40.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40"))))

mon_PJGT40.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40"))))

'Updating total paper

' mon_AllPaper.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) + Int32.Parse(mon_PJ8_16.Text) +

```
Int32.Parse(mon_PJ16_24.Text) + Int32.Parse(mon_PJ24_32.Text) + Int32.Parse(mon_PJ32_40.Text) +
Int32.Parse(mon_PJGT40.Text)))

' mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) -
Int32.Parse(dr("papLT8")) + Int32.Parse(mon_PJ8_16.Text) - Int32.Parse(dr("papBet8A16")) +
Int32.Parse(mon_PJ16_24.Text) - Int32.Parse(dr("papBet16A24")) + Int32.Parse(mon_PJ24_32.Text) -
Int32.Parse(dr("papBet24A32")) + Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40")) +
Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40")))

mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) +
Int32.Parse(mon_PJ8_16.Text) + Int32.Parse(mon_PJ16_24.Text) + Int32.Parse(mon_PJ24_32.Text) +
Int32.Parse(mon_PJ32_40.Text) + Int32.Parse(mon_PJGT40.Text))

'Solid - update date

mon_SJ8_16.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ8_16.Text) - Int32.Parse(dr("solBet8A16"))))

mon_SJ16_24.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ16_24.Text) - Int32.Parse(dr("solBet16A24"))))

mon_SJ_GT24.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ_GT24.Text) - Int32.Parse(dr("solGT24"))))

'Finally, adding all to be replaced and date updated to solid
```

```
mon_SJ_LT8.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ_LT8.Text) + Int32.Parse(dr("papLT8")) +
Int32.Parse(dr("papBet8A16")) + Int32.Parse(dr("papBet16A24")) + Int32.Parse(dr("papBet24A32")) +
Int32.Parse(dr("papBet32A40")) + Int32.Parse(dr("papGE40")) + Int32.Parse(dr("solBet8A16")) +
Int32.Parse(dr("solBet16A24")) + Int32.Parse(dr("solGT24")) + Int32.Parse(dr("leadWipeCount")) +
Int32.Parse(dr("raychem1w1w")) + Int32.Parse(dr("raychem3w1w")) + Int32.Parse(dr("other1w1w"))))

' Next

'Updating Section cost text boxes

SectionsChangedSum.Text = SectionsSummary.Items.Count

CostPerSectionSum.Text = "18000"

TotalCostSum.Text = SectionsSummary.Items.Count * 18000

'Updating percentages for cable sections

mon_Percent_Paper_GT40.Text = Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) /
Int32.Parse(Sec_CNT.Text) * 100.0)
```

```
mon_Percent_Paper_LE40.Text = Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) /
Int32.Parse(Sec_CNT.Text) * 100.0)

mon_Percent_Solid_LT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_LT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_GT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_GT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_70_75.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_70_75.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

'Updating total solids

mon_AllSolid.Text = Convert.ToString(Int32.Parse(mon_SJ8_16.Text) +
Int32.Parse(mon_SJ16_24.Text) + Int32.Parse(mon_SJ_GT24.Text) + Int32.Parse(mon_SJ_LT8.Text))

'updating total joints

mon_Joint_CNT.Text = Convert.ToString(Int32.Parse(mon_AllStop.Text) +
Int32.Parse(mon_AllPaper.Text) + Int32.Parse(mon_AllSolid.Text))
```

totalJoints = mon_Joint_CNT.Text

'Updating percentages

'Elastimold Stop Percentages

mon_Percent_E21GT93.Text = Convert.ToString((Int32.Parse(mon_E21GT93.Text) / totalJoints) * 100)

mon_Percent_E21LE93.Text = Convert.ToString((Int32.Parse(mon_E21LE93.Text) / totalJoints) * 100)

'Raychem Stop percentages

mon_Percent_R31GT94.Text = Convert.ToString((Int32.Parse(mon_R31GT94.Text) / totalJoints) * 100)

mon_Percent_R31LE93.Text = Convert.ToString((Int32.Parse(mon_R31LE93.Text) / totalJoints) * 100)

'Other Stop percentages

mon_Percent_OtherStop_LT7.Text = Convert.ToString((Int32.Parse(mon_OtherStop_LT7.Text) / totalJoints) * 100)

```
                mon_Percent_OtherStop7_12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop7_12.Text) / totalJoints) * 100)

                mon_Percent_OtherStop_GT12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop_GT12.Text) / totalJoints) * 100)


                'Updating total stop percentages

                mon_Percent_ALLStop.Text = Convert.ToString((Int32.Parse(mon_AllStop.Text) /
totalJoints) * 100)


                'Paper percentages

                mon_Percent_PJ_LT8.Text = Convert.ToString((Int32.Parse(mon_PJ_LT8.Text) /
totalJoints) * 100)

                mon_Percent_PJ8_16.Text = Convert.ToString((Int32.Parse(mon_PJ8_16.Text) /
totalJoints) * 100)

                mon_Percent_PJ16_24.Text = Convert.ToString((Int32.Parse(mon_PJ16_24.Text) /
totalJoints) * 100)

                mon_Percent_PJ24_32.Text = Convert.ToString((Int32.Parse(mon_PJ24_32.Text) /
totalJoints) * 100)

                mon_Percent_PJ32_40.Text = Convert.ToString((Int32.Parse(mon_PJ32_40.Text) /
totalJoints) * 100)
```

```
                mon_Percent_PJGT40.Text = Convert.ToString((Int32.Parse(mon_PJGT40.Text) /
totalJoints) * 100)

                'Updating total paper percentages

                mon_Percent_AllPaper.Text = Convert.ToString((Int32.Parse(mon_AllPaper.Text) /
totalJoints) * 100)

                'mon_Percent_AllPaper.Text =
Convert.ToString((Int32.Parse(mon_AllPaper_ForJointsTotal) / totalJoints) * 100)

                'Solid percentages

                mon_Percent_SJ_LT8.Text = Convert.ToString((Int32.Parse(mon_SJ_LT8.Text) /
totalJoints) * 100)

                mon_Percent_SJ8_16.Text = Convert.ToString((Int32.Parse(mon_SJ8_16.Text) /
totalJoints) * 100)

                mon_Percent_SJ16_24.Text = Convert.ToString((Int32.Parse(mon_SJ16_24.Text) /
totalJoints) * 100)

                mon_Percent_SJ_GT24.Text = Convert.ToString((Int32.Parse(mon_SJ_GT24.Text) /
totalJoints) * 100)

                'Updating total solids
```

```
mon_Percent_AllSolid.Text = Convert.ToString((Int32.Parse(mon_AllSolid.Text) / totalJoints) * 100)


End Sub


Protected Sub CalculateLPWMTBF_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles CalculateLPWMTBF.Click

If TotalLPWCurrent.Text = "" Then Exit Sub

'Dim myService As Maxim = New Maxim()

Dim feeder_Id As String

feeder_Id = New String("X")

feeder_Id = cmbSearchFeeder.SelectedValue

If feeder_Id.Length < 5 Then

  feeder_Id = "0" + feeder_Id


End If
```

```
Dim rms_weight, target_rms_weight As String

rms_weight = Convert.ToString(Int32.Parse(Sum_ClosedOld.Text) + Int32.Parse(Sum_ClosedUNR.Text))

Dim data_new_rank, data_base_rank, base_rank As String


data_base_rank = "feeder=" + feeder_Id.ToString + ";"

data_base_rank += "LPW_SUM_Avg_last_04_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_04_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_04_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_45_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_45_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_45_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_24_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_24_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_24_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_07_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_07_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_07_days=" + TotalLPWCurrent.Text + ";" + "LPW_RMS_Avg_last_04_hrs=" + rms_weight + ";" + "LPW_RMS_Min_last_04_hrs=" + rms_weight + ";" + "LPW_RMS_Max_last_04_hrs=" + rms_weight + ";" + "LPW_RMS_Avg_last_45_days=" + rms_weight + ";" + "LPW_SUM_Min_last_45_days=" + rms_weight + ";" + "LPW_SUM_Max_last_45_days=" + rms_weight + ";" + "LPW_RMS_Avg_last_24_hrs=" + rms_weight + ";" + "LPW_RMS_Min_last_24_hrs=" + rms_weight + ";" + "LPW_RMS_Max_last_24_hrs=" + rms_weight + ";" + "LPW_RMS_Avg_last_07_days=" + rms_weight + ";" + "LPW_RMS_Min_last_07_days=" + rms_weight + ";" + "LPW_RMS_Max_last_07_days=" + rms_weight

'base_rank = myService.WSRerank(data_base_rank)

'base_rank = proxy.rerank(data_base_rank.ToString)
```

'data_base_rank =

"feeder=02M43;Static_Feeder_Jeopardy=5.12;Static_Feeder_Load=365.0;Static_Emergency_Proj_Load=598.0;Stat
ic_Normal_rating=418.0;Static_Emergency_rating=613.0;Static_Shift=4.17;Static_Four_KV=0;Static_OA_CNT=0;
Static_delta=56.29;Static_TRANSFORMERS=18.0;Static_Feeds_TA=0;Static_Num_Shunt=0;Static_hipot_OA=3;
Static_hipot_freq_Paper_gte_40=0.377000004053;Static_hipot_freq_Paper_Other=0.0719999969006;Static_hipot_f
req_Solid_70_75=0.00600000005215;Static_hipot_freq_Solid_LT16=0.0430000014603;Static_hipot_freq_Solid_G
TE16=0.0359999984503;Static_hipot_freq_E21_pre93=0.0;Static_hipot_freq_Other_stops=0.122000001371;Static
_hipot_freq_R31_post93=0.25799998641;Static_hipot_freq_Paper_ns=0.234999999404;Static_hipot_freq_Solid_ns
=0.01600000076;Static_hipot_freq_Sum=1.22000002861;Static_hipot_freq_Norm_Sum=0.127000004053;Static_hi
pot_Norm_Shift=0.173999994993;Static_hipot_Norm_Delta=0.910000026226;Static_hipot_Norm_OA=0.5;Static_
hipot_freq_Wtd_Nor=0.389999985695;Static_hipot_GT_half=1;Static_hipot_Norm_WT_Nrm=0.683000028133;St
atic_hipot_freq_E21_post93=0.0;Static_hipot_freq_R31_pre93=0.0560000017285;Static_PaperC_GE_40=41;Static
_PaperOther=9;Static_PaperC_Total=50;Static_Solid_LT_16=40;Static_Solid_70_75=2;Static_Solid_Total=58;Stat
ic_Solid_GE_16=16;Static_total_sections=108;Static_Percent_PaperC_GE_40=37.962962963;Static_Percent_Pape
rOther=8.33333333333;Static_Percent_PaperC_Total=46.2962962963;Static_Percent_Solid_LT_16=37.037037037
;Static_Percent_Solid_70_75=1.85185185185;Static_Percent_Solid_GE_16=14.8148148148;Static_Percent_Solid_
Total=53.7037037037;Static_E2_1_LE_1993=0;Static_E2_1_GT_1993=0;Static_E2_1_Total=0;Static_R3_1_LE_1
993=2;Static_R3_1_GT_1993=12;Static_R3_1_Total=14;Static_Other_Stops_LT_7=3;Static_Other_Stops_7_12=2
;Static_Other_Stops_GT_12=10;Static_Other_Stops_Total=15;Static_Total_Stops=29;Static_PAPER_LT_8=0;Stati
c_PAPER_8_16=0;Static_PAPER_16_24=2;Static_PAPER_24_32=1;Static_PAPER_32_40=5;Static_PAPER_GT
_40=24;Static_PAPER_Total=32;Static_SolidJoints_LT_8=29;Static_SolidJoints_8_16=5;Static_SolidJoints_16_24
=4;Static_SolidJoints_GT_24=0;Static_SolidJoints_Total=38;Static_total_joints=99;Static_Percent_E2_1_LE_1993
=0.0;Static_Percent_E2_1_GT_1993=0.0;Static_Percent_E2_1_Total=0.0;Static_Percent_R3_1_LE_1993=2.02020
20202;Static_Percent_R3_1_GT_1993=12.1212121212;Static_Percent_R3_1_Total=14.1414141414;Static_Percent
_Other_Stops_LT_7=3.0303030303;Static_Percent_Other_Stops_7_12=2.0202020202;Static_Percent_Other_Stops
_GT_12=10.101010101;Static_Percent_Other_Stops_Total=15.1515151515;Static_Percent_Total_Stops=29.29292
92929;Static_Percent_PAPER_LT_8=0.0;Static_Percent_PAPER_8_16=0.0;Static_Percent_PAPER_16_24=2.0202

020202;Static_Percent_PAPER_24_32=1.0101010101;Static_Percent_PAPER_32_40=5.05050505051;Static_Perce
nt_PAPER_GT_40=24.2424242424;Static_Percent_PAPER_Total=32.3232323232;Static_Percent_SolidJoints_LT
_8=29.2929292929;Static_Percent_SolidJoints_8_16=5.05050505051;Static_Percent_SolidJoints_16_24=4.040404
0404;Static_Percent_SolidJoints_GT_24=0.0;Static_Percent_SolidJoints_Total=38.3838383838;Static_Transformer
_0_9=6;Static_Transformer_10_19=3;Static_Transformer_20_29=1;Static_Transformer_30_39=3;Static_Transform
er_GE_40=2;Static_Transformer_ALL=15;Static_West_Transformer_0_9=0;Static_West_Transformer_10_19=2;St
atic_West_Transformer_20_29=0;Static_West_Transformer_30_39=0;Static_West_Transformer_GE_40=0;Static_
West_Transformer_ALL=2;Static_total_transformers=17;Static_Percent_Transformer_0_9=35.2941176471;Static_
Percent_Transformer_10_19=17.6470588235;Static_Percent_Transformer_20_29=5.88235294118;Static_Percent_
Transformer_30_39=17.6470588235;Static_Percent_Transformer_GE_40=11.7647058824;Static_Percent_Transfor
mer_ALL=88.2352941176;Static_Percent_West_Transformer_0_9=0.0;Static_Percent_West_Transformer_10_19=
11.7647058824;Static_Percent_West_Transformer_20_29=0.0;Static_Percent_West_Transformer_30_39=0.0;Static
_Percent_West_Transformer_GE_40=0.0;Static_Percent_West_Transformer_ALL=11.7647058824;Static_SH1_CO
UNT=0;Static_SH1_TOTAL=0.0;Static_SH3_COUNT=1;Static_SH3_TOTAL=15.0;Static_SH5_COUNT=5;Static
_SH5_TOTAL=46.41;Static_SH10_COUNT=7;Static_SH10_TOTAL=57.83;Static_SH16_COUNT=16;Static_SH
16_TOTAL=61.77;Static_SH32_COUNT=27;Static_SH32_TOTAL=63.42;Static_SH1_AVG=0.0;Static_SH3_AV
G=15.0;Static_SH5_AVG=3.0;Static_SH10_AVG=8.26142857143;Static_SH16_AVG=3.860625;Static_SH32_AV
G=2.34888888889;Static_Feeder_CNT=28;Static_Sum_LPW_20050921=30.0;Static_RMS_LPW_20050921=10.5;
Static_Avg_LPW_20050921=1.76470588235;Static_Sum_LPW_20050824=25.5;Static_RMS_LPW_20050824=6.
5;Static_Avg_LPW_20050824=1.5;Static_Sum_LPW_20050816=43.0;Static_RMS_LPW_20050816=12.0;Static_S
um_LPW_20050802=33.0;Static_RMS_LPW_20050802=9.0;Static_Avg_LPW_20050802=1.94117647059;Static_
Avg_LPW_20050816=2.52941176471;Static_Sum_LPW_20050614=49.0;Static_RMS_LPW_20050614=16.0;Stati
c_Avg_LPW_20050614=3.0625;Static_Sections=110;Static_CSR_LE_5percent=0;Static_CSR_LE_10percent=0;St
atic_CSR_LE_15percent=0;Static_CSR_LE_20percent=0;Static_CSR_LE_25percent=1;Static_CSR_LE_30percent
=11;Static_Percent_CSR_LE_5percent=0.0;Static_Percent_CSR_LE_10percent=0.0;Static_Percent_CSR_LE_15pe
rcent=0.0;Static_Percent_CSR_LE_20percent=0.0;Static_Percent_CSR_LE_25percent=0.90909090909;Static_Perc
ent_CSR_LE_30percent=10.0;Static_joints=99;Static_JR_LE_5percent=17;Static_JR_LE_10percent=29;Static_JR_

LE_15percent=35;Static_JR_LE_20percent=38;Static_JR_LE_25percent=56;Static_JR_LE_30percent=61;Static_Percent_JR_LE_5percent=17.1717171717;Static_Percent_JR_LE_10percent=29.2929292929;Static_Percent_JR_LE_15percent=35.3535353535;Static_Percent_JR_LE_20percent=38.3838383838;Static_Percent_JR_LE_25percent=56.5656565657;Static_Percent_JR_LE_30percent=61.6161616162;Static_mon_AllStop=29;Static_mon_percent_AllStop=29.2929292929;Static_mon_AllPaper=32;Static_mon_percent_AllPaper=32.3232323232;Static_mon_AllSolid=34;Static_mon_percent_AllSolid=34.3434343434;OUT_Auto_full_history=4;OUT_Auto_allprev_season=1;OUT_Auto_prev_season=0;OUT_Auto_last_45days=0;OUT_Auto_last_7days=0;OUT_CIOA_full_history=0;OUT_CIOA_allprev_season=0;OUT_CIOA_prev_season=0;OUT_CIOA_last_45days=0;OUT_CIOA_last_7days=0;OUT_FOT_full_history=0;OUT_FOT_allprev_season=0;OUT_FOT_prev_season=0;OUT_FOT_last_45days=0;OUT_FOT_last_7days=0;OUT_OOE_full_history=6;OUT_OOE_allprev_season=1;OUT_OOE_prev_season=1;OUT_OOE_last_45days=1;OUT_OOE_last_7days=1;OUT_Other_full_history=14;OUT_Other_allprev_season=8;OUT_Other_prev_season=0;OUT_Other_last_45days=0;OUT_Other_last_7days=0;LPW_SUM_Min_last_45_days=1.0;LPW_SUM_Min_last_07_days=6.0;LPW_SUM_Min_last_24_hrs=10.0;LPW_SUM_Min_last_04_hrs=10.0;LPW_SUM_Max_last_45_days=37.0;LPW_SUM_Max_last_07_days=37.0;LPW_SUM_Max_last_24_hrs=10.0;LPW_SUM_Max_last_04_hrs=10.0;LPW_SUM_Avg_last_45_days=9.0;LPW_SUM_Avg_last_07_days=11.0;LPW_SUM_Avg_last_24_hrs=10.0;LPW_SUM_Avg_last_04_hrs=10.0;LPW_RMS_Min_last_45_days=?;LPW_RMS_Min_last_07_days=1.0;LPW_RMS_Min_last_24_hrs=1.0;LPW_RMS_Min_last_04_hrs=1.0;LPW_RMS_Max_last_45_days=33.0;LPW_RMS_Max_last_07_days=33.0;LPW_RMS_Max_last_24_hrs=1.0;LPW_RMS_Max_last_04_hrs=1.0;LPW_RMS_Avg_last_45_days=2.0;LPW_RMS_Avg_last_07_days=2.0"

```
base_rank = WebserviceClient.WebserviceClient.proxy_rerank(data_base_rank.ToString)

'lpwoutcome.Text = Chr(34) + data_base_rank.ToString + Chr(34)

lpwoutcome.Text = base_rank.ToString

Dim i As Integer

i = base_rank.IndexOf(",")
```

```
RankBaseLPW.Text = base_rank.Substring(i + 1)

Dim score As Double

score = Double.Parse(base_rank.Substring(0, i - 1))

'MTBFBaseLPW.Text = -(112.33 * score) + 278.71

MTBFBaseLPW.Text = (1.1733 * score) + 1.6807

' MTBFBaseLPW.Text = 1 / Math.Round(1.1638 * score + 1.6793, 3)

'1/round(1.1638*BR.SCORE+1.6793,3)

Dim lpwmtbf_base As Integer

lpwmtbf_base = Math.Round(1.1733 * score) + 1.6807

If lpwmtbf_base = 0 Then

   MTBFBaseLPW.Text = 4999
```

```
Else

    MTBFBaseLPW.Text = 1 / Math.Round(1.1733 * score + 1.6807, 3) * 365


End If
```

```
target_rms_weight = Convert.ToString(Int32.Parse(Target_Sum_ClosedOld.Text) + Int32.Parse(Target_Sum_ClosedUNR.Text))

data_new_rank = "feeder=" + feeder_Id.ToString + ";"

data_new_rank += "LPW_SUM_Avg_last_04_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_04_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_04_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_45_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_45_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_45_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_24_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_24_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_24_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_07_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_07_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_07_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_RMS_Avg_last_04_hrs=" + target_rms_weight + ";" + "LPW_RMS_Min_last_04_hrs=" + target_rms_weight + ";" + "LPW_RMS_Max_last_04_hrs=" +
```

```
target_rms_weight + ";" + "LPW_RMS_Avg_last_45_days=" + target_rms_weight + ";" +
"LPW_RMS_Min_last_45_days=" + target_rms_weight + ";" + "LPW_RMS_Max_last_45_days=" +
target_rms_weight + ";" + "LPW_RMS_Avg_last_24_hrs=" + target_rms_weight + ";" +
"LPW_RMS_Min_last_24_hrs=" + target_rms_weight + ";" + "LPW_RMS_Max_last_24_hrs=" +
target_rms_weight + ";" + "LPW_RMS_Avg_last_07_days=" + target_rms_weight + ";" +
"LPW_RMS_Min_last_07_days=" + target_rms_weight + ";" + "LPW_RMS_Max_last_07_days=" +
target_rms_weight
```

```
Dim new_rank As String

'new_rank = myService.WSRerank(data_new_rank)

new_rank = WebserviceClient.WebserviceClient.proxy_rerank(data_new_rank.ToString)

Dim j As Integer

j = new_rank.IndexOf(",")

Dim score_new As Double

score_new = Double.Parse(new_rank.Substring(0, j - 1))



RankNewLPW.Text = new_rank.Substring(j + 1)
```

```
RankChangeLPW.Text = Convert.ToString(Int32.Parse(RankNewLPW.Text) - Int32.Parse(RankBaseLPW.Text))


'MTBFNewLPW.Text = (1.1733 * score_new) + 1.6807

'MTBFNewLPW.Text = -(112.33 * score_new) + 278.71

'MTBFNewLPW.Text = 1 / Math.Round(1.1638 * score_new + 1.6793, 3)


Dim lpwmtbf_new As Integer


lpwmtbf_new = Math.Round(1.1733 * score_new) + 1.6807


If lpwmtbf_new = 0 Then
  MTBFNewLPW.Text = 4999
Else
  MTBFNewLPW.Text = 1 / Math.Round(1.1733 * score_new + 1.6807, 3) * 365


End If
```

```
MTBFNewLPW.Text = Math.Round(Convert.ToDouble(MTBFNewLPW.Text))

MTBFChangeLPW.Text = Convert.ToString(Convert.ToDouble(MTBFNewLPW.Text) -
Convert.ToDouble(MTBFBaseLPW.Text))

MTBFBaseLPW.Text = Math.Round(Convert.ToDouble(MTBFBaseLPW.Text))

MTBFChangeLPW.Text = Math.Round(Convert.ToDouble(MTBFChangeLPW.Text))

FOTBaseLPW.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankBaseLPW.Text) * -0.0095 + 6), 0.0), 0))

FOTNewLPW.Text = Convert.ToString(Math.Max((Int32.Parse(RankNewLPW.Text) * -
0.0095 + 6), 0.0))

FOTChangeLPW.Text = Convert.ToString(Convert.ToDouble(FOTBaseLPW.Text) -
Convert.ToDouble(FOTNewLPW.Text))



FOTBaseLPW.Text = Math.Round(Convert.ToDouble(FOTBaseLPW.Text))

FOTChangeLPW.Text = Math.Round(Convert.ToDouble(FOTChangeLPW.Text))

FOTNewLPW.Text = Math.Round(Convert.ToDouble(FOTNewLPW.Text))

End Sub

Protected Sub GVovlnonsec_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GVovlnonsec.SelectedIndexChanged

'executed when a cell in gridview is selected
```

```
Dim struct As String

struct = GVovlnonsec.SelectedRow.Cells(1).Text


Session("struct") = struct.ToString

' TextBox1.Text = Session("struct")

Dim URL As String


URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"



Dim MyScript As New System.Text.StringBuilder

With MyScript

  .Append("<script type='text/javascript'>")

  '.Append("Response.Redirect(URL.ToString)")

  .Append("window.open('" & URL & "');")


  .Append("</script>")

End With
```

```
'Response.Redirect(URL)

'Server.Transfer(URL.ToString)

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub

Protected Sub GV_OpenSwitches_update()

' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_openswitches As ArrayList

If Session("visited_openSwitches") Is Nothing Then
```

```
    visited_openswitches = New ArrayList
Else
    visited_openswitches = CType(Session("visited_openswitches"), ArrayList)
End If
If visited_openswitches.Count < 1 Then visited_openswitches.Add(0)

If GV_OpenSwitches.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_OpenSwitches.Rows
        Dim cb As CheckBox = row.FindControl("selectswitches")
        If cb IsNot Nothing And cb.Checked Then
            ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

            If visited_openswitches.Contains(id) Then
                Continue For
            Else
                visited_openswitches.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_openswitches") = visited_openswitches



      End If


  End Sub

  Protected Sub GV_fuse_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_fuse As ArrayList

    If Session("visited_fuse") Is Nothing Then

      visited_fuse = New ArrayList

    Else
```

```
    visited_fuse = CType(Session("visited_fuse"), ArrayList)
End If
If visited_fuse.Count < 1 Then visited_fuse.Add(0)

If GV_Fuse.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_Fuse.Rows
        Dim cb As CheckBox = row.FindControl("selectfuse")
        If cb IsNot Nothing And cb.Checked Then
            ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
            If visited_fuse.Contains(id) Then
                Continue For
            Else
                visited_fuse.Add(id)
            End If

            TextBox2.Text += id
```

```
            End If

        Next


        Session("visited_fuse") = visited_fuse




    End If


End Sub

Protected Sub GV_BanksOff_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_BanksOff As ArrayList

    If Session("visited_BanksOff") Is Nothing Then

        visited_BanksOff = New ArrayList

    Else

        visited_BanksOff = CType(Session("visited_BanksOff"), ArrayList)

    End If

    If visited_BanksOff.Count < 1 Then visited_BanksOff.Add(0)
```

```
If GV_BanksOff.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_BanksOff.Rows
    Dim cb As CheckBox = row.FindControl("selectBanksOff")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_BanksOff.Contains(id) Then
        Continue For
      Else
        visited_BanksOff.Add(id)
      End If


      TextBox2.Text += id
    End If
  Next
```

```
        Session("visited_BanksOff") = visited_BanksOff




    End If



End Sub



Protected Sub GV_ClosedOld_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_ClosedOld As ArrayList

    If Session("visited_ClosedOld") Is Nothing Then

        visited_ClosedOld = New ArrayList

    Else

        visited_ClosedOld = CType(Session("visited_ClosedOld"), ArrayList)

    End If

    If visited_ClosedOld.Count < 1 Then visited_ClosedOld.Add(0)



    If GV_ClosedOld.Rows.Count > 0 Then
```

```
For Each row As GridViewRow In GV_ClosedOld.Rows
  Dim cb As CheckBox = row.FindControl("selectClosedOld")
  If cb IsNot Nothing And cb.Checked Then
    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
    If visited_ClosedOld.Contains(id) Then
      Continue For
    Else
      visited_ClosedOld.Add(id)
    End If

    TextBox2.Text += id
  End If
Next

Session("visited_ClosedOld") = visited_ClosedOld
```

```
    End If


End Sub


Protected Sub GV_ClosedUNR_update()
    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
    Dim visited_ClosedUNR As ArrayList
    If Session("visited_ClosedUNR") Is Nothing Then
        visited_ClosedUNR = New ArrayList
    Else
        visited_ClosedUNR = CType(Session("visited_ClosedUNR"), ArrayList)
    End If
    If visited_ClosedUNR.Count < 1 Then visited_ClosedUNR.Add(0)


    If GV_ClosedUNR.Rows.Count > 0 Then


        For Each row As GridViewRow In GV_ClosedUNR.Rows
```

```
            Dim cb As CheckBox = row.FindControl("selectClosedUNR")
            If cb IsNot Nothing And cb.Checked Then
                ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                If visited_ClosedUNR.Contains(id) Then
                    Continue For
                Else
                    visited_ClosedUNR.Add(id)
                End If

                TextBox2.Text += id
            End If
        Next

        Session("visited_ClosedUNR") = visited_ClosedUNR

    End If
```

```
End Sub

Protected Sub GV_ClosedDNU_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_ClosedDNU As ArrayList

  If Session("visited_ClosedDNU") Is Nothing Then

    visited_ClosedDNU = New ArrayList

  Else

    visited_ClosedDNU = CType(Session("visited_ClosedDNU"), ArrayList)

  End If

  If visited_ClosedDNU.Count < 1 Then visited_ClosedDNU.Add(0)



  If GV_ClosedDNU.Rows.Count > 0 Then



    For Each row As GridViewRow In GV_ClosedDNU.Rows

      Dim cb As CheckBox = row.FindControl("selectClosedDNU")

      If cb IsNot Nothing And cb.Checked Then

        ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
```

```
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

            If visited_ClosedDNU.Contains(id) Then

               Continue For

            Else

               visited_ClosedDNU.Add(id)

            End If



            TextBox2.Text += id

         End If

      Next



      Session("visited_ClosedDNU") = visited_ClosedDNU



   End If

End Sub
```

```
Protected Sub GV_OpenMainsLTE2_update()
    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
    Dim visited_OpenMainsLTE2 As ArrayList
    If Session("visited_OpenMainsLTE2") Is Nothing Then
        visited_OpenMainsLTE2 = New ArrayList
    Else
        visited_OpenMainsLTE2 = CType(Session("visited_OpenMainsLTE2"), ArrayList)
    End If
    If visited_OpenMainsLTE2.Count < 1 Then visited_OpenMainsLTE2.Add(0)

    If GV_OpenMainsLTE2.Rows.Count > 0 Then

        For Each row As GridViewRow In GV_OpenMainsLTE2.Rows
            Dim cb As CheckBox = row.FindControl("selectOpenMainsLTE2")
            If cb IsNot Nothing And cb.Checked Then
                ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                If visited_OpenMainsLTE2.Contains(id) Then
```

```
                Continue For

            Else

                visited_OpenMainsLTE2.Add(id)

            End If



            TextBox2.Text += id

        End If

    Next



    Session("visited_OpenMainsLTE2") = visited_OpenMainsLTE2




End If



End Sub

Protected Sub GV_OpenMainsGT2_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_OpenMainsGT2 As ArrayList
```

```
If Session("visited_OpenMainsGT2") Is Nothing Then
   visited_OpenMainsGT2 = New ArrayList
Else
   visited_OpenMainsGT2 = CType(Session("visited_OpenMainsGT2"), ArrayList)
End If
If visited_OpenMainsGT2.Count < 1 Then visited_OpenMainsGT2.Add(0)

If GV_OpenMainsGT2.Rows.Count > 0 Then

   For Each row As GridViewRow In GV_OpenMainsGT2.Rows
      Dim cb As CheckBox = row.FindControl("selectOpenMainsGT2")
      If cb IsNot Nothing And cb.Checked Then
         ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
         Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
         If visited_OpenMainsGT2.Contains(id) Then
            Continue For
         Else
            visited_OpenMainsGT2.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_OpenMainsGT2") = visited_OpenMainsGT2


    End If


End Sub


Protected Sub GV_Missing_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_Missing As ArrayList

    If Session("visited_Missing") Is Nothing Then
```

```
    visited_Missing = New ArrayList
Else
    visited_Missing = CType(Session("visited_Missing"), ArrayList)
End If
If visited_Missing.Count < 1 Then visited_Missing.Add(0)

If GV_Missing.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_Missing.Rows
        Dim cb As CheckBox = row.FindControl("selectMissing")
        If cb IsNot Nothing And cb.Checked Then
            ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

            If visited_Missing.Contains(id) Then
                Continue For
            Else
                visited_Missing.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_Missing") = visited_Missing


      End If


End Sub


Protected Sub GV_OpenSwitches_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_openswitches_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_openSwitches_NbyFactorGTE10") Is Nothing Then
  visited_openswitches_NbyFactorGTE10 = New ArrayList
Else
  visited_openswitches_NbyFactorGTE10 = CType(Session("visited_openswitches_NbyFactorGTE10"), ArrayList)
End If
If visited_openswitches_NbyFactorGTE10.Count < 1 Then visited_openswitches_NbyFactorGTE10.Add(0)

If GV_OpenSwitches_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_OpenSwitches_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectswitches_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_openswitches_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
                Else

                    visited_openswitches_NbyFactorGTE10.Add(id)

                End If



                TextBox2.Text += id

            End If

        Next



        Session("visited_openswitches_NbyFactorGTE10") =
visited_openswitches_NbyFactorGTE10



    End If



End Sub

Protected Sub GV_fuse_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_fuse_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_fuse_NbyFactorGTE10") Is Nothing Then
   visited_fuse_NbyFactorGTE10 = New ArrayList
Else
   visited_fuse_NbyFactorGTE10 = CType(Session("visited_fuse_NbyFactorGTE10"),
ArrayList)
End If
If visited_fuse_NbyFactorGTE10.Count < 1 Then visited_fuse_NbyFactorGTE10.Add(0)

If GV_Fuse_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_Fuse_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectfuse_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_fuse_NbyFactorGTE10.Contains(id) Then
        Continue For
      Else
```

```
            visited_fuse_NbyFactorGTE10.Add(id)

          End If


          TextBox2.Text += id

        End If

      Next


      Session("visited_fuse_NbyFactorGTE10") = visited_fuse_NbyFactorGTE10




    End If


End Sub

Protected Sub GV_BanksOff_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_BanksOff_NbyFactorGTE10 As ArrayList

  If Session("visited_BanksOff_NbyFactorGTE10") Is Nothing Then

    visited_BanksOff_NbyFactorGTE10 = New ArrayList
```

```
                Else

                    visited_BanksOff_NbyFactorGTE10 =
CType(Session("visited_BanksOff_NbyFactorGTE10"), ArrayList)

                End If

                If visited_BanksOff_NbyFactorGTE10.Count < 1 Then
visited_BanksOff_NbyFactorGTE10.Add(0)


                If GV_BanksOff_NbyFactorGTE10.Rows.Count > 0 Then


                  For Each row As GridViewRow In GV_BanksOff_NbyFactorGTE10.Rows
                    Dim cb As CheckBox = row.FindControl("selectBanksOff_GTE10")
                    If cb IsNot Nothing And cb.Checked Then
                      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                      If visited_BanksOff_NbyFactorGTE10.Contains(id) Then
                        Continue For
                      Else
                        visited_BanksOff_NbyFactorGTE10.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_BanksOff_NbyFactorGTE10") = visited_BanksOff_NbyFactorGTE10



      End If


End Sub


Protected Sub GV_ClosedOld_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_ClosedOld_NbyFactorGTE10 As ArrayList

  If Session("visited_ClosedOld_NbyFactorGTE10") Is Nothing Then

    visited_ClosedOld_NbyFactorGTE10 = New ArrayList
```

```
Else

visited_ClosedOld_NbyFactorGTE10 =
CType(Session("visited_ClosedOld_NbyFactorGTE10"), ArrayList)

End If

If visited_ClosedOld_NbyFactorGTE10.Count < 1 Then
visited_ClosedOld_NbyFactorGTE10.Add(0)

If GV_ClosedOld_NbyFactorGTE10.Rows.Count > 0 Then

For Each row As GridViewRow In GV_ClosedOld_NbyFactorGTE10.Rows
Dim cb As CheckBox = row.FindControl("selectClosedOld_GTE10)")
If cb IsNot Nothing And cb.Checked Then
' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
If visited_ClosedOld_NbyFactorGTE10.Contains(id) Then
Continue For
Else
visited_ClosedOld_NbyFactorGTE10.Add(id)
```

```
                End If


                TextBox2.Text += id

            End If

        Next


                Session("visited_ClosedOld_NbyFactorGTE10") =
visited_ClosedOld_NbyFactorGTE10


    End If


End Sub


Protected Sub GV_ClosedUNR_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_ClosedUNR_NbyFactorGTE10 As ArrayList

    If Session("visited_ClosedUNR_NbyFactorGTE10") Is Nothing Then
```

```
visited_ClosedUNR_NbyFactorGTE10 = New ArrayList

Else

visited_ClosedUNR_NbyFactorGTE10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE10"), ArrayList)

End If

If visited_ClosedUNR_NbyFactorGTE10.Count < 1 Then
visited_ClosedUNR_NbyFactorGTE10.Add(0)


If GV_ClosedUNR_NbyFactorGTE10.Rows.Count > 0 Then


For Each row As GridViewRow In GV_ClosedUNR_NbyFactorGTE10.Rows
Dim cb As CheckBox = row.FindControl("selectClosedUNR_GTE10")
If cb IsNot Nothing And cb.Checked Then
' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
If visited_ClosedUNR_NbyFactorGTE10.Contains(id) Then
Continue For
Else
```

```
                visited_ClosedUNR_NbyFactorGTE10.Add(id)

            End If


            TextBox2.Text += id

        End If

    Next


    Session("visited_ClosedUNR_NbyFactorGTE10") =
visited_ClosedUNR_NbyFactorGTE10



End If


End Sub

Protected Sub GV_ClosedDNU_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_ClosedDNU_NbyFactorGTE10 As ArrayList

    If Session("visited_ClosedDNU_NbyFactorGTE10") Is Nothing Then
```

```
visited_ClosedDNU_NbyFactorGTE10 = New ArrayList

Else

visited_ClosedDNU_NbyFactorGTE10 = CType(Session("visited_ClosedDNU_NbyFactorGTE10"), ArrayList)

End If

If visited_ClosedDNU_NbyFactorGTE10.Count < 1 Then visited_ClosedDNU_NbyFactorGTE10.Add(0)

If GV_ClosedDNU_NbyFactorGTE10.Rows.Count > 0 Then

For Each row As GridViewRow In GV_ClosedDNU_NbyFactorGTE10.Rows
Dim cb As CheckBox = row.FindControl("selectClosedDNU_GTE10")
If cb IsNot Nothing And cb.Checked Then
' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
If visited_ClosedDNU_NbyFactorGTE10.Contains(id) Then
Continue For
Else
```

```
                visited_ClosedDNU_NbyFactorGTE10.Add(id)

            End If


            TextBox2.Text += id

        End If

    Next


    Session("visited_ClosedDNU_NbyFactorGTE10") =
visited_ClosedDNU_NbyFactorGTE10


End If


End Sub


Protected Sub GV_missing_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_missing_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_missing_NbyFactorGTE10") Is Nothing Then
  visited_missing_NbyFactorGTE10 = New ArrayList
Else
  visited_missing_NbyFactorGTE10 = CType(Session("visited_missing_NbyFactorGTE10"), ArrayList)
End If
If visited_missing_NbyFactorGTE10.Count < 1 Then visited_missing_NbyFactorGTE10.Add(0)

If GV_Missing_NbyFActorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_Missing_NbyFActorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectMissing_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_missing_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
            Else

               visited_missing_NbyFactorGTE10.Add(id)

            End If



            TextBox2.Text += id

         End If

      Next



      Session("visited_Missing_NbyFactorGTE10") = visited_missing_NbyFactorGTE10



   End If



End Sub



Protected Sub GV_OpenMainsLTE2_NbyFactorGTE10_update()

   ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

   Dim visited_OpenMainsLTE2_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_OpenMainsLTE2_NbyFactorGTE10") Is Nothing Then
  visited_OpenMainsLTE2_NbyFactorGTE10 = New ArrayList
Else
  visited_OpenMainsLTE2_NbyFactorGTE10 = CType(Session("visited_OpenMainsLTE2_NbyFactorGTE10"), ArrayList)
End If
If visited_OpenMainsLTE2_NbyFactorGTE10.Count < 1 Then visited_OpenMainsLTE2_NbyFactorGTE10.Add(0)

If GV_OpenMainsLTE2_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_OpenMainsLTE2_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectOpenMainsLTE2_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_OpenMainsLTE2_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
                Else

                    visited_OpenMainsLTE2_NbyFactorGTE10.Add(id)

                End If



                TextBox2.Text += id

            End If

        Next



        Session("visited_OpenMainsLTE2_NbyFactorGTE10") =
visited_OpenMainsLTE2_NbyFactorGTE10



    End If



End Sub

Protected Sub GV_OpenMainsGT2_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_OpenMainsGT2_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_OpenMainsGT2_NbyFactorGTE10") Is Nothing Then
  visited_OpenMainsGT2_NbyFactorGTE10 = New ArrayList
Else
  visited_OpenMainsGT2_NbyFactorGTE10 = CType(Session("visited_OpenMainsGT2_NbyFactorGTE10"), ArrayList)
End If
If visited_OpenMainsGT2_NbyFactorGTE10.Count < 1 Then visited_OpenMainsGT2_NbyFactorGTE10.Add(0)

If GV_OpenMainsGT2_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_OpenMainsGT2_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectOpenMainsGT2_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_OpenMainsGT2_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
                Else

                    visited_OpenMainsGT2_NbyFactorGTE10.Add(id)

                End If



                TextBox2.Text += id

            End If

        Next



        Session("visited_OpenMainsGT2_NbyFactorGTE10") =
visited_OpenMainsGT2_NbyFactorGTE10



    End If



End Sub
```

```
Protected Sub GV_OpenSwitches_NbyFactorGTE5LT10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_openswitches_NbyFactorGTE5LT10 As ArrayList

  If Session("visited_openSwitches_NbyFactorGTE5LT10") Is Nothing Then

    visited_openswitches_NbyFactorGTE5LT10 = New ArrayList

  Else

    visited_openswitches_NbyFactorGTE5LT10 =
CType(Session("visited_openswitches_NbyFactorGTE5LT10"), ArrayList)

  End If

  If visited_openswitches_NbyFactorGTE5LT10.Count < 1 Then
visited_openswitches_NbyFactorGTE5LT10.Add(0)


  If GV_OpenSwitches_NbyFactorGTE5LT10.Rows.Count > 0 Then


    For Each row As GridViewRow In GV_OpenSwitches_NbyFactorGTE5LT10.Rows

      Dim cb As CheckBox = row.FindControl("selectswitches_GTE5LT10")
```

```
                If cb IsNot Nothing And cb.Checked Then

                    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

                    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

                    If visited_openswitches_NbyFactorGTE5LT10.Contains(id) Then

                        Continue For

                    Else

                        visited_openswitches_NbyFactorGTE5LT10.Add(id)

                    End If



                    TextBox2.Text += id

                End If

            Next



            Session("visited_openswitches_NbyFactorGTE5LT10") =
visited_openswitches_NbyFactorGTE5LT10



        End If
```

```
End Sub

Protected Sub GV_fuse_NbyFactorGTE5LT10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_fuse_NbyFactorGTE5LT10 As ArrayList

  If Session("visited_fuse_NbyFactorGTE5LT10") Is Nothing Then

    visited_fuse_NbyFactorGTE5LT10 = New ArrayList

  Else

    visited_fuse_NbyFactorGTE5LT10 =
CType(Session("visited_fuse_NbyFactorGTE5LT10"), ArrayList)

  End If

  If visited_fuse_NbyFactorGTE5LT10.Count < 1 Then
visited_fuse_NbyFactorGTE5LT10.Add(0)


  If GV_Fuse_NbyFactorGTE5LT10.Rows.Count > 0 Then


    For Each row As GridViewRow In GV_Fuse_NbyFactorGTE5LT10.Rows

      Dim cb As CheckBox = row.FindControl("selectfuse_GTE5LT10")
```

```
If cb IsNot Nothing And cb.Checked Then

  ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

  Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

  If visited_fuse_NbyFactorGTE5LT10.Contains(id) Then

    Continue For

  Else

    visited_fuse_NbyFactorGTE5LT10.Add(id)

  End If



  TextBox2.Text += id

 End If

Next



Session("visited_fuse_NbyFactorGTE5LT10") = visited_fuse_NbyFactorGTE5LT10



End If
```

```
End Sub

Protected Sub GV_BanksOff_NbyFactorGTE5LT10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_BanksOff_NbyFactorGTE5LT10 As ArrayList

  If Session("visited_BanksOff_NbyFactorGTE5LT10") Is Nothing Then

    visited_BanksOff_NbyFactorGTE5LT10 = New ArrayList

  Else

    visited_BanksOff_NbyFactorGTE5LT10 =
CType(Session("visited_BanksOff_NbyFactorGTE5LT10"), ArrayList)

  End If

  If visited_BanksOff_NbyFactorGTE5LT10.Count < 1 Then
visited_BanksOff_NbyFactorGTE5LT10.Add(0)


  If GV_BanksOff_NbyFactorGTE5LT10.Rows.Count > 0 Then


    For Each row As GridViewRow In GV_BanksOff_NbyFactorGTE5LT10.Rows

      Dim cb As CheckBox = row.FindControl("selectBanksOff_GTE5LT10")

      If cb IsNot Nothing And cb.Checked Then
```

```
                    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

                    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

                    If visited_BanksOff_NbyFactorGTE5LT10.Contains(id) Then

                      Continue For

                    Else

                      visited_BanksOff_NbyFactorGTE5LT10.Add(id)

                    End If



                    TextBox2.Text += id

                  End If

                Next



                Session("visited_BanksOff_NbyFactorGTE5LT10")                    =
visited_BanksOff_NbyFactorGTE5LT10



              End If
```

```
End Sub

Protected Sub GV_ClosedOld_NbyFactorGTE5LT10_update()
  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
  Dim visited_ClosedOld_NbyFactorGTE5LT10 As ArrayList
  If Session("visited_ClosedOld_NbyFactorGTE5LT10") Is Nothing Then
    visited_ClosedOld_NbyFactorGTE5LT10 = New ArrayList
  Else
    visited_ClosedOld_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedOld_NbyFactorGTE5LT10"), ArrayList)
  End If
  If visited_ClosedOld_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedOld_NbyFactorGTE5LT10.Add(0)

  If GV_ClosedOld_NbyFactorGTE5LT10.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_ClosedOld_NbyFactorGTE5LT10.Rows
      Dim cb As CheckBox = row.FindControl("selectClosedOld_GTE5LT10")
```

```
                If cb IsNot Nothing And cb.Checked Then

                    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

                    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

                    If visited_ClosedOld_NbyFactorGTE5LT10.Contains(id) Then

                        Continue For

                    Else

                        visited_ClosedOld_NbyFactorGTE5LT10.Add(id)

                    End If



                    TextBox2.Text += id

                End If

            Next



            Session("visited_ClosedOld_NbyFactorGTE5LT10") =
visited_ClosedOld_NbyFactorGTE5LT10



        End If
```

```
End Sub

Protected Sub GV_ClosedUNR_NbyFactorGTE5LT10_update()
  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
  Dim visited_ClosedUNR_NbyFactorGTE5LT10 As ArrayList
  If Session("visited_ClosedUNR_NbyFactorGTE5LT10") Is Nothing Then
    visited_ClosedUNR_NbyFactorGTE5LT10 = New ArrayList
  Else
    visited_ClosedUNR_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE5LT10"), ArrayList)
  End If
  If visited_ClosedUNR_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedUNR_NbyFactorGTE5LT10.Add(0)

  If GV_ClosedUNR_NbyFactorGTE5LT10.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_ClosedUNR_NbyFactorGTE5LT10.Rows
```

```
                Dim cb As CheckBox = row.FindControl("selectClosedUNR_GTE5LT10")
                If cb IsNot Nothing And cb.Checked Then
                  ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                  Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                  If visited_ClosedUNR_NbyFactorGTE5LT10.Contains(id) Then
                    Continue For
                  Else
                    visited_ClosedUNR_NbyFactorGTE5LT10.Add(id)
                  End If

                  TextBox2.Text += id
                End If
              Next

              Session("visited_ClosedUNR_NbyFactorGTE5LT10") =
visited_ClosedUNR_NbyFactorGTE5LT10
```

```
End If


End Sub

Protected Sub GV_ClosedDNU_NbyFactorGTE5LT10_update()

' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_ClosedDNU_NbyFactorGTE5LT10 As ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE5LT10") Is Nothing Then

visited_ClosedDNU_NbyFactorGTE5LT10 = New ArrayList

Else

visited_ClosedDNU_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE5LT10"), ArrayList)

End If

If visited_ClosedDNU_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedDNU_NbyFactorGTE5LT10.Add(0)


If GV_ClosedDNU_NbyFactorGTE5LT10.Rows.Count > 0 Then


For Each row As GridViewRow In GV_ClosedDNU_NbyFactorGTE5LT10.Rows
```

```
                Dim cb As CheckBox = row.FindControl("selectClosedDNU_GTE5LT10")
                If cb IsNot Nothing And cb.Checked Then
                  ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                  Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                  If visited_ClosedDNU_NbyFactorGTE5LT10.Contains(id) Then
                    Continue For
                  Else
                    visited_ClosedDNU_NbyFactorGTE5LT10.Add(id)
                  End If

                  TextBox2.Text += id
                End If
              Next

              Session("visited_ClosedDNU_NbyFactorGTE5LT10") =
visited_ClosedDNU_NbyFactorGTE5LT10
```

```
        End If


    End Sub



    Protected Sub GV_missing_NbyFactorGTE5LT10_update()
        ' GV_missing.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
        Dim visited_missing_NbyFactorGTE5LT10 As ArrayList
        If Session("visited_missing_NbyFactorGTE5LT10") Is Nothing Then
            visited_missing_NbyFactorGTE5LT10 = New ArrayList
        Else
            visited_missing_NbyFactorGTE5LT10 =
CType(Session("visited_missing_NbyFactorGTE5LT10"), ArrayList)
        End If
        If visited_missing_NbyFactorGTE5LT10.Count < 1 Then
visited_missing_NbyFactorGTE5LT10.Add(0)


        If GV_Missing_NbyFactorGTE5LT10.Rows.Count > 0 Then
```

```
For Each row As GridViewRow In GV_Missing_NbyFactorGTE5LT10.Rows
  Dim cb As CheckBox = row.FindControl("selectMissing_GTE5LT10")
  If cb IsNot Nothing And cb.Checked Then
    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
    If visited_missing_NbyFactorGTE5LT10.Contains(id) Then
      Continue For
    Else
      visited_missing_NbyFactorGTE5LT10.Add(id)
    End If

    TextBox2.Text += id
  End If
Next

Session("visited_missing_NbyFactorGTE5LT10") =
visited_missing_NbyFactorGTE5LT10
```

```
    End If


End Sub


Protected Sub GV_OpenMainsLTE2_NbyFactorGTE5LT10_update()
    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
    Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As ArrayList
    If Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") Is Nothing Then
      visited_OpenMainsLTE2_NbyFactorGTE5LT10 = New ArrayList
    Else
      visited_OpenMainsLTE2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10"), ArrayList)
    End If
    If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Count < 1 Then
visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(0)
```

```
If GV_OpenMainsLTE2_NbyFactorGTE5LT10.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_OpenMainsLTE2_NbyFactorGTE5LT10.Rows

    Dim cb As CheckBox = row.FindControl("selectOpenMainsLTE2_GTE5LT10")

    If cb IsNot Nothing And cb.Checked Then

      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

      If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Contains(id) Then

        Continue For

      Else

        visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(id)

      End If



      TextBox2.Text += id

    End If

  Next
```

```
                Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") =
visited_OpenMainsLTE2_NbyFactorGTE5LT10

                End If

            End Sub

            Protected Sub GV_OpenMainsGT2_NbyFactorGTE5LT10_update()

              ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

              Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As ArrayList

              If Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") Is Nothing Then

                visited_OpenMainsGT2_NbyFactorGTE5LT10 = New ArrayList

              Else

                visited_OpenMainsGT2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE5LT10"), ArrayList)

              End If

              If visited_OpenMainsGT2_NbyFactorGTE5LT10.Count < 1 Then
visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(0)
```

```
If GV_OpenMainsGT2_NbyFactorGTE5LT10.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_OpenMainsGT2_NbyFactorGTE5LT10.Rows

    Dim cb As CheckBox = row.FindControl("selectOpenMainsGT2_GTE5LT10")

    If cb IsNot Nothing And cb.Checked Then

      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

      If visited_OpenMainsGT2_NbyFactorGTE5LT10.Contains(id) Then

        Continue For

      Else

        visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(id)

      End If



      TextBox2.Text += id

    End If

  Next
```

Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") = visited_OpenMainsGT2_NbyFactorGTE5LT10

End If

End Sub

Protected Sub GV_BanksOff_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_BanksOff.SelectedIndexChanged

GV_BanksOff.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_BanksOff As ArrayList

If Session("visited_BanksOff") Is Nothing Then

visited_BanksOff = New ArrayList

Else

visited_BanksOff = CType(Session("visited_BanksOff"), ArrayList)

End If

Dim id As String = GV_BanksOff.SelectedRow.Cells(0).Text

```
If visited_BanksOff.Count < 1 Then visited_BanksOff.Add(0)

If visited_BanksOff.Contains(id) Then

  Exit Sub

Else

  visited_BanksOff.Add(id)

End If



Session("visited_BanksOff") = visited_BanksOff









End Sub



Protected Sub GV_Fuse_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GV_Fuse.SelectedIndexChanged

  GV_Fuse.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
```

```
Dim visited_Fuse As ArrayList
If Session("visited_Fuse") Is Nothing Then
   visited_Fuse = New ArrayList
Else
   visited_Fuse = CType(Session("visited_Fuse"), ArrayList)
End If
Dim id As String = GV_Fuse.SelectedRow.Cells(0).Text
If visited_Fuse.Count < 1 Then visited_Fuse.Add(0)
If visited_Fuse.Contains(id) Then
   Exit Sub
Else
   visited_Fuse.Add(id)
End If

Session("visited_Fuse") = visited_Fuse
```

```
End Sub

Protected Sub GV_ClosedOld_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GV_ClosedOld.SelectedIndexChanged
    GV_ClosedOld.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
    Dim visited_ClosedOld As ArrayList
    If Session("visited_ClosedOld") Is Nothing Then
      visited_ClosedOld = New ArrayList
    Else
      visited_ClosedOld = CType(Session("visited_ClosedOld"), ArrayList)
    End If
    Dim id As String = GV_ClosedOld.SelectedRow.Cells(0).Text
    If visited_ClosedOld.Count < 1 Then visited_ClosedOld.Add(0)
    If visited_ClosedOld.Contains(id) Then
      Exit Sub
    Else
```

```
            visited_ClosedOld.Add(id)
        End If

        Session("visited_fuse") = visited_ClosedOld

    End Sub

    Protected Sub GV_ClosedUNR_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles GV_ClosedUNR.SelectedIndexChanged
        GV_ClosedUNR.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
        Dim visited_ClosedUNR As ArrayList
        If Session("visited_ClosedUNR") Is Nothing Then
            visited_ClosedUNR = New ArrayList
        Else
            visited_ClosedUNR = CType(Session("visited_ClosedUNR"), ArrayList)
```

```
End If

Dim id As String = GV_ClosedUNR.SelectedRow.Cells(0).Text

If visited_ClosedUNR.Count < 1 Then visited_ClosedUNR.Add(0)

If visited_ClosedUNR.Contains(id) Then

  Exit Sub

Else

  visited_ClosedUNR.Add(id)

End If



Session("visited_ClosedUNR") = visited_ClosedUNR



End Sub



Protected Sub GV_ClosedDNU_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles GV_ClosedDNU.SelectedIndexChanged

GV_ClosedDNU.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_ClosedDNU As ArrayList
```

```
If Session("visited_ClosedDNU") Is Nothing Then
    visited_ClosedDNU = New ArrayList
Else
    visited_ClosedDNU = CType(Session("visited_ClosedDNU"), ArrayList)
End If
Dim id As String = GV_ClosedDNU.SelectedRow.Cells(0).Text
If visited_ClosedDNU.Count < 1 Then visited_ClosedDNU.Add(0)
If visited_ClosedDNU.Contains(id) Then
    Exit Sub
Else
    visited_ClosedDNU.Add(id)
End If

Session("visited_ClosedDNU") = visited_ClosedDNU

End Sub
```

```
Protected Sub GV_OpenMainsLTE2_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles GV_OpenMainsLTE2.SelectedIndexChanged

    GV_OpenMainsLTE2.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_OpenMainsLTE2 As ArrayList

    If Session("visited_OpenMainsLTE2") Is Nothing Then

      visited_OpenMainsLTE2 = New ArrayList

    Else

      visited_OpenMainsLTE2 = CType(Session("visited_OpenMainsLTE2"), ArrayList)

    End If

    Dim id As String = GV_OpenMainsLTE2.SelectedRow.Cells(0).Text

    If visited_OpenMainsLTE2.Count < 1 Then visited_OpenMainsLTE2.Add(0)

    If visited_OpenMainsLTE2.Contains(id) Then

      Exit Sub

    Else

      visited_OpenMainsLTE2.Add(id)

    End If



    Session("visited_OpenMainsLTE2") = visited_OpenMainsLTE2
```

```
End Sub

Protected Sub GV_OpenMainsGT2_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles GV_OpenMainsGT2.SelectedIndexChanged

GV_OpenMainsGT2.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_OpenMainsGT2 As ArrayList

If Session("visited_OpenMainsGT2") Is Nothing Then

  visited_OpenMainsGT2 = New ArrayList

Else

  visited_OpenMainsGT2 = CType(Session("visited_OpenMainsGT2"), ArrayList)

End If

Dim id As String = GV_OpenMainsGT2.SelectedRow.Cells(0).Text

If visited_OpenMainsGT2.Count < 1 Then visited_OpenMainsGT2.Add(0)

If visited_OpenMainsGT2.Contains(id) Then

  Exit Sub

Else

  visited_OpenMainsGT2.Add(id)

End If
```

```
    Session("visited_OpenMainsGT2") = visited_OpenMainsGT2



End Sub



Private Sub reset_visitedLPWSessionVariables()

    'reset session variables for the case when transformer is on feeder

    Dim visited_OpenSwitches As New ArrayList

    visited_OpenSwitches.Add(0)

    Session("visited_OpenSwitches") = visited_OpenSwitches



    Dim visited_BanksOff As New ArrayList

    visited_BanksOff.Add(0)

    Session("visited_BanksOff") = visited_BanksOff



    Dim visited_Fuse As New ArrayList

    visited_Fuse.Add(0)
```

```
Session("visited_Fuse") = visited_Fuse



Dim visited_ClosedOld As New ArrayList

visited_ClosedOld.Add(0)

Session("visited_ClosedOld") = visited_ClosedOld



Dim visited_ClosedUNR As New ArrayList

visited_ClosedUNR.Add(0)

Session("visited_ClosedUNR") = visited_ClosedUNR



Dim visited_ClosedDNU As New ArrayList

visited_ClosedDNU.Add(0)

Session("visited_ClosedDNU") = visited_ClosedDNU



Dim visited_OpenMainsLTE2 As New ArrayList

visited_OpenMainsLTE2.Add(0)

Session("visited_OpenMainsLTE2") = visited_OpenMainsLTE2
```

```
Dim visited_OpenMainsGT2 As New ArrayList

visited_OpenMainsGT2.Add(0)

Session("visited_OpenMainsGT2") = visited_OpenMainsGT2



Dim visited_Missing As New ArrayList

visited_Missing.Add(0)

Session("visited_Missing") = visited_Missing



'reset session variables for the case - transformer with nearby factor GTE10



Dim visited_OpenSwitches_NbyFactorGTE10 As New ArrayList

visited_OpenSwitches_NbyFactorGTE10.Add(0)

Session("visited_OpenSwitches_NbyFactorGTE10") =
visited_OpenSwitches_NbyFactorGTE10



Dim visited_BanksOff_NbyFactorGTE10 As New ArrayList

visited_BanksOff_NbyFactorGTE10.Add(0)
```

Session("visited_BanksOff_NbyFactorGTE10") = visited_BanksOff_NbyFactorGTE10

Dim visited_Fuse_NbyFactorGTE10 As New ArrayList

visited_Fuse_NbyFactorGTE10.Add(0)

Session("visited_Fuse_NbyFactorGTE10") = visited_Fuse_NbyFactorGTE10

Dim visited_ClosedOld_NbyFactorGTE10 As New ArrayList

visited_ClosedOld_NbyFactorGTE10.Add(0)

Session("visited_ClosedOld_NbyFactorGTE10") = visited_ClosedOld_NbyFactorGTE10

Dim visited_ClosedUNR_NbyFactorGTE10 As New ArrayList

visited_ClosedUNR_NbyFactorGTE10.Add(0)

Session("visited_ClosedUNR_NbyFactorGTE10") = visited_ClosedUNR_NbyFactorGTE10

Dim visited_ClosedDNU_NbyFactorGTE10 As New ArrayList

visited_ClosedDNU_NbyFactorGTE10.Add(0)

Session("visited_ClosedDNU_NbyFactorGTE10") = visited_ClosedDNU_NbyFactorGTE10

Dim visited_Missing_NbyFactorGTE10 As New ArrayList

visited_Missing_NbyFactorGTE10.Add(0)

Session("visited_Missing_NbyFactorGTE10") = visited_Missing_NbyFactorGTE10

Dim visited_OpenMainsLTE2_NbyFactorGTE10 As New ArrayList

visited_OpenMainsLTE2_NbyFactorGTE10.Add(0)

Session("visited_OpenMainsLTE2_NbyFactorGTE10") = visited_OpenMainsLTE2_NbyFactorGTE10

Dim visited_OpenMainsGT2_NbyFactorGTE10 As New ArrayList

visited_OpenMainsGT2_NbyFactorGTE10.Add(0)

Session("visited_OpenMainsGT2_NbyFactorGTE10") = visited_OpenMainsGT2_NbyFactorGTE10

'reset session variables for the case - transformer with nearby factor GTE5lt10

```
Dim visited_OpenSwitches_NbyFactorGTE5LT10 As New ArrayList

visited_OpenSwitches_NbyFactorGTE5LT10.Add(0)

Session("visited_OpenSwitches_NbyFactorGTE5LT10") =
visited_OpenSwitches_NbyFactorGTE5LT10


Dim visited_BanksOff_NbyFactorGTE5LT10 As New ArrayList

visited_BanksOff_NbyFactorGTE5LT10.Add(0)

Session("visited_BanksOff_NbyFactorGTE5LT10") =
visited_BanksOff_NbyFactorGTE5LT10


Dim visited_Fuse_NbyFactorGTE5LT10 As New ArrayList

visited_Fuse_NbyFactorGTE5LT10.Add(0)

Session("visited_Fuse_NbyFactorGTE5LT10") = visited_Fuse_NbyFactorGTE5LT10


Dim visited_ClosedOld_NbyFactorGTE5LT10 As New ArrayList

visited_ClosedOld_NbyFactorGTE5LT10.Add(0)

Session("visited_ClosedOld_NbyFactorGTE5LT10") =
visited_ClosedOld_NbyFactorGTE5LT10
```

```
Dim visited_ClosedUNR_NbyFactorGTE5LT10 As New ArrayList

visited_ClosedUNR_NbyFactorGTE5LT10.Add(0)

Session("visited_ClosedUNR_NbyFactorGTE5LT10") =
visited_ClosedUNR_NbyFactorGTE5LT10


Dim visited_ClosedDNU_NbyFactorGTE5LT10 As New ArrayList

visited_ClosedDNU_NbyFactorGTE5LT10.Add(0)

Session("visited_ClosedDNU_NbyFactorGTE5LT10") =
visited_ClosedDNU_NbyFactorGTE5LT10


Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As New ArrayList

visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(0)

Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") =
visited_OpenMainsLTE2_NbyFactorGTE5LT10


Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As New ArrayList

visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(0)

Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") =
visited_OpenMainsGT2_NbyFactorGTE5LT10
```

```
Dim visited_Missing_NbyFactorGTE5LT10 As New ArrayList

visited_Missing_NbyFactorGTE5LT10.Add(0)

Session("visited_Missing_NbyFactorGTE5LT10") =
visited_Missing_NbyFactorGTE5LT10

End Sub

Protected Sub UpdateTargetVaultCounts_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UpdateTargetVaultCounts.Click

update_Target_OpenSwitches()

update_Target_Fuse()

update_Target_BanksOff()

update_Target_ClosedOld()
```

update_Target_ClosedUNR()

update_Target_ClosedDNU()

update_Target_OpenMainsLTE2()

update_Target_OpenMainsGT2()

' Target_Total_LPW_Calc()

update_Target_Missing()

update_Target_OpenSwitches_NbyFactorGTE10()

update_Target_Fuse_NbyFactorGTE10()

update_Target_BanksOff_NbyFactorGTE10()

update_Target_ClosedOld_NbyFactorGTE10()

update_Target_ClosedUNR_NbyFactorGTE10()

update_Target_ClosedDNU_NbyFactorGTE10()

update_Target_OpenMainsLTE2_NbyFactorGTE10()

update_Target_OpenMainsGT2_NbyFactorGTE10()

update_Target_MIssing_NbyFactorGTE10()

update_Target_OpenSwitches_NbyFactorGTE5LT10()

update_Target_Fuse_NbyFactorGTE5LT10()

update_Target_BanksOff_NbyFactorGTE5LT10()

update_Target_ClosedOld_NbyFactorGTE5LT10()

update_Target_ClosedUNR_NbyFactorGTE5LT10()

update_Target_ClosedDNU_NbyFactorGTE5LT10()

update_Target_OpenMainsLTE2_NbyFactorGTE5LT10()

update_Target_OpenMainsGT2_NbyFactorGTE5LT10()

update_Target_Missing_NbyFactorGTE5LT10()

Target_Total_LPW_Calc()

LPW_Cost_Calculation()

Dim MyScript As New System.Text.StringBuilder

With MyScript

```
.Append("<script type='text/javascript'>")
.Append("alert ('update completed');")
.Append("</script>")
End With

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
ScriptManager.RegisterStartupScript(Me, Me.GetType(), "updatesdone",
MyScript.ToString, False)

End Sub
Private Sub reset_lpwtotals()
TotalLPWCurrent.Text = ""
```

```
End Sub

''' <summary>

''' below are the methods to update target vault counts

''' based on checked boxes

''' </summary>

''' <remarks></remarks>

Private Sub update_Target_OpenSwitches()

  GV_OpenSwitches_update()

  Dim visited_OpenSwitches As New ArrayList

  If Session("visited_OpenSwitches") Is Nothing Then Exit Sub



  visited_OpenSwitches = CType(Session("visited_OpenSwitches"), ArrayList)



  Target_OpenSwitchesFeeder.Text = Int32.Parse(OpenSwitchesFeederCurrent.Text) -
(visited_OpenSwitches.Count - 1)

  'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

  TextBox2.Text = visited_OpenSwitches.Count

  'TextBox2.Text = vault_count
```

```
End Sub

Private Sub update_Target_Fuse()
  GV_fuse_update()
  Dim visited_fuse As New ArrayList
  If Session("visited_fuse") Is Nothing Then Exit Sub

  visited_fuse = CType(Session("visited_fuse"), ArrayList)

  Target_FuseFeederCurrent.Text = Int32.Parse(FuseFeederCurrent.Text) -
(visited_fuse.Count - 1)

  Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 *
Int32.Parse(Target_Fuse_GTE10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)
End Sub
```

```
Private Sub update_Target_BanksOff()

  GV_BanksOff_update()

  Dim visited_BanksOff As New ArrayList

  If Session("visited_BanksOff") Is Nothing Then Exit Sub


  visited_BanksOff = CType(Session("visited_BanksOff"), ArrayList)


  Target_BanksOffFeederCurrent.Text = Int32.Parse(BanksOffFeederCurrent.Text) -
(visited_BanksOff.Count - 1)

End Sub

Private Sub update_Target_ClosedOld()

  GV_ClosedOld_update()

  Dim visited_ClosedOld As New ArrayList

  If Session("visited_ClosedOld") Is Nothing Then Exit Sub


  visited_ClosedOld = CType(Session("visited_ClosedOld"), ArrayList)


  Target_ClosedOld.Text = Int32.Parse(ClosedOld.Text) - (visited_ClosedOld.Count - 1)
```

```
End Sub



Private Sub update_Target_ClosedUNR()
  GV_ClosedUNR_update()
  Dim visited_ClosedUNR As New ArrayList
  If Session("visited_ClosedUNR") Is Nothing Then Exit Sub

  visited_ClosedUNR = CType(Session("visited_ClosedUNR"), ArrayList)

  Target_ClosedUNR.Text = Int32.Parse(ClosedUNR.Text) - (visited_ClosedUNR.Count -
1)
End Sub
Private Sub update_Target_ClosedDNU()
  GV_ClosedDNU_update()
  Dim visited_ClosedDNU As New ArrayList
  If Session("visited_ClosedDNU") Is Nothing Then Exit Sub
```

```
visited_ClosedDNU = CType(Session("visited_ClosedDNU"), ArrayList)

Target_ClosedDNUCurrent.Text = Int32.Parse(ClosedDNUCurrent.Text) -
(visited_ClosedDNU.Count - 1)

End Sub

Private Sub update_Target_OpenMainsLTE2()

GV_OpenMainsLTE2_update()

Dim visited_OpenMainsLTE2 As New ArrayList

If Session("visited_OpenMainsLTE2") Is Nothing Then Exit Sub

visited_OpenMainsLTE2 = CType(Session("visited_OpenMainsLTE2"), ArrayList)

Target_OpenMainsLTE2Current.Text = Int32.Parse(OpenMainsLTE2Current.Text) -
(visited_OpenMainsLTE2.Count - 1)

End Sub

Private Sub update_Target_OpenMainsGT2()
```

```
GV_OpenMainsGT2_update()

Dim visited_OpenMainsGT2 As New ArrayList

If Session("visited_OpenMainsGT2") Is Nothing Then Exit Sub


visited_OpenMainsGT2 = CType(Session("visited_OpenMainsGT2"), ArrayList)


Target_OpenMainsGT2Current.Text = Int32.Parse(OpenMainsGT2Current.Text) -
(visited_OpenMainsGT2.Count - 1)

End Sub
```

```
Private Sub update_Target_Missing()

    GV_Missing_update()



    Dim visited_Missing As New ArrayList

    If Session("visited_Missing") Is Nothing Then Exit Sub



    visited_Missing = CType(Session("visited_Missing"), ArrayList)



    Target_MissingFeederCurrent.Text = Int32.Parse(MissingFeederCurrent.Text) -
(visited_Missing.Count - 1)

    'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

    TextBox2.Text = visited_Missing.Count

    'TextBox2.Text = vault_count



End Sub
```

```
Protected Sub GV_OpenSwitches_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_OpenSwitches_NbyFactorGTE10.SelectedIndexChanged

    GV_OpenSwitches_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_openswitches_NbyFactorGTE10 As ArrayList

    If Session("visited_openSwitches_NbyFactorGTE10") Is Nothing Then

      visited_openswitches_NbyFactorGTE10 = New ArrayList

    Else

      visited_openswitches_NbyFactorGTE10 = CType(Session("visited_openswitches_NbyFactorGTE10"), ArrayList)

    End If

    Dim id As String = GV_OpenSwitches_NbyFactorGTE10.SelectedRow.Cells(0).Text

    If visited_openswitches_NbyFactorGTE10.Count < 1 Then visited_openswitches_NbyFactorGTE10.Add(0)

    If visited_openswitches_NbyFactorGTE10.Contains(id) Then

      Exit Sub

    Else
```

visited_openswitches_NbyFactorGTE10.Add(id)

End If

Session("visited_openswitches_NbyFactorGTE10") = visited_openswitches_NbyFactorGTE10

End Sub

Protected Sub GV_BanksOff_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_BanksOff_NbyFactorGTE10.SelectedIndexChanged

GV_BanksOff_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_BanksOff_NbyFactorGTE10 As ArrayList

If Session("visited_BanksOff_NbyFactorGTE10") Is Nothing Then

visited_BanksOff_NbyFactorGTE10 = New ArrayList

Else

```
visited_BanksOff_NbyFactorGTE10 =
CType(Session("visited_BanksOff_NbyFactorGTE10"), ArrayList)

End If

Dim id As String = GV_BanksOff_NbyFactorGTE10.SelectedRow.Cells(0).Text

If visited_BanksOff_NbyFactorGTE10.Count < 1 Then
visited_BanksOff_NbyFactorGTE10.Add(0)

If visited_BanksOff_NbyFactorGTE10.Contains(id) Then

  Exit Sub

Else

  visited_BanksOff_NbyFactorGTE10.Add(id)

End If



Session("visited_BanksOff_NbyFactorGTE10") = visited_BanksOff_NbyFactorGTE10



End Sub
```

```
Protected Sub GV_Fuse_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles GV_Fuse_NbyFactorGTE10.SelectedIndexChanged

    GV_Fuse_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue



    Dim visited_Fuse_NbyFactorGTE10 As ArrayList

    If Session("visited_Fuse_NbyFactorGTE10") Is Nothing Then

      visited_Fuse_NbyFactorGTE10 = New ArrayList

    Else

      visited_Fuse_NbyFactorGTE10  =  CType(Session("visited_Fuse_NbyFactorGTE10"),
ArrayList)

    End If

    Dim id As String = GV_Fuse_NbyFactorGTE10.SelectedRow.Cells(0).Text

    If visited_Fuse_NbyFactorGTE10.Count < 1 Then visited_Fuse_NbyFactorGTE10.Add(0)

    If visited_Fuse_NbyFactorGTE10.Contains(id) Then

      Exit Sub

    Else
```

```
            visited_Fuse_NbyFactorGTE10.Add(id)

        End If


        Session("visited_Fuse_NbyFactorGTE10") = visited_Fuse_NbyFactorGTE10




    End Sub


    Protected Sub GV_ClosedOld_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles GV_ClosedOld_NbyFactorGTE10.SelectedIndexChanged

        GV_ClosedOld_NbyFactorGTE10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

        Dim visited_ClosedOld_NbyFactorGTE10 As ArrayList

        If Session("visited_ClosedOld_NbyFactorGTE10") Is Nothing Then

            visited_ClosedOld_NbyFactorGTE10 = New ArrayList

        Else
```

```
                visited_ClosedOld_NbyFactorGTE10 =
CType(Session("visited_ClosedOld_NbyFactorGTE10"), ArrayList)

                End If

                Dim id As String = GV_ClosedOld_NbyFactorGTE10.SelectedRow.Cells(0).Text

                If visited_ClosedOld_NbyFactorGTE10.Count < 1 Then
visited_ClosedOld_NbyFactorGTE10.Add(0)

                If visited_ClosedOld_NbyFactorGTE10.Contains(id) Then

                  Exit Sub

                Else

                  visited_ClosedOld_NbyFactorGTE10.Add(id)

                End If



                Session("visited_ClosedOld_NbyFactorGTE10") = visited_ClosedOld_NbyFactorGTE10



            End Sub
```

```
Protected Sub GV_ClosedUNR_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_ClosedUNR_NbyFactorGTE10.SelectedIndexChanged

        GV_ClosedUNR_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

        Dim visited_ClosedUNR_NbyFactorGTE10 As ArrayList

        If Session("visited_ClosedUNR_NbyFactorGTE10") Is Nothing Then

          visited_ClosedUNR_NbyFactorGTE10 = New ArrayList

        Else

          visited_ClosedUNR_NbyFactorGTE10 = CType(Session("visited_ClosedUNR_NbyFactorGTE10"), ArrayList)

        End If

        Dim id As String = GV_ClosedUNR_NbyFactorGTE10.SelectedRow.Cells(0).Text

        If visited_ClosedUNR_NbyFactorGTE10.Count < 1 Then visited_ClosedUNR_NbyFactorGTE10.Add(0)

        If visited_ClosedUNR_NbyFactorGTE10.Contains(id) Then

          Exit Sub

        Else
```

visited_ClosedUNR_NbyFactorGTE10.Add(id)

End If

Session("visited_ClosedUNR_NbyFactorGTE10") = visited_ClosedUNR_NbyFactorGTE10

End Sub

Protected Sub GV_ClosedDNU_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_ClosedDNU_NbyFactorGTE10.SelectedIndexChanged

GV_ClosedDNU_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_ClosedDNU_NbyFactorGTE10 As ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE10") Is Nothing Then

visited_ClosedDNU_NbyFactorGTE10 = New ArrayList

Else

visited_ClosedDNU_NbyFactorGTE10 = CType(Session("visited_ClosedDNU_NbyFactorGTE10"), ArrayList)

```
End If

Dim id As String = GV_ClosedDNU_NbyFactorGTE10.SelectedRow.Cells(0).Text

If visited_ClosedDNU_NbyFactorGTE10.Count < 1 Then
visited_ClosedDNU_NbyFactorGTE10.Add(0)

If visited_ClosedDNU_NbyFactorGTE10.Contains(id) Then

  Exit Sub

Else

  visited_ClosedDNU_NbyFactorGTE10.Add(id)

End If


Session("visited_ClosedDNU_NbyFactorGTE10") =
visited_ClosedDNU_NbyFactorGTE10


End Sub


Protected Sub GV_OpenMainsLTE2_NbyFactorGTE10_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs) Handles
GV_OpenMainsLTE2_NbyFactorGTE10.SelectedIndexChanged
```

```
GV_OpenMainsLTE2_NbyFactorGTE10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_OpenMainsLTE2_NbyFactorGTE10 As ArrayList

If Session("visited_OpenMainsLTE2_NbyFactorGTE10") Is Nothing Then

  visited_OpenMainsLTE2_NbyFactorGTE10 = New ArrayList

Else

  visited_OpenMainsLTE2_NbyFactorGTE10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE10"), ArrayList)

End If

Dim id As String = GV_OpenMainsLTE2_NbyFactorGTE10.SelectedRow.Cells(0).Text

If visited_OpenMainsLTE2_NbyFactorGTE10.Count < 1 Then
visited_OpenMainsLTE2_NbyFactorGTE10.Add(0)

If visited_OpenMainsLTE2_NbyFactorGTE10.Contains(id) Then

  Exit Sub

Else

  visited_OpenMainsLTE2_NbyFactorGTE10.Add(id)

End If
```

Session("visited_OpenMainsLTE2_NbyFactorGTE10") = visited_OpenMainsLTE2_NbyFactorGTE10

End Sub

Protected Sub GV_OpenMainsGT2_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_OpenMainsGT2_NbyFactorGTE10.SelectedIndexChanged

GV_OpenMainsGT2_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_OpenMainsGT2_NbyFactorGTE10 As ArrayList

If Session("visited_OpenMainsGT2_NbyFactorGTE10") Is Nothing Then

visited_OpenMainsGT2_NbyFactorGTE10 = New ArrayList

Else

visited_OpenMainsGT2_NbyFactorGTE10 = CType(Session("visited_OpenMainsGT2_NbyFactorGTE10"), ArrayList)

End If

Dim id As String = GV_OpenMainsGT2_NbyFactorGTE10.SelectedRow.Cells(0).Text

If visited_OpenMainsGT2_NbyFactorGTE10.Count < 1 Then visited_OpenMainsGT2_NbyFactorGTE10.Add(0)

If visited_OpenMainsGT2_NbyFactorGTE10.Contains(id) Then

```
        Exit Sub

      Else

        visited_OpenMainsGT2_NbyFactorGTE10.Add(id)

      End If



      Session("visited_OpenMainsGT2_NbyFactorGTE10")                    =
visited_OpenMainsGT2_NbyFactorGTE10



    End Sub



    'code to update values in GTE10 columns

    Private Sub update_Target_OpenSwitches_NbyFactorGTE10()

      GV_OpenSwitches_NbyFactorGTE10_update()

      Dim visited_OpenSwitches_NbyFactorGTE10 As New ArrayList
```

```
If Session("visited_OpenSwitches_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_OpenSwitches_NbyFactorGTE10 = CType(Session("visited_OpenSwitches_NbyFactorGTE10"), ArrayList)

Target_OpenSwitches_GTE10.Text = Int32.Parse(OpenSwitches_GTE10.Text) - (visited_OpenSwitches_NbyFactorGTE10.Count - 1)

'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

'TextBox2.Text = visited_OpenSwitches_NbyFactorGTE10.Count

'TextBox2.Text = vault_count

End Sub

Private Sub update_Target_Fuse_NbyFactorGTE10()

GV_fuse_NbyFactorGTE10_update()

Dim visited_fuse_NbyFactorGTE10 As New ArrayList

If Session("visited_fuse_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_fuse_NbyFactorGTE10 = CType(Session("visited_fuse_NbyFactorGTE10"), ArrayList)
```

```
                Target_Fuse_GTE10.Text = Int32.Parse(Fuse_GTE10.Text) -
(visited_fuse_NbyFactorGTE10.Count - 1)

                Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 *
Int32.Parse(Target_Fuse_GTE10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)

            End Sub

            Private Sub update_Target_BanksOff_NbyFactorGTE10()

              GV_BanksOff_NbyFactorGTE10_update()

              Dim visited_BanksOff_NbyFactorGTE10 As New ArrayList

              If Session("visited_BanksOff_NbyFactorGTE10") Is Nothing Then Exit Sub

              visited_BanksOff_NbyFactorGTE10 =
CType(Session("visited_BanksOff_NbyFactorGTE10"), ArrayList)

              Target_BanksOff_GTE10.Text = Int32.Parse(BanksOff_GTE10.Text) -
(visited_BanksOff_NbyFactorGTE10.Count - 1)

            End Sub

            Private Sub update_Target_ClosedOld_NbyFactorGTE10()
```

```
GV_ClosedOld_NbyFactorGTE10_update()

Dim visited_ClosedOld_NbyFactorGTE10 As New ArrayList

If Session("visited_ClosedOld_NbyFactorGTE10") Is Nothing Then Exit Sub


visited_ClosedOld_NbyFactorGTE10 =
CType(Session("visited_ClosedOld_NbyFactorGTE10"), ArrayList)


Target_ClosedOld_GTE10.Text = Int32.Parse(ClosedOld_GTE10.Text) -
(visited_ClosedOld_NbyFactorGTE10.Count - 1)


End Sub


Private Sub update_Target_ClosedUNR_NbyFactorGTE10()

GV_ClosedUNR_NbyFactorGTE10_update()

Dim visited_ClosedUNR_NbyFactorGTE10 As New ArrayList

If Session("visited_ClosedUNR_NbyFactorGTE10") Is Nothing Then Exit Sub
```

```
visited_ClosedUNR_NbyFactorGTE10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE10"), ArrayList)

Target_ClosedUNR_GTE10.Text = Int32.Parse(ClosedUNR_GTE10.Text) -
(visited_ClosedUNR_NbyFactorGTE10.Count - 1)

End Sub

Private Sub update_Target_ClosedDNU_NbyFactorGTE10()

GV_ClosedDNU_NbyFactorGTE10_update()

Dim visited_ClosedDNU_NbyFactorGTE10 As New ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_ClosedDNU_NbyFactorGTE10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE10"), ArrayList)

Target_ClosedDNU_GTE10.Text = Int32.Parse(ClosedDNU_GTE10.Text) -
(visited_ClosedDNU_NbyFactorGTE10.Count - 1)

End Sub

Private Sub update_Target_Missing_NbyFactorGTE10()
```

```
GV_Missing_NbyFactorGTE10_update()

Dim visited_Missing_NbyFactorGTE10 As New ArrayList

If Session("visited_Missing_NbyFactorGTE10") Is Nothing Then Exit Sub


visited_Missing_NbyFactorGTE10 =
CType(Session("visited_Missing_NbyFactorGTE10"), ArrayList)


Target_Missing_GTE10.Text = Int32.Parse(ClosedDNU_GTE10.Text) -
(visited_Missing_NbyFactorGTE10.Count - 1)

End Sub


Private Sub update_Target_OpenMainsLTE2_NbyFactorGTE10()

GV_OpenMainsLTE2_NbyFactorGTE10_update()


Dim visited_OpenMainsLTE2_NbyFactorGTE10 As New ArrayList

If Session("visited_OpenMainsLTE2_NbyFactorGTE10") Is Nothing Then Exit Sub
```

visited_OpenMainsLTE2_NbyFactorGTE10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE10"), ArrayList)

Target_OMLTE2_GTE10.Text = Int32.Parse(OMLTE2_GTE10.Text) -
(visited_OpenMainsLTE2_NbyFactorGTE10.Count - 1)

End Sub

Private Sub update_Target_OpenMainsGT2_NbyFactorGTE10()

GV_OpenMainsGT2_NbyFactorGTE10_update()

Dim visited_OpenMainsGT2_NbyFactorGTE10 As New ArrayList

If Session("visited_OpenMainsGT2_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_OpenMainsGT2_NbyFactorGTE10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE10"), ArrayList)

Target_OMGT2_GTE10.Text = Int32.Parse(OMGT2_GTE10.Text) -
(visited_OpenMainsGT2_NbyFactorGTE10.Count - 1)

```
End Sub


'code to update values in GTE5LT10 columns
Private Sub update_Target_OpenSwitches_NbyFactorGTE5LT10()
    GV_OpenSwitches_NbyFactorGTE5LT10_update()
    Dim visited_OpenSwitches_NbyFactorGTE5LT10 As New ArrayList
    If Session("visited_OpenSwitches_NbyFactorGTE5LT10") Is Nothing Then Exit Sub
```

```
visited_OpenSwitches_NbyFactorGTE5LT10 =
CType(Session("visited_OpenSwitches_NbyFactorGTE5LT10"), ArrayList)

Target_OpenSwitches_GTE5LT10.Text = Int32.Parse(OpenSwitches_GTE5LT10.Text) -
(visited_OpenSwitches_NbyFactorGTE5LT10.Count - 1)

'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

TextBox2.Text = visited_OpenSwitches_NbyFactorGTE5LT10.Count

'TextBox2.Text = vault_count


End Sub


Private Sub update_Target_Fuse_NbyFactorGTE5LT10()
  GV_fuse_NbyFactorGTE5LT10_update()
  Dim visited_fuse_NbyFactorGTE5LT10 As New ArrayList
  If Session("visited_fuse_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_fuse_NbyFactorGTE5LT10 =
CType(Session("visited_fuse_NbyFactorGTE5LT10"), ArrayList)
```

```
                Target_Fuse_GTE5LT10.Text = Int32.Parse(Fuse_GTE5LT10.Text) -
(visited_fuse_NbyFactorGTE5LT10.Count - 1)

                Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 *
Int32.Parse(Target_Fuse_GTE5LT10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)

                End Sub

                Private Sub update_Target_BanksOff_NbyFactorGTE5LT10()

                  GV_BanksOff_NbyFactorGTE5LT10_update()

                  Dim visited_BanksOff_NbyFactorGTE5LT10 As New ArrayList

                  If Session("visited_BanksOff_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

                  visited_BanksOff_NbyFactorGTE5LT10 =
CType(Session("visited_BanksOff_NbyFactorGTE5LT10"), ArrayList)

                  Target_BanksOff_GTE5LT10.Text = Int32.Parse(BanksOff_GTE5LT10.Text) -
(visited_BanksOff_NbyFactorGTE5LT10.Count - 1)

                End Sub

                Private Sub update_Target_ClosedOld_NbyFactorGTE5LT10()
```

```
GV_ClosedOld_NbyFactorGTE5LT10_update()

Dim visited_ClosedOld_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_ClosedOld_NbyFactorGTE5LT10") Is Nothing Then Exit Sub


visited_ClosedOld_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedOld_NbyFactorGTE5LT10"), ArrayList)


Target_ClosedOld_GTE5LT10.Text = Int32.Parse(ClosedOld_GTE5LT10.Text) -
(visited_ClosedOld_NbyFactorGTE5LT10.Count - 1)


End Sub



Private Sub update_Target_ClosedUNR_NbyFactorGTE5LT10()

GV_ClosedUNR_NbyFactorGTE5LT10_update()

Dim visited_ClosedUNR_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_ClosedUNR_NbyFactorGTE5LT10") Is Nothing Then Exit Sub
```

```
visited_ClosedUNR_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE5LT10"), ArrayList)

Target_ClosedUNR_GTE5LT10.Text = Int32.Parse(ClosedUNR_GTE5LT10.Text) -
(visited_ClosedUNR_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_ClosedDNU_NbyFactorGTE5LT10()

GV_ClosedDNU_NbyFactorGTE5LT10_update()

Dim visited_ClosedDNU_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_ClosedDNU_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE5LT10"), ArrayList)

Target_ClosedDNU_GTE5LT10.Text = Int32.Parse(ClosedDNU_GTE5LT10.Text) -
(visited_ClosedDNU_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_OpenMainsLTE2_NbyFactorGTE5LT10()

GV_OpenMainsLTE2_NbyFactorGTE5LT10_update()
```

```
Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_OpenMainsLTE2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10"), ArrayList)

Target_OMLTE2_GTE5LT10.Text = Int32.Parse(OMLTE2_GTE5LT10.Text) -
(visited_OpenMainsLTE2_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_OpenMainsGT2_NbyFactorGTE5LT10()

GV_OpenMainsGT2_NbyFactorGTE5LT10_update()

Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_OpenMainsGT2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE5LT10"), ArrayList)
```

```
Target_OMGT2_GTE5LT10.Text = Int32.Parse(OMGT2_GTE5LT10.Text) -
(visited_OpenMainsGT2_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_missing_NbyFactorGTE5LT10()

  GV_missing_NbyFactorGTE5LT10_update()

  Dim visited_missing_NbyFactorGTE5LT10 As New ArrayList

  If Session("visited_missing_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

  visited_missing_NbyFactorGTE5LT10 =
CType(Session("visited_missing_NbyFactorGTE5LT10"), ArrayList)

  Target_Missing_GTE5LT10.Text = Int32.Parse(Missing_GTE5LT10.Text) -
(visited_missing_NbyFactorGTE5LT10.Count - 1)

  'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

  TextBox2.Text = visited_missing_NbyFactorGTE5LT10.Count

  'TextBox2.Text = vault_count
```

```
End Sub

Protected Sub GV_OpenSwitches_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs)

GV_OpenSwitches_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_openswitches_NbyFactorGTE5LT10 As ArrayList

If Session("visited_openSwitches_NbyFactorGTE5LT10") Is Nothing Then

  visited_openswitches_NbyFactorGTE5LT10 = New ArrayList

Else

  visited_openswitches_NbyFactorGTE5LT10 =
CType(Session("visited_openswitches_NbyFactorGTE5LT10"), ArrayList)

End If

Dim id As String = GV_OpenSwitches_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_openswitches_NbyFactorGTE5LT10.Count < 1 Then
visited_openswitches_NbyFactorGTE5LT10.Add(0)

If visited_openswitches_NbyFactorGTE5LT10.Contains(id) Then
```

```
            Exit Sub
        Else
            visited_openswitches_NbyFactorGTE5LT10.Add(id)
        End If

        Session("visited_openswitches_NbyFactorGTE5LT10") =
visited_openswitches_NbyFactorGTE5LT10

    End Sub

    Protected Sub GV_BanksOff_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender
As Object, ByVal e As System.EventArgs)
        GV_BanksOff_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue
        Dim visited_BanksOff_NbyFactorGTE5LT10 As ArrayList
        If Session("visited_BanksOff_NbyFactorGTE5LT10") Is Nothing Then
            visited_BanksOff_NbyFactorGTE5LT10 = New ArrayList
        Else
```

```
visited_BanksOff_NbyFactorGTE5LT10 =
CType(Session("visited_BanksOff_NbyFactorGTE5LT10"), ArrayList)

End If

Dim id As String = GV_BanksOff_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_BanksOff_NbyFactorGTE5LT10.Count < 1 Then
visited_BanksOff_NbyFactorGTE5LT10.Add(0)

If visited_BanksOff_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else

  visited_BanksOff_NbyFactorGTE5LT10.Add(id)

End If



Session("visited_BanksOff_NbyFactorGTE5LT10") =
visited_BanksOff_NbyFactorGTE5LT10




End Sub
```

```
Protected Sub GV_Fuse_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)

GV_Fuse_NbyFactorGTE5LT10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue



Dim visited_Fuse_NbyFactorGTE5LT10 As ArrayList

If Session("visited_Fuse_NbyFactorGTE5LT10") Is Nothing Then

  visited_Fuse_NbyFactorGTE5LT10 = New ArrayList

Else

  visited_Fuse_NbyFactorGTE5LT10 = CType(Session("visited_Fuse_NbyFactorGTE5LT10"), ArrayList)

End If

Dim id As String = GV_Fuse_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_Fuse_NbyFactorGTE5LT10.Count < 1 Then visited_Fuse_NbyFactorGTE5LT10.Add(0)

If visited_Fuse_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else

  visited_Fuse_NbyFactorGTE5LT10.Add(id)
```

```
End If

Session("visited_Fuse_NbyFactorGTE5LT10") = visited_Fuse_NbyFactorGTE5LT10

End Sub

Protected Sub GV_ClosedOld_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender
As Object, ByVal e As System.EventArgs)

GV_ClosedOld_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_ClosedOld_NbyFactorGTE5LT10 As ArrayList

If Session("visited_ClosedOld_NbyFactorGTE5LT10") Is Nothing Then

  visited_ClosedOld_NbyFactorGTE5LT10 = New ArrayList

Else

  visited_ClosedOld_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedOld_NbyFactorGTE5LT10"), ArrayList)

End If
```

```
Dim id As String = GV_ClosedOld_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_ClosedOld_NbyFactorGTE5LT10.Count < 1 Then visited_ClosedOld_NbyFactorGTE5LT10.Add(0)

If visited_ClosedOld_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else

  visited_ClosedOld_NbyFactorGTE5LT10.Add(id)

End If


Session("visited_ClosedOld_NbyFactorGTE5LT10") = visited_ClosedOld_NbyFactorGTE5LT10



End Sub


Protected Sub GV_ClosedUNR_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)

GV_ClosedUNR_NbyFactorGTE5LT10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
```

```
Dim visited_ClosedUNR_NbyFactorGTE5LT10 As ArrayList
If Session("visited_ClosedUNR_NbyFactorGTE5LT10") Is Nothing Then
  visited_ClosedUNR_NbyFactorGTE5LT10 = New ArrayList
Else
  visited_ClosedUNR_NbyFactorGTE5LT10 = CType(Session("visited_ClosedUNR_NbyFactorGTE5LT10"), ArrayList)
End If
Dim id As String = GV_ClosedUNR_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text
If visited_ClosedUNR_NbyFactorGTE5LT10.Count < 1 Then visited_ClosedUNR_NbyFactorGTE5LT10.Add(0)
If visited_ClosedUNR_NbyFactorGTE5LT10.Contains(id) Then
  Exit Sub
Else
  visited_ClosedUNR_NbyFactorGTE5LT10.Add(id)
End If

Session("visited_ClosedUNR_NbyFactorGTE5LT10") = visited_ClosedUNR_NbyFactorGTE5LT10
```

```
End Sub

Protected Sub GV_ClosedDNU_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs)

GV_ClosedDNU_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_ClosedDNU_NbyFactorGTE5LT10 As ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE5LT10") Is Nothing Then

  visited_ClosedDNU_NbyFactorGTE5LT10 = New ArrayList

Else

  visited_ClosedDNU_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE5LT10"), ArrayList)

End If

Dim id As String = GV_ClosedDNU_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_ClosedDNU_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedDNU_NbyFactorGTE5LT10.Add(0)

If visited_ClosedDNU_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else
```

```
            visited_ClosedDNU_NbyFactorGTE5LT10.Add(id)

        End If


        Session("visited_ClosedDNU_NbyFactorGTE5LT10") =
visited_ClosedDNU_NbyFactorGTE5LT10


    End Sub


    Protected Sub GV_OpenMainsLTE2_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs)

        GV_OpenMainsLTE2_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

        Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As ArrayList

        If Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") Is Nothing Then

            visited_OpenMainsLTE2_NbyFactorGTE5LT10 = New ArrayList

        Else

            visited_OpenMainsLTE2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10"), ArrayList)

        End If
```

```
Dim id As String = GV_OpenMainsLTE2_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Count < 1 Then visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(0)

If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else

  visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(id)

End If



Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") = visited_OpenMainsLTE2_NbyFactorGTE5LT10



End Sub



Protected Sub GV_OpenMainsGT2_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
```

```
GV_OpenMainsGT2_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As ArrayList
If Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") Is Nothing Then
  visited_OpenMainsGT2_NbyFactorGTE5LT10 = New ArrayList
Else
  visited_OpenMainsGT2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE5LT10"), ArrayList)
End If
Dim id As String =
GV_OpenMainsGT2_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text
If visited_OpenMainsGT2_NbyFactorGTE5LT10.Count < 1 Then
visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(0)
If visited_OpenMainsGT2_NbyFactorGTE5LT10.Contains(id) Then
  Exit Sub
Else
  visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(id)
End If
```

```
Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") =
visited_OpenMainsGT2_NbyFactorGTE5LT10

End Sub

Private Sub IR_Scandatetime()
  Dim feeder As String
  feeder = cmbSearchFeeder.SelectedValue.ToString
  'If feeder.Length < 5 Then
  '   feeder = "0" + feeder.ToString
  'End If

  'Dim sql As String = "SELECT * FROM LoadPocket.dbo.latest_lpw WHERE Feeder = '"
& cmbSearchFeeder.SelectedValue & "'"
  'Dim sql As String = "SELECT * FROM
[Transformer_OM_BO].[dbo].[loadpocketweight_test] WHERE Feeder ='" + feeder + "' And
```

IR_nwk_scan_datetime = (select max(IR_nwk_scan_datetime) from [Transformer_OM_BO].[dbo].[loadpocketweight_test] where Feeder ='" + feeder + "')"

Dim sql As String = "SELECT * FROM LoadPocket.dbo.latest_lpw WHERE Feeder ='" + feeder + "' And nwk_scan_datetime = (select max(nwk_scan_datetime) from LoadPocket.dbo.latest_lpw where Feeder ='" + feeder + "')"

'Dim sql As String = "SELECT * FROM rerank_dynamic_Attrib WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("DB2FeederSuscept").ConnectionString)

'wDim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

```
If dt.Rows.Count = 0 Then

   Exit Sub

End If




Dim dr As DataRow = dt.Rows(0)

'getting the rank from the precomputed table now

'RankBaseLoad.Text = drcheck(dr, "Rank")

IR_ScanTime.Text = dr("nwk_scan_datetime")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing



End Sub



Protected Sub GVPILCRuns_Select_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles GVPILCRuns_Select.SelectedIndexChanged
```

```
Dim dt As DataTable

Dim Run As Integer

Dim feeder As String

feeder = cmbSearchFeeder.SelectedValue

Run = Int32.Parse(GVPILCRuns_Select.SelectedItem.Cells(2).Text)



dt = GetSectionInRunTable(Run, feeder)



Dim struct As String

struct = dt.Rows(0).Item("from_struct")

Session("struct") = struct



Dim URL As String



URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"
```

```
Dim MyScript1 As New System.Text.StringBuilder

With MyScript1

  .Append("<script type='text/javascript'>")

  .Append("window.open('" & URL & "');")

  .Append("</script>")

End With



'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript1.ToString, False)



'RegisterStartupScript(Me, Me.GetType(), "FeederMap1", MyScript1.ToString, False)



'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text
```

```
        Dim m_deltaMTBF As String

        m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

        'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



        If Equals(m_deltaMTBF, "no") Then

           GVPILCRuns_Select.Columns(7).Visible = False

        End If



    End Sub



        Protected Sub GVPILCRuns_Select_SortCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridSortCommandEventArgs) Handles GVPILCRuns_Select.SortCommand



        Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder.Text))

        Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))



        ' dv1.Sort = e.SortExpression + " " + "desc"
```

```
dv1.Sort = e.SortExpression

If ViewState("sortdirection").ToString.Equals("ASC") Then

   dv1.Sort = e.SortExpression + " " + "desc"

   ViewState("sortdirection") = "DESC"

Else

   dv1.Sort = e.SortExpression + " " + "ASC"

   ViewState("sortdirection") = "ASC"

End If



Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PICLSections")

dt1 = dv1.ToTable()

dt2 = dv.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn
```

```
'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

GVPILCRuns_Select.DataSource = dss

GVPILCRuns_Select.DataMember = dss.Tables(0).TableName()

GVPILCRuns_Select.DataBind()

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1

  If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVPILCRuns_Select.Items(visited_run_grid).Cells(2).Text),  Nothing))
Then

    GVPILCRuns_Select.Items(visited_run_grid).ForeColor = Drawing.Color.Red

  End If
```

```
        Next

    End Sub

    Protected Sub GVPILCRuns_Select_TemplateSelection(ByVal sender As Object, ByVal e
As DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVPILCRuns_Select.TemplateSelection

        e.TemplateFilename = "Templates\\PILCSections.ascx"

    End Sub

End Class
```

End Namespace

APPENDIX B

```
Imports System.Data

Imports System.Data.SqlClient

'Imports Modules

Imports WebserviceClient

Imports UpdateAttributes



Namespace EdisonProgram

  Partial Class CapitalPlanning_Rerank

    Inherits System.Web.UI.Page



#Region " Page Setup "

    Dim VisitedGlbStructures As New ArrayList

    Dim VisitedGlbJoints As New ArrayList

    Dim Duplicate_sections_list As Integer

    Dim Visited_sections_count As Integer

    Shared Uattr As UpdateAttributes.UpdateAttributes
```

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load

    If Not Page.IsPostBack Then

        Uattr = New UpdateAttributes.UpdateAttributes

        SetupPage()

    End If

End Sub

Private Sub SetupPage()

    BindSearchNetworkList()

    BindSearchFeederList()

    CAPTWizard.ActiveStepIndex = 0
```

```
    SetupFields()



End Sub

Private Sub SetupFields()

    'ovsec.Visible = True

    'ovruns.Visible = True



    'Runs.Visible = True

    'for now the quury runs too slow....

    'XLP.Visible = False



    'vs2220: added clearing logic for globals



    visitedGlbStructures.Clear()

    visitedGlbJoints.Clear()

    SectionsSummary.Items.Clear()

    VisitedGlbRuns.Items.Clear()

    Session("strategies_load") = "0"
```

```
Session("strategies_pilc") = "0"



Dim visited_visionIds As New ArrayList

visited_visionIds.Add(0)

Session("visited_sections_nonPILC") = visited_visionIds



Dim visited_openswitches As New ArrayList

visited_openswitches.Add(0)

Session("visited_openswitches") = visited_openswitches



ViewState("sortdirection") = ""







'visitedGlbRuns.Clear()

duplicate_sections_list = 0

visited_sections_count = 0
```

"added by shringika on 07-30-2008 to clear up summary fields

TotalCostSum.Text = ""

CostPerSectionSum.Text = ""

SectionsChangedSum.Text = ""

sections_area.InnerText = ""

BindAttributes()

BindRankingLPW()

BindFOT()

Reset_New_Attributes()

'Assumes replace PILC is strategy of choice

'Replace_Paper()

'Calculate_New_Load()

Reset_MTBFs()

Initial_Costs()

```
'modification by shringika on 03/11/2008 for LPW tier

'BindAttributes1()

'modification ends here 03/11/2008

'ovsec.Visible = False

'ovruns.Visible = False



'Runs.Visible = False

'XLP.Visible = False




VaultDetailsPanel.Visible = False

Dim feeder_id As String = feeder_withzeroes_fix(cmbSearchFeeder.SelectedValue)

Uattr.ChangeFeeder(feeder_id)



End Sub
```

```
Private Sub SectionsSummarySetup()

    'ovsec.Visible = True

    'ovruns.Visible = True



    'Runs.Visible = True

    'for now the quury runs too slow....

    'XLP.Visible = False



    'vs2220: added clearing logic for globals

    visitedGlbStructures.Clear()

    visitedGlbJoints.Clear()

    SectionsSummary.Items.Clear()

    sections_area.InnerText = ""



    SectionsChangedSum.Text = ""

    CostPerSectionSum.Text = ""

    TotalCostSum.Text = ""
```

```
        duplicate_sections_list = 0

        visited_sections_count = 0



        VisitedGlbRuns.Items.Clear()

        Session("strategies_load") = "0"

        Session("strategies_pilc") = "0"



        Dim visited_visionIds As New ArrayList

        visited_visionIds.Add(0)

        Session("visited_sections_nonPILC") = visited_visionIds



        Dim visited_openswitches As New ArrayList

        visited_openswitches.Add(0)

        Session("visited_openswitches") = visited_openswitches



        reset_visitedLPWSessionVariables()

        VaultDetailsPanel.Visible = False

    End Sub
```

```
#End Region


#Region " Search "

    Protected Sub cmbSearchBorough_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbSearchBorough.SelectedIndexChanged

        '--added by shringika on 20-dec-2007.

        cmbFeederType.SelectedIndex = 0

        cmbPercentage.SelectedIndex = 3

        cmbPercentage.Enabled = False

        '--*modification ends here shringika

        BindSearchNetworkList()

        BindSearchFeederList()

    End Sub


    Protected Sub cmbSearchNetwork_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles cmbSearchNetwork.SelectedIndexChanged

        BindSearchFeederList()

    End Sub
```

```
Private Sub BindSearchNetworkList()

    cmbSearchNetwork.DataSource = GetNetworkTable(cmbSearchBorough.SelectedValue)

    cmbSearchNetwork.DataValueField = "Network_ID"

    'cmbSearchNetwork.DataValueField = "Network"

    cmbSearchNetwork.DataTextField = "Network_Name"

    'cmbSearchNetwork.DataTextField = "Network"

    cmbSearchNetwork.DataBind()



    '--added by Shringika on 07-Jan-2008 for including "All Networks" option for feeder
selection

    cmbSearchNetwork.Items.Insert(0, New ListItem("All Networks", "All"))

    If cmbSearchBorough.SelectedIndex <> 0 Then cmbSearchNetwork.SelectedIndex = 1

    '--*modification by Shringika on 07-jan-2008 ends here



End Sub
```

```
Private Sub BindSearchFeederList()

    '--modified by shringika to check if overloading percentage selection returns no records on
dec 20 '2007

    '--modified on 2 january 2008 to include wizard start disable/enable function

    'clear section summary

    SectionsSummarySetup()

    CAPTWizard.DisplaySideBar = True

    Dim btn As Button =
TryCast(CAPTWizard.FindControl("StartNavigationTemplateContainerID").FindControl("StartNextButton"),
Button)

    btn.Enabled = True


    Dim dl As DataList =
TryCast(CAPTWizard.FindControl("SideBarContainer").FindControl("SideBarList"), DataList)


    dl.Enabled = True

    msglabel.Text = ""

    '--*modification on 2nd jan ends
```

```
If cmbFeederType.SelectedValue.Contains("All") Then

    cmbSearchFeeder.DataSource = GetFeederTable(cmbSearchNetwork.SelectedValue)
    cmbSearchFeeder.DataValueField = "feeder"
    cmbSearchFeeder.DataTextField = "feeder"
    cmbSearchFeeder.DataBind()
    cmbSearchFeeder.SelectedIndex = 0
    SetupFields()
    Exit Sub

End If

Dim sql1 As String
```

```
Dim myConnection1 As New SqlConnection

myConnection1 = New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

' sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "'"

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "'"

If cmbFeederType.SelectedValue.Contains("Load") Then


Dim pct2 As Integer


pct2 = Int32.Parse(cmbPercentage.SelectedValue)


'--modified by shringika on 7-jan-2008 to include all netwrok case

If cmbSearchNetwork.SelectedIndex <> 0 Then
```

' sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' AND percentage >= " & pct2 & " "

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' AND percentage >= " & pct2 & " "

Else

If cmbSearchBorough.SelectedIndex <> 0 And cmbSearchNetwork.SelectedValue.Contains("All") Then

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' AND percentage >= " & pct2 & " ORDER BY feeder "

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' AND percentage >= " & pct2 & " ORDER BY feeder "

Else

'all boroughs all n/w case

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND percentage >= " & pct2 & " ORDER BY feeder"

sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND percentage >= " & pct2 & " ORDER BY feeder"

End If

End If

End If

If cmbFeederType.SelectedIndex = 2 Then

'myConnection1 = New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim pct2 As Integer

```
pct2 = Int32.Parse(cmbPercentage.SelectedValue)

'--modified by shringika on 7-jan-2008 to include all netwrok case

If cmbSearchNetwork.SelectedIndex <> 0 Then


sql1 = "SELECT [feeder]=(case when (left([EdisonML].[dbo].[Feeders_With_PILC].[feeder],1)=0) then right([EdisonML].[dbo].[Feeders_With_PILC].[feeder],4) else [EdisonML].[dbo].[Feeders_With_PILC].[feeder] end)"

sql1 = sql1 + " FROM [EdisonML].[dbo].[Feeders_With_PILC] INNER JOIN [EdisonML].[dbo].[Feeder_run_length]"

sql1 = sql1 + " ON [EdisonML].[dbo].[Feeders_With_PILC].[feeder]=[EdisonML].[dbo].[Feeder_run_length].[feeder]"

sql1 = sql1 + " WHERE network='" & cmbSearchNetwork.SelectedValue & "' "

sql1 = sql1 + " AND (([feeder_pilcruns_length] *100/[feeder_cable_length]) >= " & pct2 & ") "

'sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) "

sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) "
```

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Feeders_With_PILC WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "' AND (pct_PILC >= " & pct2 & ") "

Else

If cmbSearchBorough.SelectedIndex <> 0 And cmbSearchNetwork.SelectedValue.Contains("All") Then

sql1 = "SELECT [feeder]=(case when (left([EdisonML].[dbo].[Feeders_With_PILC].[feeder],1)=0) then right([EdisonML].[dbo].[Feeders_With_PILC].[feeder],4) else [EdisonML].[dbo].[Feeders_With_PILC].[feeder] end)"

sql1 = sql1 + " FROM [EdisonML].[dbo].[Feeders_With_PILC] INNER JOIN [EdisonML].[dbo].[Feeder_run_length]"

sql1 = sql1 + " ON [EdisonML].[dbo].[Feeders_With_PILC].[feeder]=[EdisonML].[dbo].[Feeder_run_length].[feeder]"

sql1 = sql1 + " WHERE borough='" & cmbSearchBorough.SelectedValue & "' "

sql1 = sql1 + " AND (([feeder_pilcruns_length] *100/[feeder_cable_length]) >= " & pct2 & ") "

"sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT zfeeder from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) "

sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_Canonicalfeeder WHERE iscanonical=1) "

'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Feeders_With_PILC WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' AND (pct_PILC >= " & pct2 & ") ORDER BY feeder "

Else

'all boroughs all n/w case

sql1 = "SELECT [feeder]=(case when (left([EdisonML].[dbo].[Feeders_With_PILC].[feeder],1)=0) then right([EdisonML].[dbo].[Feeders_With_PILC].[feeder],4) else [EdisonML].[dbo].[Feeders_With_PILC].[feeder] end)"

sql1 = sql1 + " FROM [EdisonML].[dbo].[Feeders_With_PILC] INNER JOIN [EdisonML].[dbo].[Feeder_run_length]"

sql1 = sql1 + " ON [EdisonML].[dbo].[Feeders_With_PILC].[feeder]=[EdisonML].[dbo].[Feeder_run_length].[feeder]"

'sql1 = sql1 + " WHERE borough='" & cmbSearchBorough.SelectedValue & "' "

```
                sql1 = sql1 + " AND (([feeder_pilcruns_length] *100/[feeder_cable_length]) >= "
& pct2 & ") "

                ' sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select
DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) "

                sql1 = sql1 + "and [EdisonML].[dbo].[Feeders_With_PILC].[feeder]IN (Select
DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) "


                'sql1 = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then
right(feeder,4) else feeder end) FROM EdisonML.dbo.Feeders_With_PILC WHERE feeder IN (Select DISTINCT
feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND pct_PILC >= " & pct2 & " ORDER BY
feeder"

                    End If

                End If

            End If



            '--*modification by shringika on 7-jan-2008 ends here

            Dim myCommand1 As New SqlDataAdapter(sql1, myConnection1)

            myCommand1.SelectCommand.CommandTimeout = 180

            Dim dt1 As DataTable = New DataTable
```

```
myCommand1.Fill(dt1)
If dt1.Rows.Count > 0 Then
    msglabel.Text = " "
    FMFeeder.Enabled = True

    '--modification begins here on 20th dec.2007

    'cmbSearchFeeder.DataSource = GetFeederTable(cmbSearchNetwork.SelectedValue)
    cmbSearchFeeder.DataSource = dt1
    cmbSearchFeeder.DataValueField = "feeder"
    cmbSearchFeeder.DataTextField = "feeder"
    cmbSearchFeeder.DataBind()
    cmbSearchFeeder.SelectedIndex = 0
    SetupFields()

Else
    FMFeeder.Enabled = False
    msglabel.Text = "no feeder found matching the criteria"
```

```
cmbSearchFeeder.Items.Clear()

ResetFeederAttributes()



' CAPTWizard.DisplaySideBar = False

' Dim btn1 As Button =
TryCast(CAPTWizard.FindControl("StartNavigationTemplateContainerID").FindControl("StartNextButton"),
Button)

' btn1.Enabled = False

btn.Enabled = False

dl.Enabled = False



"--*modification on 2 january'2008 for wizard start button display ends



End If



myCommand1 = Nothing
```

```
myConnection1.Close()

myConnection1 = Nothing



TargetCostCheck.Visible = False

actualTotalCostPILC.Text = ""

actualPercentPILC.Text = ""

SectionsChangedPILC.Text = ""

TotalCostPILC.Text = ""

RblTotalCostPercentagePILC.Items(0).Enabled = True

downloadtoExcelPanel.Visible = False

RblTotalCostPercentagePILC.SelectedIndex = 0



PercentPILC.SelectedIndex = 0

CostPercentPanel.Visible = True
```

```
'--modification on 20-dec-2007 ends here

'--original bindsearchfeederlist() sub




'cmbSearchFeeder.DataSource = GetFeederTable(cmbSearchNetwork.SelectedValue)

'cmbSearchFeeder.DataValueField = "FeederID"

'cmbSearchFeeder.DataTextField = "FeederID"

'cmbSearchFeeder.DataBind()

'SetupFields()

End Sub

Private Function GetNetworkTable(ByVal borough As String) As DataTable




'--modified by shringika on 3-jan-2008 to add "All" borough case



Dim sql As String



If cmbSearchBorough.SelectedValue.Contains("A") Then
```

'enable all boroughs again -- agb

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE borough <>'M' AND Network_ID IN (SELECT DISTINCT Nwk FROM Attributes.dbo.Rerank_Attributes as A INNER JOIN Attributes.dbo.tblFeeders_2007 as F on (A.Feeder= F.FeederID)) ORDER BY Network_Name"

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Network_ID IN (SELECT DISTINCT Nwk FROM Attributes.dbo.Rerank_Attributes as A INNER JOIN Attributes.dbo.tblFeeders_2007 as F on (A.Feeder= F.FeederID)) ORDER BY Network_Name"

'commented on 01/12/2009 for getting only network_id

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Network_ID IN (SELECT DISTINCT Network from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) ORDER BY Network_Name"

sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Network_ID IN (SELECT DISTINCT Network from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) ORDER BY Network_Name"

'sql = "Select DISTINCT Network from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1"

Else

'sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE Borough='" & borough & "' AND Network_ID IN (SELECT DISTINCT Nwk FROM Attributes.dbo.Rerank_Attributes as A INNER JOIN Attributes.dbo.tblFeeders_2007 as F on (A.Feeder= F.FeederID)) ORDER BY Network_Name"

'commented below on 01/12/2009 for getting only network_id

sql = "SELECT Network_ID, (Network_Name+'('+Network_ID+')') as Network_Name FROM Networks WHERE District_cd ='" & borough & "' AND Network_ID IN (SELECT DISTINCT Network from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) ORDER BY Network_Name"

' sql = "SELECT DISTINCT Network from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1 and Borough='" & borough & "' "

End If

'--**end of modification for "All" boroughs case 3-jan-2008 shringika

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(Sql, myConnection)

```
        Dim dt As DataTable = New DataTable
        myCommand.SelectCommand.CommandTimeout = 120
        myCommand.Fill(dt)

        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing

        Return dt
    End Function

    Private Function GetFeederTable(ByVal network As String) As DataTable
        '--modified by shringika on dec 20'2007
        Dim sql As String
        Dim myConnection As New SqlConnection
        myConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)
```

' '--modified by shringika 7-jan-2008 included if clause for "All" network case--feedrtype = all case

'cmbfeedertype=all case

'specific borugh case

If cmbSearchBorough.SelectedIndex <> 0 Then

If cmbSearchNetwork.SelectedValue.Contains("All") Then

'sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) AND borough='" & cmbSearchBorough.SelectedValue & "'ORDER BY feeder"

'commented on 01/12/2009 for edisonml

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "'ORDER BY feeder"

sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from

EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "'ORDER BY feeder"

Else

'commented on 01/12/2009 for edisonml

'sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) AND borough='" & cmbSearchBorough.SelectedValue & "' and network='" & cmbSearchNetwork.SelectedValue & "'ORDER BY feeder"

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' and network='" & cmbSearchNetwork.SelectedValue & "'ORDER BY feeder"

sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND borough='" & cmbSearchBorough.SelectedValue & "' and network='" & cmbSearchNetwork.SelectedValue & "'ORDER BY feeder"

End If

Else

'all borough case

If cmbSearchNetwork.SelectedValue.Contains("All") Then

'sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) ORDER BY feeder"

'commented on 01/12/2009 for edisonml

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) ORDER BY feeder"

sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) ORDER BY feeder"

Else

'commented on 01/12/2009 for edisonml

' sql = "SELECT DISTINCT(feeder) as FeederID FROM Assets.dbo.Rerank_Cable_Runs WHERE feeder IN(SELECT feeder from Att_Cable_section_rank_pilc_all_abbr) AND network='" & cmbSearchNetwork.SelectedValue & "' ORDER BY feeder"

'sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4) else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT feeder_ID from

```
EdisonMl.dbo.FDR_Feeders WHERE iscanonical=1) AND network='" & cmbSearchNetwork.SelectedValue & "'
ORDER BY feeder"

                    sql = "SELECT DISTINCT feeder =(case when left(feeder,1)=0 then right(feeder,4)
else feeder end) FROM EdisonML.dbo.Rerank_Cable_Runs WHERE feeder IN (Select DISTINCT zfeeder from
EdisonMl.dbo.Rerank_CanonicalFeeder WHERE iscanonical=1) AND network='" &
cmbSearchNetwork.SelectedValue & "' ORDER BY feeder"

                End If

            End If



            '--*modification ends 9-jan-2008



            Dim myCommand As New SqlDataAdapter(sql, myConnection)



            Dim dt As DataTable = New DataTable

            myCommand.SelectCommand.CommandTimeout = 120

            myCommand.Fill(dt)

            myCommand = Nothing

            myConnection.Close()
```

```
            myConnection = Nothing


            Return dt

        End Function


#End Region


#Region " Attributes "


    '0 for now!!

    Private Function drcheck(ByVal dr As DataRow, ByVal item As String) As String

        If IsDBNull(dr(item)) Then Return "9999" Else Return dr(item)

    End Function


    Private Sub BindAttributes()

        BindNetworkJeopardyAttributes()
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim fdrvalue As String

fdrvalue = cmbSearchFeeder.SelectedValue

If cmbSearchFeeder.SelectedValue.Length < 5 Then

fdrvalue = "0" + cmbSearchFeeder.SelectedValue

End If

'

'FeederOAChartLink.NavigateUrl = "http://edpro.conedison.net/EdisonProgram/CapitalPlanning/Chart/OA.aspx?feeder=" & fdrvalue

FeederOAChartLink.NavigateUrl = "Chart/OA.aspx?feeder=" & fdrvalue

Dim sql As String = "SELECT * FROM Rerank_Attributes WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



'NB: Grid views monitor Feeder to re-query!!!



Label2.Text = cmbSearchFeeder.SelectedValue



Feeder.Text = cmbSearchFeeder.SelectedValue

Feeder1.Text = cmbSearchFeeder.SelectedValue

Feeder2.Text = cmbSearchFeeder.SelectedValue



Feeder4.Text = cmbSearchFeeder.SelectedValue

' Feeder5.Text = cmbSearchFeeder.SelectedValue

Feeder6.Text = cmbSearchFeeder.SelectedValue

Feeder7.Text = cmbSearchFeeder.SelectedValue

feeder11.Text = cmbSearchFeeder.SelectedValue
```

```
feeder12.Text = cmbSearchFeeder.SelectedValue

feeder20.Text = cmbSearchFeeder.SelectedValue

Feeder_Name.Text = cmbSearchFeeder.SelectedValue

SelectedFeederName.Text = cmbSearchFeeder.SelectedValue




FLoad.Text = drcheck(dr, "mon_Load")

Normal_Rating.Text = drcheck(dr, "mon_Normal_Rating")

Emergency_Rating.Text = drcheck(dr, "mon_Emergency_Rating")

Shift.Text = drcheck(dr, "mon_Shift")

Delta.Text = drcheck(dr, "mon_Delta")

Emergency_Load.Text = drcheck(dr, "mon_Emergency_Load")

Feeders_4kv.Text = drcheck(dr, "mon_4kv")

Paper_GT40.Text = drcheck(dr, "mon_Paper_GT40")

Paper_LE40.Text = drcheck(dr, "mon_Paper_LE40")

'changed by shringika to exclude mon_solid_LT16 and mon_solid_GT16

Solid_LT16.Text = drcheck(dr, "mon_Solid_LT16")

Solid_70_75.Text = drcheck(dr, "mon_Solid_70_75")
```

```
Solid_GT16.Text = drcheck(dr, "mon_Solid_GT16")

Percent_Paper_GT40.Text = drcheck(dr, "mon_percent_Paper_GT40")

Percent_Paper_LE40.Text = drcheck(dr, "mon_percent_Paper_LE40")

Percent_Solid_LT16.Text = drcheck(dr, "mon_percent_Solid_LT16")

Percent_Solid_70_75.Text = drcheck(dr, "mon_percent_Solid_70_75")

Percent_Solid_GT16.Text = drcheck(dr, "mon_percent_Solid_GT16")

Sec_CNT.Text = drcheck(dr, "mon_Sec_CNT")

E21GT93.Text = drcheck(dr, "mon_E21GT93")

E21LE93.Text = drcheck(dr, "mon_E21LE93")

R31GT94.Text = drcheck(dr, "mon_R31GT94")

R31LE93.Text = drcheck(dr, "mon_R31LE93")

OtherStop_LT7.Text = drcheck(dr, "mon_OtherStop_LT7")

OtherStop7_12.Text = drcheck(dr, "mon_OtherStop7_12")

OtherStop_GT12.Text = drcheck(dr, "mon_OtherStop_GT12")

AllStop.Text = drcheck(dr, "mon_AllStop")

Percent_ALLStop.Text = drcheck(dr, "mon_percent_AllStop")

PJ_LT8.Text = drcheck(dr, "mon_PJ_LT8")

PJ8_16.Text = drcheck(dr, "mon_PJ8_16")
```

```
PJ16_24.Text = drcheck(dr, "mon_PJ16_24")

' PJ_GT24.Text = drcheck(dr, "mon_PJ_GT24")

PJ24_32.Text = drcheck(dr, "mon_PJ24-32")

PJ32_40.Text = drcheck(dr, "mon_PJ32-40")

PJGT40.Text = drcheck(dr, "mon_PJGT40")

AllPaper.Text = drcheck(dr, "mon_AllPaper")

Percent_AllPaper.Text = drcheck(dr, "mon_Percent_AllPaper")

SJ_LT8.Text = drcheck(dr, "mon_SJ_LT8")

SJ8_16.Text = drcheck(dr, "mon_SJ8_16")

SJ16_24.Text = drcheck(dr, "mon_SJ16_24")

SJ_GT24.Text = drcheck(dr, "mon_SJ_GT24")

AllSolid.Text = drcheck(dr, "mon_AllSolid")

Percent_AllSolid.Text = drcheck(dr, "mon_percent_AllSolid")

Percent_E21GT93.Text = drcheck(dr, "mon_percent_E21GT93")

Percent_E21LE93.Text = drcheck(dr, "mon_percent_E21LE93")

Percent_R31GT94.Text = drcheck(dr, "mon_percent_R31GT94")

Percent_R31LE93.Text = drcheck(dr, "mon_percent_R31LE93")

Percent_OtherStop_LT7.Text = drcheck(dr, "mon_percent_OtherStop_LT7")
```

```
Percent_OtherStop7_12.Text = drcheck(dr, "mon_percent_OtherStop7_12")

Percent_OtherStop_GT12.Text = drcheck(dr, "mon_percent_OtherStop_GT12")

Percent_PJ_LT8.Text = drcheck(dr, "mon_percent_PJ_LT8")

Percent_PJ8_16.Text = drcheck(dr, "mon_percent_PJ8_16")

Percent_PJ16_24.Text = drcheck(dr, "mon_percent_PJ16_24")

'Percent_PJ_GT24.Text = drcheck(dr, "mon_percent_PJ_GT24")

Percent_PJ24_32.Text = drcheck(dr, "mon_percent_PJ24-32")

Percent_PJ32_40.Text = drcheck(dr, "mon_percent_PJ32-40")

Percent_PJGT40.Text = drcheck(dr, "mon_percent_PJGT40")

Percent_SJ_LT8.Text = drcheck(dr, "mon_percent_SJ_LT8")

Percent_SJ8_16.Text = drcheck(dr, "mon_percent_SJ8_16")

Percent_SJ16_24.Text = drcheck(dr, "mon_percent_SJ16_24")

Percent_SJ_GT24.Text = drcheck(dr, "mon_percent_SJ_GT24")

Joint_CNT.Text = drcheck(dr, "mon_Joint_CNT")

XFMR_GT40.Text = drcheck(dr, "mon_XFMR_GT40")

XFMR30_39.Text = drcheck(dr, "mon_XFMR30_39")

XFMR20_29.Text = drcheck(dr, "mon_XFMR20_29")

XFMR10_19.Text = drcheck(dr, "mon_XFMR10_19")
```

```
XFMR0_9.Text = drcheck(dr, "mon_XFMR0_9")

W_XFMR_GT40.Text = drcheck(dr, "mon_W_XFMR_GT40")

W_XFMR30_39.Text = drcheck(dr, "mon_W_XFMR30_39")

W_XFMR20_29.Text = drcheck(dr, "mon_W_XFMR20_29")

W_XFMR10_19.Text = drcheck(dr, "mon_W_XFMR10_19")

W_XFMR0_9.Text = drcheck(dr, "mon_W_XFMR0_9")

Percent_XFMR_GT40.Text = drcheck(dr, "mon_percent_XFMR_GT40")

Percent_XFMR30_39.Text = drcheck(dr, "mon_percent_XFMR30_39")

Percent_XFMR20_29.Text = drcheck(dr, "mon_percent_XFMR20_29")

Percent_XFMR10_19.Text = drcheck(dr, "mon_percent_XFMR10_19")

Percent_XFMR0_9.Text = drcheck(dr, "mon_percent_XFMR0_9")

Percent_W_XFMR_GT40.Text = drcheck(dr, "mon_percent_W_XFMR_GT40")

Percent_W_XFMR30_39.Text = drcheck(dr, "mon_percent_W_XFMR30_39")

Percent_W_XFMR20_29.Text = drcheck(dr, "mon_percent_W_XFMR20_29")

Percent_W_XFMR10_19.Text = drcheck(dr, "mon_percent_W_XFMR10_19")

Percent_W_XFMR0_9.Text = drcheck(dr, "mon_percent_W_XFMR0_9")

XFMR_CNT_SUMMED.Text = drcheck(dr, "mon_XFMR_CNT_SUMMED")

fJeopardy.Text = drcheck(dr, "Jeopardy")
```

```
            fCNT.Text = drcheck(dr, "CNT")




            myCommand = Nothing

            myConnection.Close()

            myConnection = Nothing

        End If

    End Sub



    Private Sub BindNetworkJeopardyAttributes()

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found



        'deals with the All network selection case

        If cmbSearchNetwork.SelectedValue.Contains("All") Then

            base_ntwk_jeopardy_ranking.Text = ""

            Exit Sub

        End If
```

```
If cmbSearchFeeder.Items.Count > 0 Then

Dim sql As String = "SELECT [2008 Ranking] as ntwk_ranking, ntwk FROM [Network_Jeopardy_2008_5_31_08] where ntwk = '" & cmbSearchNetwork.SelectedValue & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

base_ntwk_jeopardy_ranking.Text = drcheck(dr, "ntwk_ranking")
```

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If
End Sub
Private Sub Reset_New_Attributes()

    mon_Load.Text = FLoad.Text
    mon_Normal_Rating.Text = Normal_Rating.Text
    mon_Emergency_Rating.Text = Emergency_Rating.Text
    mon_Shift.Text = Shift.Text
    mon_Delta.Text = Delta.Text
    mon_Emergency_Load.Text = Emergency_Load.Text
    mon_4kv.Text = Feeders_4kv.Text
    mon_Paper_GT40.Text = Paper_GT40.Text
    mon_Paper_LE40.Text = Paper_LE40.Text
    mon_Solid_LT16.Text = Solid_LT16.Text
```

mon_Solid_70_75.Text = Solid_70_75.Text

mon_Solid_GT16.Text = Solid_GT16.Text

mon_Percent_Paper_GT40.Text = Percent_Paper_GT40.Text

mon_Percent_Paper_LE40.Text = Percent_Paper_LE40.Text

mon_Percent_Solid_LT16.Text = Percent_Solid_LT16.Text

mon_Percent_Solid_70_75.Text = Percent_Solid_70_75.Text

mon_Percent_Solid_GT16.Text = Percent_Solid_GT16.Text

mon_Sec_CNT.Text = Sec_CNT.Text

mon_E21GT93.Text = E21GT93.Text

mon_E21LE93.Text = E21LE93.Text

mon_R31GT94.Text = R31GT94.Text

mon_R31LE93.Text = R31LE93.Text

mon_OtherStop_LT7.Text = OtherStop_LT7.Text

mon_OtherStop7_12.Text = OtherStop7_12.Text

mon_OtherStop_GT12.Text = OtherStop_GT12.Text

mon_AllStop.Text = AllStop.Text

mon_Percent_ALLStop.Text = Percent_ALLStop.Text

mon_PJ_LT8.Text = PJ_LT8.Text

mon_PJ8_16.Text = PJ8_16.Text

mon_PJ16_24.Text = PJ16_24.Text

'mon_PJ_GT24.Text = PJ_GT24.Text

mon_PJ24_32.Text = PJ24_32.Text

mon_PJ32_40.Text = PJ32_40.Text

mon_PJGT40.Text = PJGT40.Text

mon_AllPaper.Text = AllPaper.Text

mon_Percent_AllPaper.Text = Percent_AllPaper.Text

mon_SJ_LT8.Text = SJ_LT8.Text

mon_SJ8_16.Text = SJ8_16.Text

mon_SJ16_24.Text = SJ16_24.Text

mon_SJ_GT24.Text = SJ_GT24.Text

mon_AllSolid.Text = AllSolid.Text

mon_Percent_AllSolid.Text = Percent_AllSolid.Text

mon_Percent_E21GT93.Text = Percent_E21GT93.Text

mon_Percent_E21LE93.Text = Percent_E21LE93.Text

mon_Percent_R31GT94.Text = Percent_R31GT94.Text

mon_Percent_R31LE93.Text = Percent_R31LE93.Text

mon_Percent_OtherStop_LT7.Text = Percent_OtherStop_LT7.Text

mon_Percent_OtherStop7_12.Text = Percent_OtherStop7_12.Text

mon_Percent_OtherStop_GT12.Text = Percent_OtherStop_GT12.Text

mon_Percent_PJ_LT8.Text = Percent_PJ_LT8.Text

mon_Percent_PJ8_16.Text = Percent_PJ8_16.Text

mon_Percent_PJ16_24.Text = Percent_PJ16_24.Text

'mon_Percent_PJ_GT24.Text = Percent_PJ_GT24.Text

mon_Percent_PJ24_32.Text = Percent_PJ24_32.Text

mon_Percent_PJ32_40.Text = Percent_PJ32_40.Text

mon_Percent_PJGT40.Text = Percent_PJGT40.Text

mon_Percent_SJ_LT8.Text = Percent_SJ_LT8.Text

mon_Percent_SJ8_16.Text = Percent_SJ8_16.Text

mon_Percent_SJ16_24.Text = Percent_SJ16_24.Text

mon_Percent_SJ_GT24.Text = Percent_SJ_GT24.Text

mon_Joint_CNT.Text = Joint_CNT.Text

mon_XFMR_GT40.Text = XFMR_GT40.Text

mon_XFMR30_39.Text = XFMR30_39.Text

mon_XFMR20_29.Text = XFMR20_29.Text

mon_XFMR10_19.Text = XFMR10_19.Text

mon_XFMR0_9.Text = XFMR0_9.Text

mon_W_XFMR_GT40.Text = W_XFMR_GT40.Text

mon_W_XFMR30_39.Text = W_XFMR30_39.Text

mon_W_XFMR20_29.Text = W_XFMR20_29.Text

mon_W_XFMR10_19.Text = W_XFMR10_19.Text

mon_W_XFMR0_9.Text = W_XFMR0_9.Text

mon_Percent_XFMR_GT40.Text = Percent_XFMR_GT40.Text

mon_Percent_XFMR30_39.Text = Percent_XFMR30_39.Text

mon_Percent_XFMR20_29.Text = Percent_XFMR20_29.Text

mon_Percent_XFMR10_19.Text = Percent_XFMR10_19.Text

mon_Percent_XFMR0_9.Text = Percent_XFMR0_9.Text

mon_Percent_W_XFMR_GT40.Text = Percent_W_XFMR_GT40.Text

mon_Percent_W_XFMR30_39.Text = Percent_W_XFMR30_39.Text

mon_Percent_W_XFMR20_29.Text = Percent_W_XFMR20_29.Text

mon_Percent_W_XFMR10_19.Text = Percent_W_XFMR10_19.Text

mon_Percent_W_XFMR0_9.Text = Percent_W_XFMR0_9.Text

mon_XFMR_CNT_SUMMED.Text = XFMR_CNT_SUMMED.Text

```
        Jeopardy.Text = fJeopardy.Text

        CNT.Text = fCNT.Text

        LPW.Text = fLPW.Text

        AVG_LPW.Text = fAVG_LPW.Text

    End Sub



#End Region



#Region " Ranking & LPW "



    Private Sub BindRankingLPW()

        If cmbSearchFeeder.Items.Count > 0 Then

            Dim feeder As String

            feeder = cmbSearchFeeder.SelectedValue

            If Feeder.Length < 5 Then

                Feeder = "0" + Feeder.ToString

            End If
```

```
' Dim sql As String = "SELECT * FROM FeederSuscept.dbo.rerank_dynamic_Attrib WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim sql As String = "SELECT * FROM LoadPocket.dbo.latest_lpw WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

'Dim sql As String = "SELECT * FROM [Transformer_OM_BO].[dbo].[loadpocketweight_test] WHERE Feeder = '" + feeder + "'"

'Dim sql As String = "SELECT * FROM rerank_dynamic_Attrib WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("DB2FeederSuscept").ConnectionString)

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300
```

```
Dim dt As DataTable = New DataTable



myCommand.Fill(dt)

If dt.Rows.Count = 0 Then

   fAVG_LPW.Text = "0"

   fLPW.Text = "0"

   Exit Sub

End If




Dim dr As DataRow = dt.Rows(0)

'getting the rank from the precomputed table now

'RankBaseLoad.Text = drcheck(dr, "Rank")

Dim x, y, z As Integer



x = CInt(drcheck(dr, "Total_weight"))

y = CInt(drcheck(dr, "RMS_weight"))
```

```
        z = CInt(drcheck(dr, "no_of_xfrs"))

        fLPW.Text = Convert.ToString(x + y)

        ' IR_ScanTime.Text = dr("IR_nwk_scan_datetime").ToString

        ' fAVG_LPW.Text = drcheck(dr, "AVG_LPW")

        fAVG_LPW.Text = Convert.ToString(x / z)



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing



        IR_Scandatetime()

    End If

End Sub



Private Sub BindFOT()

    If cmbSearchFeeder.Items.Count > 0 Then
```

'Dim sql As String = "SELECT Isnull(mtbf,9999) AS MTBF, Isnull(stdevmtbf,9999) AS StDevMTBF, Isnull(countoa,0) AS CountOA, Isnull(countcioa,0)AS CountCIOA, Isnull(countalloutages,0) AS CountAllOutages, ISnull(avgoutageduration,9999) AS AvgOutageDuration, IsNull(lastautodate, '1900-01-01') as LastAutoDate,Isnull(Datediff(DAY,lastautodate,Getdate()),9999) AS DaysLastOA, IsNull(lastscheduledworkdate, '1900-01-01') as LastScheduledWorkDate,Isnull(lastsuccessfulhipotdate,'1900-01-01') AS LastSuccessfulHipotDate from FeederStudy.dbo.tblFeederStats WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

'Dim sql As String = "SELECT Isnull(mtbf,9999) AS MTBF, Isnull(stdevmtbf,9999) AS StDevMTBF, Isnull(countoa,0) AS CountOA, Isnull(countcioa,0)AS CountCIOA, Isnull(countalloutages,0) AS CountAllOutages, ISnull(avgoutageduration,9999) AS AvgOutageDuration, IsNull(lastautodate, '1900-01-01') as LastAutoDate,Isnull(Datediff(DAY,lastautodate,Getdate()),9999) AS DaysLastOA, IsNull(lastscheduledworkdate, '1900-01-01') as LastScheduledWorkDate,Isnull(lastsuccessfulhipotdate,'1900-01-01') AS LastSuccessfulHipotDate from B18003R1.ENG.dbo.tblFeederStats WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim sql As String = "SELECT * from [Attributes].[dbo].[Rerank_CAPT_tblFeederStats] WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 120

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

```
ACTMtbf.Text = ""

'Other stats fill

'StDevMTBF.Text = dr("StDevMTBF")

CountOA.Text = ""

CountCIOA.Text = ""

CountAllOutages.Text = ""

LastAutoDate.Text = ""

DaysLastOA.Text = ""

LastScheduledWorkDate.Text = ""

LastSuccessfulHipotDate.Text = ""



If (dt.Rows.Count > 0) Then

   Dim dr As DataRow = dt.Rows(0)









   ACTMtbf.Text = dr("MTBF")

   'Other stats fill
```

```
'StDevMTBF.Text = dr("StDevMTBF")

CountOA.Text = dr("CountOA")

CountCIOA.Text = dr("CountCIOA")

CountAllOutages.Text = dr("CountAllOutages")

LastAutoDate.Text = dr("LastAutoDate")

DaysLastOA.Text = dr("DaysLastOA")

LastScheduledWorkDate.Text = dr("LastScheduledWorkDate")

LastSuccessfulHipotDate.Text = dr("LastSuccessfulHipotDate")



'LastSuccessfulHipotDate.Text = cmbSearchFeeder.SelectedValue



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Else

LastAutoDate.Text = "No Stats New Feeder"



End If
```

End If

End Sub

#End Region

Protected Sub cmbSearchFeeder_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmbSearchFeeder.SelectedIndexChanged

'--modified by shringika on 20 dec. included if an end if to check 'itemcount in cmbsearchfeeder

If cmbSearchFeeder.Items.Count > 0 Then

```
'ovsec.Visible = True

'ovruns.Visible = True

'PILC.Visible = True

'Runs.Visible = True

'XLP.Visible = False



'ELAS.Visible = True

' Dim fdrcheck As String

'fdrcheck = cmbSearchFeeder.SelectedValue

'If cmbSearchFeeder.SelectedValue.Length < 5 Then

'fdrcheck = "0" + cmbSearchFeeder.SelectedValue

'End If



'FeederOAChartLink.NavigateUrl =
"http://edpro.conedison.net/EdisonProgramDevAlpha/CapitalPlanning/Chart/OA.aspx?feeder=" & fdrcheck

'Link included in bindattributes
```

```
BindAttributes()



reset_lpwtotals()

BindRankingLPW()

BindFOT()

Reset_New_Attributes()

'Assumes replace PILC is strategy of choice

'Replace_Paper()

'Calculate_New_Load()

Reset_MTBFs()

Initial_Costs()

SectionsSummarySetup()

'modification by shringika on 03/11/2008 for LPW tier

'BindAttributes1()

'modification ends here LPW



'ovsec.Visible = False

'ovruns.Visible = False
```

```
'PILC.Visible = False

'Runs.Visible = False

'XLP.Visible = False



'delete later

'ELAS.Visible = False



ovsec.Visible = False

ovruns.Visible = False

'runstop.Visible = False



Runs.Visible = False

XLP.Visible = False

PILCCablesWithStopJoints.Visible = False

GVPILCStopJoints.Visible = False

GVRelStopsPanel.Visible = False

GVRelElastPanel.Visible = False
```

```
downloadtoExcelPanel.Visible = False

UpdateAttributesElast.Visible = False

RelGVElastPanel.Visible = False

PILCCalculationReset()

PILCCablesWorstStopJoints.Visible = False

Runs_Select.Visible = False

' change feeder for update attribute

Dim feeder_id As String



feeder_id = feeder_withzeroes_fix(cmbSearchFeeder.SelectedValue)



Uattr.ChangeFeeder(feeder_id)



TargetCostCheck.Visible = False

actualTotalCostPILC.Text = ""

actualPercentPILC.Text = ""

SectionsChangedPILC.Text = ""

TotalCostPILC.Text = ""
```

```
                RblTotalCostPercentagePILC.Items(0).Enabled = True

                downloadtoExcelPanel.Visible = False

                RblTotalCostPercentagePILC.SelectedIndex = 0



                PercentPILC.SelectedIndex = 0

                CostPercentPanel.Visible = True



            End If



        End Sub



        Protected Sub Paper_GT40_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Paper_GT40.TextChanged

            Percent_Paper_GT40.Text = Int32.Parse(Paper_GT40.Text) / Int32.Parse(Sec_CNT.Text)
* 100.0

            'Convert.ToDouble(Paper_GT40.Text).ToString()
```

```
End Sub

''' <summary>

''' returns a table of overloaded sections for replacement

''' </summary>

''' <param name="Feeder"> selected fedeer</param>

''' <param name="PILC"> whether pilc cables or non pilc</param>

''' <returns> a table of ocverloaded sections</returns>

''' <remarks></remarks>


Private Function GetOvsecTable(ByVal Feeder As String, ByVal PILC As Boolean) As
DataTable

  Dim sql As String

  If Feeder.Length < 5 Then Feeder = "0" + Feeder.ToString

  Dim pct As Integer

  If StrategyLoadDropDown.SelectedIndex = 2 Then

    pct = 100

  ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then
```

pct = 105

Else

pct = 110

End If

If PILC Then

'sql = "SELECT * from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND isnull(insulation,'UNK') = 'paper'"

sql = "SELECT (select count(*) from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >=" & pct & " AND isnull(insulation,'UNK') = 'paper') as ovl_cnt,(case when insulation is null then 'epr' else insulation end) as insulation , [borough] ,[network],[vision_id],[feeder] ,[from_struct],[to_struct] ,[load_percentage] ,[run_id] ,[installation_date] from edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') = 'paper'"

'sql = "SELECT borough,network,feeder,From_struct as [from struct], to_struct as [to struct], insulation,load_percentage, run_id as ,vision_id , installation_date as year from edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') = 'paper'"

Else

sql = "SELECT (select count(*) from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper') as ovl_cnt, (case when insulation is null then 'epr' else insulation end) as insulation , [borough] ,[network],[vision_id],[feeder]

,[from_struct],[to_struct] ,[load_percentage] ,[run_id] ,[installation_date] from Edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

'sql = "SELECT borough,network,feeder,From_struct as [from struct], to_struct as [to struct], insulation,load_percentage, run_id, vision_id, installation_date as year from edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 120

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Return dt

End Function

```
Private Function GetOvsecTable_EXCEL(ByVal Feeder As String, ByVal PILC As Boolean) As DataTable

    Dim sql As String

    If Feeder.Length < 5 Then Feeder = "0" + Feeder.ToString

    Dim pct As Integer

    If StrategyLoadDropDown.SelectedIndex = 2 Then

      pct = 100

    ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then

      pct = 105

    Else

      pct = 110

    End If

    If PILC Then

      sql = "SELECT  borough, network,feeder, From_struct as [from struct], to_struct as [to struct], insulation, load_percentage, run_id ,vision_id , installation_date from  edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' and run_id in(select run_id from  edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') = 'paper')"

    Else
```

'sql = "SELECT (select count(*) from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper') as ovl_cnt,* from Edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

sql = "SELECT borough, network, feeder, From_struct as [from struct], to_struct as [to struct], insulation, load_percentage, run_id, vision_id, installation_date from edisonml.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "' AND load_percentage >= " & pct & " AND isnull(insulation,'UNK') != 'paper'"

End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Return dt

```
End Function

Private Function GetsecTable(ByVal Feeder As String, ByVal PILC As Boolean) As
DataTable

  Dim sql As String

  If (Feeder.Length < 5) Then

    Feeder = "0" + Feeder.ToString

  End If

  If PILC Then

    ' commented to change to edisonml 01/13/2009

    sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND isnull(insulation,'UNK') = 'paper'"

    'sql  =  "SELECT    *  from  EdisonML.dbo.Rerank_OVL  WHERE  (case  when
left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "'  AND isnull(insulation,'UNK') = 'paper'"

  Else
```

```
sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND isnull(insulation,'UNK') != 'paper'"

'sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE (case when
left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND isnull(insulation,'UNK') != 'paper'"

End If

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)



Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



myCommand = Nothing

myConnection.Close()

myConnection = Nothing



Return dt

End Function
```

```
Private Function GetOvrunTable(ByVal Feeder As String) As DataTable

    'Dim sql As String = "SELECT * FROM Attributes.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "' and percentage_load >= 100"


    'commented on 01/12/2009 for edisonml

    'Dim sql As String = "SELECT (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" + Feeder + "' AND percentage_load >= 100) as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from Attributes.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND percentage_load >= 100"

    'commented on 01/14/2009 to exclude deltamtbf column because of bulk_reran.....view missing

    'Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE (case when left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND percentage_load >= 100) as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON (case when left(Runs.feeder,1)=0 then right(Runs.feeder,4) else Runs.feeder end)= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND percentage_load >= 100"

    If (Feeder.Length < 5) Then
```

```
    Feeder = "0" + Feeder.ToString
End If

Dim pct As Integer
If StrategyLoadDropDown.SelectedIndex = 2 Then
    pct = 100
ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then
    pct = 105
Else
    pct = 110
End If
'commented below sql on 03/05/2009 for picking up delta information
' Dim sql As String = "SELECT *,length as ovl_cnt FROM EdisonML.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "' and percentage_load >=" & pct & ""
'added on 03/05/2009
Dim sql As String = "SELECT length as ovl_cnt , (case when delta is null then 0 else cast([delta] as decimal(4,1)) end) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS
```

Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND Runs.Run_id is not null and percentage_load >= " & pct & ""

'Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE feeder = '" + Feeder + "' AND percentage_load >= 100) as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN Attributes.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON (case when left(Runs.feeder,1)=0 then right(Runs.feeder,4) else Runs.feeder end)= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND percentage_load >= 100"

'Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE (case when left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND percentage_load >=" & pct & ") as ovl_cnt, cast([delta] as decimal(4,1)) as delta, Runs.feeder, runs.run_id,runs.percentage_load,runs.length from EdisonML.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON (case when left(Runs.feeder,1)=0 then right(Runs.feeder,4) else Runs.feeder end)= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' and percentage_load >=" & pct & ""

'SELECT Runs.feeder, Runs.Run_ID, Runs.rank,percentage_load, Runs.LoadXrank, Runs.length, (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, cast([delta] as decimal(4,1)) as delta from Attributes.dbo.Rerank_PILC_Runs AS Runs INNER JOIN

```
EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder
AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & Feeder & "' ORDER BY Runs.rank DESC

                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)


                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)


                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing


                Return dt

            End Function
```

```
Private Function GetPILCLoadTable(ByVal Feeder As String) As DataTable

    If (Feeder.Length < 5) Then

        Feeder = "0" + Feeder.ToString

    End If
```

```
    Dim sql As String = "SELECT [csa], [nwk], [feeder], [from_struc], [to_struc], [Insulation]
,(case when [load_percentage] is null then 99999 else [load_percentage] END) as load_percentage, [run_id] ,
[length] , [From_Leg], [To_Leg], [Vision_ID] ,[installation_date], (case when [boro cable rank] is null then 99999
else [boro cable rank] END) as [boro cable rank] ,[boro cable 5tile grp] ,(case when [nwk cable rank] is null then
99999 else [nwk cable rank] END) as [nwk cable rank] ,[nwk cable 5tile grp]  ,(case when [feeder pilc rank] is null
then 99999 else [feeder pilc rank] END) as [feeder pilc rank] ,[feeder pilc 5tile grp],[description],(case when
[LoadXrank] is null then 0 else [Loadxrank] END) as Loadxrank FROM Rerank_pilc_load  WHERE Run_id is not
null and Feeder = '" + Feeder + "'"

    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)

    Dim dt As DataTable = New DataTable

    myCommand.SelectCommand.CommandTimeout = 60

    myCommand.Fill(dt)
```

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing

        Return dt
    End Function

    Private Function GetRunLoadTable(ByVal Feeder As String) As DataTable
        If (Feeder.Length < 5) Then
            Feeder = "0" + Feeder.ToString
        End If

        'probably need to provide the lower rank section as the rank
        'Dim sql As String = "SELECT (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, * from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "'"
```

```
Dim sql As String = "SELECT (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, * from Edisonml.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "'"

'Dim sql As String = "SELECT * FROM Attributes.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

With totalColumn
```

```
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "tsec"
    '   .Expression = "sum(length)"
End With
With perColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "psec"
End With
With pertotColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "ptsec"
End With
With dt.Columns
    .Add(totalColumn)
    .Add(perColumn)
    .Add(pertotColumn)
End With
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing



        Return dt

    End Function



    Private Function GetPILCRunLoadTable(ByVal Feeder As String) As DataTable

        If (Feeder.Length < 5) Then

            Feeder = "0" + Feeder.ToString

        End If




        'probably need to provide the lower rank section as the rank

        'commented on 01/15/2009 for excluding delta mtbf column

        'Dim sql As String = "SELECT Runs.feeder, Runs.Run_ID, Runs.rank,percentage_load,
Runs.LoadXrank, Runs.length, (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" &
```

Feeder & "') as ovl_cnt, cast([delta] as decimal(4,1)) as delta from Attributes.dbo.Rerank_PILC_Runs AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & Feeder & "' ORDER BY Runs.rank DESC"

'modified on 03/05/2009 for including delta info from bulk_rerank

'Dim sql As String = "SELECT feeder,Run_ID,rank,percentage_load,LoadXrank,length, (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt from EdisonML.dbo.Rerank_PILC_Runs WHERE feeder = '" & Feeder & "' ORDER BY rank DESC"

'(case when [Feeder Joint Rank] is null then 99999 else [feeder joint rank] END)

Dim sql As String = "SELECT Runs.feeder, Runs.Run_ID,(case when Runs.rank is null then 99999 else Runs.rank end) as rank, (case when [percentage_load] is null then 99999 else [percentage_load] END) as percentage_load, (case when [LoadXrank] is null then 0 else [Loadxrank] END) as Loadxrank, Runs.length, (select sum(length) from EdisonML.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt, (case when delta is null then 0 else cast([delta] as decimal(4,1)) end) as delta, is_spur, is_basket from EdisonML.dbo.Rerank_PILC_Runs AS Runs LEFT OUTER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & Feeder & "' and Runs.Run_id is not null ORDER BY Runs.rank DESC"

'modification on 03/05/2009 for delta ends here

'Dim sql As String = "SELECT feeder,Run_ID,rank,percentage_load,LoadXrank,length, (select sum(length) from Attributes.dbo.Rerank_PILC_Runs WHERE Feeder = '" & Feeder & "') as ovl_cnt from Attributes.dbo.Rerank_PILC_Runs WHERE feeder = '" & Feeder & "' ORDER BY rank DESC"

'Dim sql As String = "SELECT * FROM Attributes.dbo.Rerank_pilc_runs WHERE [Feeder] = '" + Feeder + "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

With totalColumn

.DataType = System.Type.GetType("System.Decimal")

```
    .ColumnName = "tsec"
    '   .Expression = "sum(length)"
End With
With perColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "psec"
End With
With pertotColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "ptsec"
End With
With dt.Columns
    .Add(totalColumn)
    .Add(perColumn)
    .Add(pertotColumn)
End With
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing


        Return dt

    End Function


    Private Function GetOVLLoadTableForRun(ByVal RunId As String) As DataTable

        Dim sql As String = "SELECT * FROM Rerank_OVL  WHERE [run_id] = '" + RunId +
"'"

        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

        Dim myCommand As New SqlDataAdapter(sql, myConnection)


        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)
```

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing

        Return dt
    End Function

    Private Function GetCableById(ByVal CableId As String) As DataTable
        Dim sql As String = "SELECT * FROM Rerank_OVL  WHERE [vision_id] = '" + CableId
+ "'"
        Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)
        Dim myCommand As New SqlDataAdapter(sql, myConnection)
        'SectionsSummary.Items.Add(sql)
        Dim dt As DataTable = New DataTable
        myCommand.Fill(dt)

        myCommand = Nothing
```

```
myConnection.Close()

myConnection = Nothing


Return dt

End Function


Private Function GetJointsForCable(ByVal Feeder As String, ByVal Str As String) As
DataTable

Dim sql As String = "SELECT * FROM Rerank_CAPT_jeoJoints WHERE Feeder = '" +
Feeder + "' and Structure = '" + Str + "'"

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)


Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


myCommand = Nothing

myConnection.Close()
```

```
myConnection = Nothing


Return dt

End Function


Private Function GetJointsForCable(ByVal Feeder As String, ByVal Str As String, ByVal Leg As String) As DataTable

Dim sql As String = "SELECT * FROM Rerank_CAPT_jeoJoints WHERE Feeder = '" + Feeder + "' and Structure = '" + Str + "' and Leg = '" + Leg + "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)


Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


myCommand = Nothing

myConnection.Close()

myConnection = Nothing
```

```
    Return dt

End Function


Private Function GetXLPTable(ByVal Feeder As String) As DataTable

    If Feeder.Length < 5 Then

        Feeder = "0" + Feeder

    End If

    'modified by RTau 01/15

    Dim sql As String = "SELECT RL.feeder, RL.run_id,(case when RL.[Feeder PILC
RANK] is null then 99999 else RL.[Feeder PILC RANK] END) AS Feeder_Cable_RANK,(case when
RL.[load_percentage] is null then 99999 else RL.[load_percentage] END) as load_percentage, (case when
RL.[LoadXrank] is null then 99999 else RL.[Loadxrank] END) as LoadXrank, RLR.length as length,
RL.Description, RL.INSTALLATION_DATE, RL.insulation FROM EdisonML.dbo.Rerank_Load RL,
Edisonml.dbo.Rerank_Load_Runs RLR WHERE RL.run_id = RLR.run_id and RL.Feeder = '" & Feeder & "' AND
(RL.insulation = 'XLP') AND (year(RL.INSTALLATION_DATE) between 1970 and 1975)"

    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("edisonml").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)

    myCommand.SelectCommand.CommandTimeout = 120
```

```
Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

With totalColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "tsec"

End With

With perColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "psec"

End With

With pertotColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "ptsec"
```

```
        End With
        With dt.Columns
          .Add(totalColumn)
          .Add(perColumn)
          .Add(pertotColumn)
        End With

        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing

        Return dt
    End Function

    Private Function GetElastimoldTable(ByVal Feeder As String) As DataTable
        'Changed by RTau 12/13/07 Dim sql As String = "SELECT * FROM
[Rerank_Joints] WHERE [Feeder] = '" + Feeder + "' AND [Stop] = 'stop' and [Type] = 'ELASTIMOLD' and
[Config] = '2W-1W'"
```

' Dim sql As String = "Select distinct Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints WHERE Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by run_id order by min([Feeder Joint Rank]) asc"

'Dim sql As String = "SELECT Runs.[Run_ID],Runs.[rank] as worst_joint_rank,Runs.[elastimold],Runs.[RAYCHEM] ,Runs.[other_stops],runs.[length], Runs.[stops],cast([delta] as decimal(4,1)) as delta FROM [Rerank_PILC_Runs_Joints_Types_2008] AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & cmbSearchFeeder.SelectedValue & "' and stops > 0 order by elastimold desc,raychem desc, other_stops desc"

'Dim sql As String = "Select distinct Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints WHERE Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by run_id order by min([Feeder Joint Rank]) asc"

'Dim sql As String = "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as delta FROM Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

'feeder_withzeroes_fix(Feeder)

If Feeder.Length < 5 Then

```
Feeder = "0" + Feeder.ToString

End If

'modified on 03/05/2009 for including delta mtbf column

Dim sql As String = "Select distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as delta FROM Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE runs.Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by Runs.run_id order by min([Feeder Joint Rank]) asc"

' Dim sql As String = "Select distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints AS Runs WHERE runs.Feeder = '" + Feeder + "' AND Stop = 's' and Type = 'ELASTIMOLD' and Config = '2W-1W' GROUP by Runs.run_id order by min([Feeder Joint Rank]) asc"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

'takes 30 secs to do!

myCommand.SelectCommand.CommandTimeout = 60

Dim dt As DataTable = New DataTable



myCommand.Fill(dt)
```

```
'******  added by RTau 12/12/07  ***************

Dim totalColumn As DataColumn = New DataColumn

Dim perColumn As DataColumn = New DataColumn

Dim pertotColumn As DataColumn = New DataColumn

Dim secColumn As DataColumn = New DataColumn

With totalColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "tsec"

End With

With perColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "psec"

End With

With pertotColumn

   .DataType = System.Type.GetType("System.Decimal")

   .ColumnName = "ptsec"

End With
```

```
With secColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "sec"
End With
With dt.Columns
   .Add(totalColumn)
   .Add(perColumn)
   .Add(pertotColumn)
   .Add(secColumn)
End With
'*****  end added by RTau 12/12/07   ***************

myCommand = Nothing
myConnection.Close()
myConnection = Nothing

Return dt
End Function
```

```
Private Function GetStopTable(ByVal Feeder As String) As DataTable

    Dim sql As String = "SELECT * FROM [Rerank_Joints] WHERE [Feeder] = '" + Feeder +
"' AND [Stop] = 'stop'"

    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)

    'takes 30 secs to do!

    myCommand.SelectCommand.CommandTimeout = 120

    Dim dt As DataTable = New DataTable

    myCommand.Fill(dt)




    myCommand = Nothing

    myConnection.Close()

    myConnection = Nothing
```

```
    Return dt

End Function


Private Function GetRunJointsTable(ByVal Feeder As String) As DataTable

    If (Feeder.Length < 5) Then

        Feeder = "0" + Feeder.ToString

    End If


    Dim sql As String = "SELECT distinct wei_id as Joint_ID, Run_ID, (case when [Feeder
Joint Rank] is null then 99999 else [feeder joint rank ] END) as Joint_Rank ,structure, leg, type, config, stop, year
FROM Rerank_Runs_For_Joints WHERE Feeder = '" + Feeder + "' order by Joint_Rank asc"

    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)

    'takes 30 secs to do!

    myCommand.SelectCommand.CommandTimeout = 120

    Dim dt As DataTable = New DataTable

    myCommand.Fill(dt)
```

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing

        Return dt
    End Function

    Private Function GetRunStopTable(ByVal Feeder As String) As DataTable
        If (Feeder.Length < 5) Then
            Feeder = "0" + Feeder.ToString
        End If

        'modified on 03/05/2009 for including delta mtbf column edisonml
        Dim sql As String = "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as
Worst_Joint_Rank,  max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as
```

delta FROM Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '" + Feeder + "' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

'Dim sql As String = "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops FROM Rerank_Runs_For_Joints AS Runs WHERE Runs.Feeder = '" + Feeder + "' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

'dim sql as string= "SELECT distinct Runs.Run_ID, min([Feeder Joint Rank]) as Worst_Joint_Rank, max(length) as length, count(distinct wei_id) as Stops,cast(max([delta]) as decimal(4,1)) as delta FROM EdisonML.dbo.Rerank_Runs_For_Joints AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.Feeder = '4x40' AND Stop = 's' and insulation = 'paper' group by Runs.run_id order by min([Feeder Joint Rank]) asc"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

'takes 30 secs to do!

myCommand.SelectCommand.CommandTimeout = 120

```
Dim dt As DataTable = New DataTable


myCommand.Fill(dt)


Dim totalColumn As DataColumn = New DataColumn
Dim perColumn As DataColumn = New DataColumn
Dim pertotColumn As DataColumn = New DataColumn
Dim secColumn As DataColumn = New DataColumn
With totalColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "tsec"
   '   .Expression = "sum(length)"
End With
With perColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "psec"
End With
With pertotColumn
```

```
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "ptsec"
End With
With secColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "sec"
End With
With dt.Columns
    .Add(totalColumn)
    .Add(perColumn)
    .Add(pertotColumn)
    .Add(secColumn)
End With




myCommand = Nothing
myConnection.Close()
```

```
myConnection = Nothing


Return dt

End Function


Protected Sub CalculateCostLoad_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostLoad.Click

CheckCommitLoad.Checked = False


CalculateCostLoad_CommitChanges()


If SectionsChangedLoad.Text = "0" Then


  ovruns.Visible = True

  ovsec.Visible = True


  UpdateAttributesLoad.Visible = False

  Exit Sub
```

```
    End If

End Sub

Private Sub CalculateCostLoad_CommitChanges()

    '--modified by shringika on dec 21'2007 included if case

    If cmbSearchFeeder.Items.Count > 0 Then

        BindAttributes()

        BindRankingLPW()

        BindFOT()

        ' Reset_New_Attributes()

        'modification by shringika on 03/11/2008 for LPW tier

        ' BindAttributes1()

        'modification ends here 03/11/2008

        Dim sec_changed As Integer
```

```
Dim dv, dv1 As DataView

'I don't use this strategy now
If StrategyLoadDropDown.SelectedIndex = 1 Then

   dv1 = New DataView(GetsecTable(Feeder.Text, True))

   sec_changed = Replace_Overloaded_Sections(dv1)

End If

If StrategyLoadDropDown.SelectedIndex > 1 Then

   dv = New DataView(GetOvrunTable(Feeder.Text))
   'dv.Sort = "load_percentage DESC"
   'GVovruns.DataSource = dv1
   'GVovruns.DataBind()
```

```
'want to detail view to have all sections not just overloaded

dv1 = New DataView(GetsecTable(Feeder.Text, True))

'dv.Sort = "percentage_load DESC"

'GVovsec.DataSource = dv

'GVovsec.DataBind()



Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("OVLRuns")

Dim dt2 As DataTable = New DataTable("OVLSections")

dt1 = dv.ToTable()

dt2 = dv1.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("run_id")
```

```
Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

GVovlruns.DataSource = dss

GVovlruns.DataMember = dss.Tables(0).TableName()

GVovlruns.DataBind()

'GVpilcruns.RowExpanded(0) = True



Dim m_deltaMTBF As String

m_deltaMTBF                                                                    =
Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

   GVovlruns.Columns(5).Visible = False

End If
```

```
Dim dv2 As DataView = New DataView(GetOvsecTable(Feeder.Text, False))



'dv2.Sort = "percentage_load DESC"

GVovlnonsec.DataSource = dv2

GVovlnonsec.DataBind()



Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVovlruns.Items.Count) - 1

  If Not (Equals(VisitedGlbRuns.Items.FindByText(GVovlruns.Items(visited_run_grid).Cells(2).Text), Nothing)) Then

    GVovlruns.Items(visited_run_grid).ForeColor = Drawing.Color.Red

  End If

Next
```

```
            sec_changed = Replace_Overloaded_Sections(dv2) + Replace_Overloaded_Runs(dv)
            SectionsChangedLoad.Text = Convert.ToString(sec_changed)
            Total_Cost_Load()
        End If
        UpdateAttributesLoad.Visible = True
        CommitChangesLoad.Visible = True
        CommitChangesLoad.Enabled = True
        ovruns.Visible = True
        ovsec.Visible = True


    End If
    'included end if for if case on dec 21 2007
    SelectedStrategyLoadPercentage.Text = StrategyLoadDropDown.SelectedValue
    loadStrategyInfo.Visible = True
```

```
End Sub

Private Function GetSectionInRunTable(ByVal RunID As Integer, ByVal Feeder As String)
As DataTable

Dim sql As String



If (Feeder.Length < 5) Then

  Feeder = "0" + Feeder.ToString

End If




" If PILC Then

' commented to change to edisonml 01/13/2009

sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND Run_id = '" & RunID & " ' AND  isnull(insulation,'UNK') = 'paper'"

'sql  =  "SELECT  *  from  EdisonML.dbo.Rerank_OVL  WHERE  (case  when
left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "'  AND isnull(insulation,'UNK') = 'paper'"

' Else

' sql = "SELECT  *  from EdisonML.dbo.Rerank_OVL WHERE Feeder = '" + Feeder + "'
AND Run_id = '" & RunID & " ' AND isnull(insulation,'UNK') != 'paper'"
```

```
'sql = "SELECT * from EdisonML.dbo.Rerank_OVL WHERE (case when left(feeder,1)=0 then right(feeder,4) else feeder end) = '" + Feeder + "' AND isnull(insulation,'UNK') != 'paper'"

' End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)


Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


myCommand = Nothing

myConnection.Close()

myConnection = Nothing


Return dt

End Function
```

```
Protected Sub GVovlruns_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GVovlruns.SelectedIndexChanged

    Dim dt As DataTable

    Dim Run As Integer

    Dim feeder As String

    feeder = cmbSearchFeeder.SelectedValue

    Run = Int32.Parse(GVovlruns.SelectedItem.Cells(2).Text)



    dt = GetSectionInRunTable(Run, feeder)



    Dim struct As String

    struct = dt.Rows(0).Item("from_struct")

    Session("struct") = struct



    Dim URL As String



    URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"
```

```
Dim MyScript As New System.Text.StringBuilder

With MyScript

  .Append("<script type='text/javascript'>")

  .Append("window.open('" & URL & "');")

  .Append("</script>")

End With



'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)



'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text
```

```
End Sub

Private Sub GVovlruns_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles GVovlruns.TemplateSelection

  e.TemplateFilename = "Templates\\OVLSections.ascx"

End Sub

Protected Sub btnovlrunExcel_click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnovlrunExcel.Click

  Dim dt As DataTable = GetOvsecTable_EXCEL(Feeder.Text, True)

  dt.Merge(GetOvsecTable_EXCEL(Feeder.Text, False))

  If dt Is Nothing Then

    Return

  End If
```

```
dt.DefaultView.Sort = "Run_Id ASC, load_percentage DESC"

Dim cols As Integer = dt.Columns.Count

'Output excel file
Response.Clear()
Response.ContentType = "application/xls"
Response.AddHeader("content-disposition", "attachment;
filename=""CapitalPlanning_PILC_Rank.xls""")
Response.Write("<table border='1'>")

Dim i As Integer

'Print Row Headings
Response.Write("<tr>")
Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Borough</td>")
Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Network</td>")
```

```
Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>Feeder</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>From Struct</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>To Struct</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>Insulation</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>% load</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>run id</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>vision id</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR: silver'>Year</td>")

Response.Write("</tr>")


'Print Rows

For Each row As DataRow In dt.Rows
```

```
        Response.Write("<tr>")
        For i = 0 To row.ItemArray.Length() - 1
            Response.Write("<td>" & row(i) & "</td>")
        Next
        Response.Write("</tr>")
    Next
    Response.Write("</table>")
    Response.End()
End Sub



Private Sub CalculateCostPILC_CommitChanges()
    visited_sections_count = 0

    '--modified by shringika on dec 21'2007 included if case
    If cmbSearchFeeder.Items.Count > 0 Then
        If RblTotalCostPercentagePILC.SelectedIndex > 0 Then
            If PercentPILC.Text.Equals("") Or CostPerSectionPILC.Text.Equals("") Then
```

```
            Exit Sub


        End If
    Else
        If TotalCostPILC.Text.Equals("") Then
            TargetCostCheck.Visible = True


            Exit Sub


        End If
        If CostPerSectionPILC.Text.Equals("") Then Exit Sub


    End If


    Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder.Text))
    Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))
```

```
BindAttributes()

BindRankingLPW()

BindFOT()

'Reset_New_Attributes()

'modification by shringika on 003/11/2008 for LPW tier

' BindAttributes1()

'modification ends here 03/11/2008

' temporary default sort

'dv.Sort = "Run_ID ASC"

'dv1.Sort = "rank DESC"



If StrategyPILCDropDown.SelectedIndex = 1 Then



  'dv.Sort = "Run_ID ASC"

  dv.Sort = "[Feeder PILC Rank] ASC"

  dv1.Sort = "rank ASC"

  'Replace_paper_percent uses the sort order from above
```

```
End If

If StrategyPILCDropDown.SelectedIndex = 2 Then
   'dv.Sort = "Run_ID ASC"
   dv.Sort = "load_percentage DESC"
   dv1.Sort = "percentage_load DESC"
End If

If StrategyPILCDropDown.SelectedIndex = 3 Then
   'dv.Sort = "Run_ID ASC"
   dv.Sort = "LoadXrank DESC"
   dv1.Sort = "LoadXrank DESC"
End If

If StrategyPILCDropDown.SelectedIndex = 4 Then
```

```
    StopJointsCalculation()

    Runs.Visible = False

    GVpilcruns.Visible = False

    PILCCablesWithStopJoints.Visible = True

    downloadtoExcelPanel.Visible = True

    GVPILCStopJoints.Visible = True

    UpdateAttributesPILC.Visible = True

    CommitChangesPILC.Visible = True

    'CommitChangesPILC.Enabled = True

    If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"

    Exit Sub

End If



'worst stop joint replace logic

If StrategyPILCDropDown.SelectedIndex = 5 Then



    WorstStopJointsCalculation()
```

```
    Runs.Visible = False

    GVpilcruns.Visible = False

    PILCCablesWorstStopJoints.Visible = True

    downloadtoExcelPanel.Visible = True

    GVPILCStopJointTypes.Visible = True

    UpdateAttributesPILC.Visible = True

    CommitChangesPILC.Visible = True

    'CommitChangesPILC.Enabled = True

    If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"

    Exit Sub

End If



'added by shringika for including total cost case on 02/14/2008

If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

    If CheckCommitPILC.Checked Then
```

SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - duplicate_sections_list)

'Updating Section cost text boxes

Else

'SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - visited_sections_count)

SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1))

End If

Total_Cost_PILC()

Else

If CheckCommitPILC.Checked Then

SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - duplicate_sections_list)

Else

```
                    'SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - visited_sections_count)

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1))

                End If

                Total_Cost_PILC()



            End If

            If SectionsChangedPILC.Text = "0" Then

                actualPercentPILC.Text = "0"

                GVpilcruns.Visible = False

                UpdateAttributesPILC.Visible = False

                downloadtoExcelPanel.Visible = False

                Exit Sub

            End If
```

```
'GVpilcsec.DataSource = dv

'GVpilcsec.DataBind()



'GVpilcruns.DataSource = dv1

'GVpilcruns.DataBind()



' Code for hierarchgrid



Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PICLSections")

dt1 = dv1.ToTable()

dt2 = dv.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)
```

```
Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

Runs.Visible = True

GVpilcruns.Visible = True

UpdateAttributesPILC.Visible = True

CommitChangesPILC.Visible = True

downloadtoExcelPanel.Visible = False

'CommitChangesPILC.Enabled = True

GVpilcruns.DataSource = dss

GVpilcruns.DataMember = dss.Tables(0).TableName()

GVpilcruns.DataBind()
```

```
Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

   GVpilcruns.Columns(10).Visible = False

End If

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVpilcruns.Items.Count) - 1

   If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVpilcruns.Items(visited_run_grid).Cells(2).Text), Nothing)) Then



      GVpilcruns.Items(visited_run_grid).ForeColor = Drawing.Color.Red



   End If
```

```
Next

'GVpilcruns.RowExpanded(0) = True

'lbdebug.Text = dss.Tables(0).TableName()

'CalculateCostPILCAll(StrategyPILCDropDown.SelectedIndex)

End If

'--modification by shringika ends here20-dec-2007
```

```
End Sub



Private Sub WorstStopJointsCalculation()



  Dim dv1 As DataView = New DataView(GetRunsWithWorstStopTable(Feeder.Text))

  Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))

  'runstop.Visible = True

  If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

    If CheckCommitPILC.Checked Then

      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - duplicate_sections_list)

    Else

      'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - visited_sections_count)

      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2))

    End If

    Total_Cost_PILC()
```

```
Else


    If CheckCommitPILC.Checked Then

        SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - duplicate_sections_list)

    Else

        'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - visited_sections_count)

        SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2))

    End If

    Total_Cost_PILC()


End If

If SectionsChangedPILC.Text = "0" Then UpdateAttributesPILC.Visible = False


' Code for hierarchgrid


Dim dss As DataSet = New DataSet()
```

```
Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PILCJoint_Sections")

dt1 = dv1.ToTable()

dt2 = dv2.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("PILCJoint_Sections", dcc1, dcc2, False)

dss.Relations.Add(drr)



'bind the DataSet to the HierarGrid



GVPILCStopJointTypes.DataSource = dss

GVPILCStopJointTypes.DataMember = dss.Tables(0).TableName()
```

```
GVPILCStopJointTypes.DataBind()

Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

  GVPILCStopJointTypes.Columns(5).Visible = False

End If

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVPILCStopJointTypes.Items.Count) - 1

  If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVPILCStopJointTypes.Items(visited_run_grid).Cells(1).Text),
Nothing)) Then


    GVPILCStopJointTypes.Items(visited_run_grid).ForeColor = Drawing.Color.Red



  End If

Next
```

```
        PILCCablesWorstStopJoints.Visible = True

        GVPILCStopJointTypes.Visible = True



    End Sub


    Protected Sub GVpilcruns_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GVpilcruns.SelectedIndexChanged

        Dim dt As DataTable

        Dim Run As Integer

        Dim feeder As String

        feeder = cmbSearchFeeder.SelectedValue

        Run = Int32.Parse(GVpilcruns.SelectedItem.Cells(2).Text)



        dt = GetSectionInRunTable(Run, feeder)
```

```
Dim struct As String

struct = dt.Rows(0).Item("from_struct")

Session("struct") = struct



Dim URL As String



URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"



Dim MyScript1 As New System.Text.StringBuilder

With MyScript1

   .Append("<script type='text/javascript'>")

   .Append("window.open('" & URL & "');")

   .Append("</script>")

End With
```

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap", MyScript.ToString)

ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap", MyScript1.ToString, False)

'RegisterStartupScript(Me, Me.GetType(), "FeederMap1", MyScript1.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub

Private Sub GVpilcruns_TemplateSelection(ByVal sender As Object, ByVal e As DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles GVpilcruns.TemplateSelection

e.TemplateFilename = "Templates\\PILCSections.ascx"

End Sub

Private Sub CalculateCostPILC_SelectRuns()

```
Visited_sections_count = 0

'--modified by shringika on dec 21'2007 included if case
If cmbSearchFeeder.Items.Count > 0 Then

  Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder.Text))
  Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))

  BindAttributes()
  BindRankingLPW()
  BindFOT()
  If StrategyPILCDropDown.SelectedIndex = 1 Then

    'dv.Sort = "Run_ID ASC"
    dv.Sort = "[Feeder PILC Rank] ASC"
    dv1.Sort = "rank ASC"

    'dv1.Sort = "is_basket DESC, is_spur ASC, delta DESC"
```

```
    dv1.Sort = "is_basket DESC, is_spur ASC, rank ASC"

    ViewState("sortdirection") = "ASC"

    'Replace_paper_percent uses the sort order from above

End If



If StrategyPILCDropDown.SelectedIndex = 2 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "load_percentage DESC"

    dv1.Sort = "percentage_load DESC"

    dv1.Sort = "is_basket DESC, is_spur ASC, percentage_load DESC"

    ViewState("sortdirection") = "DESC"

End If



If StrategyPILCDropDown.SelectedIndex = 3 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "LoadXrank DESC"

    dv1.Sort = "LoadXrank DESC"

    dv1.Sort = "is_basket DESC, is_spur ASC, LoadXrank DESC"
```

```
    ViewState("sortdirection") = "DESC"

End If



'If StrategyPILCDropDown.SelectedIndex = 4 Then


'   StopJointsCalculation()

'   Runs.Visible = False

'   GVpilcruns.Visible = False

'   PILCCablesWithStopJoints.Visible = True

'   downloadtoExcelPanel.Visible = True

'   GVPILCStopJoints.Visible = True

'   UpdateAttributesPILC.Visible = True

'   CommitChangesPILC.Visible = True

'   'CommitChangesPILC.Enabled = True

'   If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"

'   Exit Sub
```

```
'End If



"worst stop joint replace logic

'If StrategyPILCDropDown.SelectedIndex = 5 Then



'   WorstStopJointsCalculation()

'   Runs.Visible = False

'   GVpilcruns.Visible = False

'   PILCCablesWorstStopJoints.Visible = True

'   downloadtoExcelPanel.Visible = True

'   GVPILCStopJointTypes.Visible = True

'   UpdateAttributesPILC.Visible = True

'   CommitChangesPILC.Visible = True

'   'CommitChangesPILC.Enabled = True

'   If SectionsChangedPILC.Text = "0" Then actualPercentPILC.Text = "0"

'   Exit Sub

'End If
```

```
'added by shringika for including total cost case on 02/14/2008

" " "If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

" " "   If CheckCommitPILC.Checked Then

"     "     "                                   SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - Duplicate_sections_list)

" " "     'Updating Section cost text boxes



" " "   Else

"     "     "                                   'SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1) - visited_sections_count)

"     "     "                                   SectionsChangedPILC.Text     =
Convert.ToString(Replace_Paper_percent(PercentPILC.Text, dv1))

" " "   End If

" " "   Total_Cost_PILC()

" " "Else
```

```
" " "   If CheckCommitPILC.Checked Then

" "   "   "                                   SectionsChangedPILC.Text   =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - Duplicate_sections_list)

" " "   Else

" "   "   "                                   'SectionsChangedPILC.Text   =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1) - visited_sections_count)

" "   "   "                                   SectionsChangedPILC.Text   =
Convert.ToString(Replace_Paper_TotalCost(TotalCostPILC.Text, dv1))

" " "   End If

" " "   Total_Cost_PILC()



" " "End If

" " "If SectionsChangedPILC.Text = "0" Then

" " "   actualPercentPILC.Text = "0"

" " "   GVpilcruns.Visible = False

" " "   UpdateAttributesPILC.Visible = False

" " "   downloadtoExcelPanel.Visible = False
```

```
" " "    Exit Sub

" " "End If



'GVpilcsec.DataSource = dv

'GVpilcsec.DataBind()



'GVpilcruns.DataSource = dv1

'GVpilcruns.DataBind()



' Code for hierarchgrid
```

```
Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PICLSections")

dt1 = dv1.ToTable()

dt2 = dv.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

Runs_Select.Visible = True

GVPILCRuns_Select.Visible = True

UpdateAttributesPILC.Visible = False
```

```
'CommitChangesPILC.Visible = False

downloadtoExcelPanel.Visible = False

'CommitChangesPILC.Enabled = True

GVPILCRuns_Select.DataSource = dss

GVPILCRuns_Select.DataMember = dss.Tables(0).TableName()

GVPILCRuns_Select.DataBind()

GVpilcruns.Visible = False

Runs.Visible = False



Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") And Not RblTotalCostPercentagePILC.SelectedIndex =
2 Then

   GVPILCRuns_Select.Columns(7).Visible = False

End If

Dim visited_run_grid As Integer
```

```
For visited_run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1

If Not (Equals(VisitedGlbRuns.Items.FindByText(GVPILCRuns_Select.Items(visited_run_grid).Cells(2).Text), Nothing)) Then

GVPILCRuns_Select.Items(visited_run_grid).ForeColor = Drawing.Color.Red

End If

Next

'GVpilcruns.RowExpanded(0) = True
```

```
'lbdebug.Text = dss.Tables(0).TableName()



'CalculateCostPILCAll(StrategyPILCDropDown.SelectedIndex)









    End If

    '--modification by shringika ends here20-dec-2007

End Sub



Private Sub CalculateCostPILC_SelectRunsCommitChanges()



    'Dim run_grid As Integer

    'Dim chk1 As CheckBox

    Dim count As Integer
```

```
count = 0

'For run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1



'   chk1 = GVPILCRuns_Select.Items(run_grid).Cells(7).FindControl("chkselectrun")

'   If chk1.Checked Then

'      count += 1

'      TextBox3.Text = count.ToString

'      Dim MyScript1 As New System.Text.StringBuilder

'      With MyScript1

'         .Append("<script type='text/javascript'>")

'         .Append("alert ('checkbox checked');")

'         .Append("</script>")

'      End With

'                        Page.ClientScript.RegisterStartupScript(GetType(String),    "check",
MyScript1.ToString)



'      'Else

'      '   Dim MyScript1 As New System.Text.StringBuilder
```

```
'        ' With MyScript1

'        '     .Append("<script type='text/javascript'>")

'        '     .Append("alert ('no checkbox');")

'        '     .Append("</script>")

'        ' End With

'                ' Page.ClientScript.RegisterStartupScript(GetType(String), "check",
MyScript1.ToString)



'  End If



'Next

Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))

Dim m_deltaMTBF As String

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);

'm_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))
```

```
'If Equals(m_deltaMTBF, "no") Then

'    GVPILCRuns_Select.Columns(7).Visible = False

'End If



    SectionsChangedPILC.Text = Replace_PILCRunsSelect(dv1).ToString

    Total_Cost_PILC()

End Sub

Private Function Replace_PILCRunsSelect(ByVal runs As DataView) As Integer



    Dim total_sections, pilc_sections, running_total As Integer

    Dim runId As Integer

    Dim run_grid As Integer



    total_sections = Int32.Parse(Sec_CNT.Text)

    Dim select_runs As ArrayList

    select_runs = New ArrayList

    select_runs.Add(0)

    running_total = 0
```

```
Dim chk2 As CheckBox

If (runs.Table.Rows.Count > 0) Then

  pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))



  'running_percent = 0

  'running_total = 0

  Dim run As String



  For run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1



    run = GVPILCRuns_Select.Items(run_grid).Cells(2).Text



    chk2 = GVPILCRuns_Select.Items(run_grid).Cells(7).FindControl("chkselectrun")

    If chk2.Checked Then

      If Not Equals(VisitedGlbRuns.Items.FindByText(run.ToString), Nothing) Then

        Continue For
```

```
        Else
            select_runs.Add(run.ToString)
            If CheckCommitPILC.Checked Then
                Update_Attributes(run, False)
                VisitedGlbRuns.Items.Add(run)
            End If

        End If
    End If

Next

Dim j, k As Integer
For j = 1 To select_runs.Count - 1
```

```
For k = 0 To runs.Table.Rows.Count - 1

runId = runs.Item(k).Item("Run_ID")

If Not Equals(select_runs(j).ToString, runId.ToString) Then

Continue For

Else

running_total = running_total + runs.Item(k).Item("length")

End If

Next

Next

'commented on 02/18/2010

'  running_percent = running_total / pilc_sections * 100.0

'  running_tot_percent = running_total / total_sections * 100.0

'  If CheckCommitPILC.Checked Then
```

```
'      Update_Attributes(runId, False)

'          If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)

'  End If

'  visited_sections_check(runId, False)



'   'end of modification by shringika on 06/16/2008 for calling update attributes only
when commit changes box is checked

'  actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))




Else

'  'when no rows returned from table



  actualPercentPILC.Text = 0

End If

actualPercentPILC.Text = Convert.ToString(Math.Round(running_total / pilc_sections *
100))
```

```
'ovl_cnt = running_total

'Return ovl_cnt

Return running_total



End Function



Protected Sub CalculateCostRel_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostRel.Click

'--modified by shringika on 20-dec-2007 included if case

' /this one is for xlp tier

CheckCommitRel.Checked = False

CalculateCostRel_CommitChanges()



End Sub

Private Sub CalculateCostRel_CommitChanges()

visited_sections_count = 0

If cmbSearchFeeder.Items.Count > 0 Then
```

```
BindAttributes()

BindRankingLPW()

BindFOT()

' Reset_New_Attributes()

UpdateAttributesRel.Visible = True

CommitChangesRel.Visible = True

XLP.Visible = True

GridViewXLP.Visible = True

Dim MenuChoice As Integer

'added on 01/21/2009 for sectionschanged issue in xlp- dt1 and sectionschanged

Dim dt1 As DataTable

Dim sectionschanged As Integer



MenuChoice = StrategyRelDropDown.SelectedIndex



Select Case MenuChoice

  Case 1 To 3

    'XLP  modified by RTau 1/17/08
```

```
Dim dv1 As DataView = New DataView(GetXLPTable(Feeder.Text))

'added on 01/21/2009 for sectionschanged issue in xlp- dt1 and sectionschanged

dt1 = GetXLPTable(Feeder.Text)

sectionschanged = dt1.Rows.Count

Select Case MenuChoice

  Case 1

    'Sort by Cable rank

    dv1.Sort = "Feeder_Cable_RANK ASC"

  Case 2

    'Sort by Load

    dv1.Sort = "load_percentage DESC"

  Case 3

    'Sort by LoadxRank

    dv1.Sort = "LoadXrank DESC"

End Select
```

```
If CheckCommitRel.Checked Then

    SectionsChangedRel.Text =
Convert.ToString(Replace_XLP_percent(RelPercent.Text, dv1) - duplicate_sections_list)

Else

    'SectionsChangedRel.Text =
Convert.ToString(Replace_XLP_percent(RelPercent.Text, dv1) - visited_sections_count)

    SectionsChangedRel.Text =
Convert.ToString(Replace_XLP_percent(RelPercent.Text, dv1))

End If



If SectionsChangedRel.Text = "0" Then

    UpdateAttributesRel.Visible = False

    'XLP.Visible = False

End If



CommitChangesRel.Visible = True

Total_Cost_Rel()



' Code for GridView
```

```
GridViewXLP.DataSource = dv1

GridViewXLP.DataBind()



End Select



Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GridViewXLP.Rows.Count) - 1

  If Not (Equals(VisitedGlbRuns.Items.FindByText(GridViewXLP.Rows(visited_run_grid).Cells(0).Text), Nothing)) Then

    GridViewXLP.Rows(visited_run_grid).ForeColor = Drawing.Color.Red

  End If



Next

'added on 01/21/2009 for sectionschanged issue in xlp

SectionsChangedRel.Text = sectionschanged
```

```
End If

'end if added on 21-dec-2007 by shringika



End Sub



Protected Sub StrategyDropDownLoad_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles StrategyLoadDropDown.SelectedIndexChanged

'--modified by shringika on 21-dec-2007 included if case

If cmbSearchFeeder.Items.Count > 0 Then


  BindAttributes()

  BindRankingLPW()
```

```
BindFOT()

'Reset_New_Attributes()

'added by shringika on 03/11/2008 for LPW tier

'BindAttributes1()



'modification ends here 03/11/2008



If StrategyLoadDropDown.SelectedIndex = 1 Then



   ovsec.Visible = True

   ovruns.Visible = True



End If



If StrategyLoadDropDown.SelectedIndex = 2 Then



   ovsec.Visible = True
```

```
                ovruns.Visible = True


                End If

            End If

            'iend if added by shringika on 21-dec-2007

        End Sub


        Protected Sub StrategyDropDownPILC_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles StrategyPILCDropDown.SelectedIndexChanged

            TargetCostCheck.Visible = False

            actualTotalCostPILC.Text = ""

            actualPercentPILC.Text = ""

            SectionsChangedPILC.Text = ""

            TotalCostPILC.Text = ""

            RblTotalCostPercentagePILC.Items(0).Enabled = True

            downloadtoExcelPanel.Visible = False

            RblTotalCostPercentagePILC.SelectedIndex = 0
```

```
Runs.Visible = False

Runs_Select.Visible = False

PILCCablesWithStopJoints.Visible = False

UpdateAttributesPILC.Visible = False

CommitChangesPILC.Visible = False

PILCCablesWorstStopJoints.Visible = False

downloadtoExcelPanel.Visible = False

PercentPILC.SelectedIndex = 0

Reset_MTBF_Load()

Reset_MTBF_PILC()

Reset_MTBF_Rel()



If StrategyPILCDropDown.SelectedIndex = 4 Then
```

```
' RblTotalCostPercentagePILC.Items(0).Enabled = False

'RblTotalCostPercentagePILC.Items(1).Selected = True



'TotalCostPILC.Enabled = False

'PercentPILC.Enabled = True



Runs.Visible = False

downloadtoExcelPanel.Visible = False

PILCCablesWithStopJoints.Visible = False

UpdateAttributesPILC.Visible = False

CommitChangesPILC.Visible = True



PercentPILC.SelectedIndex = 0









End If
```

```
End Sub

Protected Sub StrategyDropDownRel_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles StrategyRelDropDown.SelectedIndexChanged

BindAttributes()

BindRankingLPW()

BindFOT()

UpdateAttributesRel.Visible = False



RelXLPCalculationReset()




' Reset_New_Attributes()

'added by shringika on 03/11/2008 for LPW tier

'BindAttributes1()

'modification ends here 03/11/2008

If StrategyRelDropDown.SelectedIndex = 1 Then
```

```
'elastimolds



'ELAS.Visible = True

XLP.Visible = False

'runstop.Visible = False

'may need ot have its own view...

' runstop.Visible = True




End If


If StrategyRelDropDown.SelectedIndex = 2 Then


'Replace_Stops()


'ELAS.Visible = False
```

```
                XLP.Visible = False

                'runstop.Visible = True




            End If

            If (StrategyRelDropDown.SelectedIndex >= 3) And (StrategyRelDropDown.SelectedIndex
<= 5) Then

                'Replace_XLP_percent()

                'ELAS.Visible = False

                XLP.Visible = True

                ' runstop.Visible = False



            End If
```

```
End Sub

Private Function Replace_Overloaded_Sections(ByVal secs As DataView) As Integer

  Dim i As Integer
  Dim ovl_cnt As Int32

  Dim Id As Integer

  ovl_cnt = 0

  If secs.Table.Rows.Count = 0 Then
    Return 0
  End If
```

```
Dim dr As DataRow = secs.Table.Rows(0)

'   ovl_cnt = Convert.ToString(Int32.Parse(drcheck(dr, "ovl_cnt")))



SectionsChangedLoad.Text = Convert.ToString(ovl_cnt)

Total_Cost_Load()



For i = 0 To (secs.Table.Rows.Count - 1)

   Id = secs.Table.Rows.Item(i).Item("vision_id")



   Dim visited_visionIds As ArrayList

   If Session("visited_sections_nonPILC") Is Nothing Then

      visited_visionIds = New ArrayList

   Else

      visited_visionIds = CType(Session("visited_sections_nonPILC"), ArrayList)

   End If
```

```
If visited_visionIds.Count < 1 Then visited_visionIds.Add(0)



If visited_visionIds.Contains(Id) Then Continue For

ovl_cnt += 1

If CheckCommitLoad.Checked Then

   visited_visionIds.Add(Id)

   Session("visited_sections_nonPILC") = visited_visionIds



   Update_Attributes(Id, True)

End If

Next



'vs2220: Commented out for centralized filling of SummarySections

'If Not (Equals(SectionsSummary.Items.FindByText("No Sections"), Nothing)) Then
SectionsSummary.Items.Remove("No Sections")

'added by shringika on 07/29/2008 to fix populating secions changed details on commiting
changes again

'If CheckCommitLoad.Checked Then Return ovl_cnt
```

```
'For i = 0 To (secs.Table.Rows.Count - 1)

'Id = secs.Table.Rows.Item(i).Item("vision_id")

'SectionsSummary.Items.Add(Convert.ToString(Id))



'Next



Return ovl_cnt

End Function

Private Function Replace_Overloaded_Runs(ByVal runs As DataView) As Integer



Dim ovl_cnt As Int32

Dim i As Integer

Dim runId As Integer

'coomented by shringika on 01/19/2009 for getting total no. of sections changed on 1st tier

ovl_cnt = 0
```

```
If (runs.Table.Rows.Count > 0) Then

  'commented on 01/19/2009 , included this tatement within loop with i



  'ovl_cnt = ovl_cnt + Int32.Parse(runs.Item(0).Item("ovl_cnt"))



  For i = 0 To (runs.Table.Rows.Count - 1)

    runId = runs.Item(i).Item("Run_ID")

    If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For

    ovl_cnt = ovl_cnt + Int32.Parse(runs.Item(i).Item("ovl_cnt"))

    ' Update_Attributes(runId, False)

    If CheckCommitLoad.Checked Then

      Update_Attributes(runId, False)

      VisitedGlbRuns.Items.Add(runId)



    End If
```

```
    Next

  Else : Return 0



  End If



  Return ovl_cnt




End Function

'not to be used anymore -- agb 7/30/08

Private Function Replace_Paper() As Integer

  If CheckCommitPILC.Checked Then

    mon_Paper_GT40.Text = "0"

    mon_Paper_LE40.Text = "0"

    mon_Solid_LT16.Text = Convert.ToString(Int32.Parse(Solid_LT16.Text) +
Int32.Parse(Paper_LE40.Text) + Int32.Parse(Paper_GT40.Text))

    mon_Percent_Paper_GT40.Text = "0"

    mon_Percent_Paper_LE40.Text = "0"
```

```
                mon_Percent_Solid_LT16.Text = Convert.ToString((Int32.Parse(Solid_LT16.Text) +
Int32.Parse(Paper_LE40.Text) + Int32.Parse(Paper_GT40.Text)) / Int32.Parse(Sec_CNT.Text) * 100.0)

                mon_AllPaper.Text = "0"

                mon_Percent_AllPaper.Text = "0"

                mon_E21GT93.Text = "0"

                mon_E21LE93.Text = "0"

                mon_R31GT94.Text = "0"

                mon_R31LE93.Text = "0"

                mon_OtherStop_LT7.Text = "0"

                mon_OtherStop7_12.Text = "0"

                mon_OtherStop_GT12.Text = "0"

                mon_AllStop.Text = "0"

                mon_Percent_ALLStop.Text = "0"

                mon_PJ_LT8.Text = "0"

                mon_PJ8_16.Text = "0"

                mon_PJ16_24.Text = "0"

                ' mon_PJ_GT24.Text = "0"

                mon_PJ24_32.Text = "0"
```

```
mon_PJ32_40.Text = "0"

mon_PJGT40.Text = "0"

mon_SJ_LT8.Text = Convert.ToString(Int32.Parse(SJ_LT8.Text) +
Int32.Parse(AllStop.Text) + Int32.Parse(AllPaper.Text))

mon_Percent_E21GT93.Text = "0"

mon_Percent_E21LE93.Text = "0"

mon_Percent_R31GT94.Text = "0"

mon_Percent_R31LE93.Text = "0"

mon_Percent_OtherStop_LT7.Text = "0"

mon_Percent_OtherStop7_12.Text = "0"

mon_Percent_OtherStop_GT12.Text = "0"

mon_Percent_PJ_LT8.Text = "0"

mon_Percent_PJ8_16.Text = "0"

mon_Percent_PJ16_24.Text = "0"

'mon_Percent_PJ_GT24.Text = "0"

mon_Percent_PJ24_32.Text = "0"

mon_Percent_PJ32_40.Text = "0"

mon_Percent_PJGT40.Text = "0"
```

```
                mon_Percent_SJ_LT8.Text =
Convert.ToString(Convert.ToDouble(Percent_SJ_LT8.Text) + Convert.ToDouble(Percent_ALLStop.Text) +
Convert.ToDouble(Percent_AllPaper.Text))

                mon_AllSolid.Text = Convert.ToString(Int32.Parse(mon_AllSolid.Text) +
Int32.Parse(AllStop.Text) + Int32.Parse(AllPaper.Text))

                mon_Percent_AllSolid.Text =
Convert.ToString(Convert.ToDouble(Percent_AllSolid.Text) + Convert.ToDouble(Percent_ALLStop.Text) +
Convert.ToDouble(Percent_AllPaper.Text))

                'Might want to move this to a separate sub

                'SectionsChangedPILC.Text = Convert.ToString(Int32.Parse(Paper_LE40.Text) +
Int32.Parse(Paper_GT40.Text))

                'Total_Cost_PILC()

            End If

            Return Int32.Parse(Paper_LE40.Text) + Int32.Parse(Paper_GT40.Text)

        End Function

        Private Function Replace_Paper_percent(ByVal percent As String, ByVal runs As
DataView) As Integer
```

```
Dim ovl_cnt As Int32



Dim i As Integer

Dim total_sections, pilc_sections, running_total As Integer

Dim per As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer



'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)

'If percent.Length = 0 Then percent = "100"

per = Int32.Parse(percent)



'If (per = 100) Then

'section count is from attributes in function so can be different, bummer!

'Replace_Paper()

'changed by shringika and made ==100 for 100 percent
```

```
'actualPercentPILC.Text = "100"

'If (runs.Table.Rows.Count > 0) Then



'Return Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'Else

'actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

'Return 0

'End If

'End If




If (runs.Table.Rows.Count > 0) Then

   pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))



   'running_percent = 0

   'running_total = 0
```

```
For i = 0 To (runs.Table.Rows.Count - 1)

  runId = runs.Item(i).Item("Run_ID")

  'modified on 08/15/2008 for eliminATING ACCUMULATED PERCENT
CALCULATION FOR VISITED RUNS

  If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For

  running_total = running_total + runs.Item(i).Item("length")

  running_percent = running_total / pilc_sections * 100.0

  running_tot_percent = running_total / total_sections * 100.0

  If (running_percent > per) Then

    running_total = running_total - runs.Item(i).Item("length")
```

```
running_percent = running_total / pilc_sections * 100.0

running_tot_percent = running_total / total_sections * 100.0

runId = runs.Item(i).Item("Run_ID")



actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

Exit For

End If



runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

'shringika, add commit logic here and other places this is called

If CheckCommitPILC.Checked Then



Update_Attributes(runId, False)

If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)
```

```
                End If

                visited_sections_check(runId, False)



                'end of modification by shringika on 06/16/2008 for calling update attributes only when commit changes box is checked

                actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))



            Next

        Else

            'when no rows returned from table



            actualPercentPILC.Text = 0

        End If



        ovl_cnt = running_total

        Return ovl_cnt
```

```
End Function

Private Function Replace_Paper_TotalCost(ByVal totalcost As String, ByVal runs As
DataView) As Integer

    Dim ovl_cnt As Int32

    Dim i As Integer

    Dim total_sections, pilc_sections, running_total As Integer

    Dim total_cost As Integer

    Dim running_percent, running_tot_percent As Decimal

    Dim runId As Integer

    Dim cost_per_section As Integer

    'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))

    total_sections = Int32.Parse(Sec_CNT.Text)

    total_cost = Int32.Parse(totalcost)
```

```
cost_per_section = Int32.Parse(CostPerSectionPILC.Text)

Dim total_replaced_sections As Integer
total_replaced_sections = Math.Floor(total_cost / cost_per_section)

If (runs.Table.Rows.Count > 0) Then
   pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))
   lblPILCTargetPercent.Text = pilc_sections
   For i = 0 To (runs.Table.Rows.Count - 1)

      runId = runs.Item(i).Item("Run_ID")
      If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For
      running_total = running_total + runs.Item(i).Item("length")
```

```
If (running_total > total_replaced_sections) Then
   running_total = running_total - runs.Item(i).Item("length")
   actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))
   Exit For
End If

running_percent = running_total / pilc_sections * 100.0
running_tot_percent = running_total / total_sections * 100.0

runs.Item(i).Item("tsec") = running_total
runs.Item(i).Item("psec") = Math.Round(running_percent)
runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)
'modified on 08/13/08 for visited section elimination
If CheckCommitPILC.Checked Then
   If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)
```

```
            Update_Attributes(runId, False)

        End If

        visited_sections_check(runId, False)



        actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

        If (running_total.Equals(total_replaced_sections)) Then Exit For



    Next

End If



ovl_cnt = running_total

'actualPercentPILC.Text = Math.Round(running_total / pilc_sections) * 100.0



Return ovl_cnt



'Might want to move this to a separate sub

'SectionsChangedPILC.Text = Convert.ToString(ovl_cnt)
```

```
'Total_Cost_PILC()

End Function

' end of function added by shringika on 02/14/2008



Private Function Replace_Elastimold_percent(ByVal percent As String, ByVal runs As DataView, ByVal joints As DataView) As Integer

'*** added by RTau 12/11

Dim ovl_cnt As Int32



Dim i, j As Integer

Dim total_joints, total_sections, total_estops As Integer

Dim running_total, running_sec_total As Integer

Dim per As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer

Dim visitedRuns As New ArrayList

Dim visitedJoints As New ArrayList
```

```
Dim dupcount As Integer



total_sections = Int32.Parse(Sec_CNT.Text)

'If percent.Length = 0 Then percent = "100"

per = Int32.Parse(percent)

running_sec_total = 0



If (runs.Table.Rows.Count > 0) Then

   total_estops = Int32.Parse(mon_E21LE93.Text) + Int32.Parse(mon_E21GT93.Text)

   " If total_estops = 0 Then Return 0

   total_joints = Int32.Parse(mon_Joint_CNT.Text)

   running_total = 0



   For i = 0 To (runs.Table.Rows.Count - 1)

      runId = runs.Item(i).Item("Run_ID")

      If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For
```

```
dupcount = 0

'Joints can be in more than one run! Have to check that

'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND type =
'ELASTIMOLD' AND config = '2W-1W' AND stop = 'S'"

For j = 0 To (joints.Count - 1)

  If visitedJoints.Contains(joints(j)("Joint_ID")) Then

    dupcount = dupcount + 1

  Else

    visitedJoints.Add(joints(j)("Joint_ID"))

  End If



Next



joints.RowFilter = Nothing



running_total = running_total + runs.Item(i).Item("Stops") - dupcount

running_sec_total = running_sec_total + runs.Item(i).Item("length")
```

```
running_percent = (running_total / total_estops) * 100.0

running_tot_percent = (running_total / total_joints) * 100.0



If (running_percent > per) Then

   running_total = running_total - runs.Item(i).Item("length")

   running_percent = (running_total / total_estops) * 100.0

   running_tot_percent = running_total / total_joints * 100.0

   runId = runs.Item(i).Item("Run_ID")



   'actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

   Exit For

End If



runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)
```

```
runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

'I dont think this check does anything! agb

If (Not (visitedRuns.Contains(runId))) Then

   visitedRuns.Add(runId)



   If CheckCommitElast.Checked Then



      Update_Attributes(runId, False)

      If  Equals(VisitedGlbRuns.Items.FindByText(runId.ToString),  Nothing)  Then
VisitedGlbRuns.Items.Add(runId)

   End If



End If



visited_sections_check(runId, False)
```

```
        ' If (running_percent >= per) Then Exit For


      Next

    End If



    ovl_cnt = running_sec_total


    Return ovl_cnt


End Function
```

```
Private Function Replace_Stops_percent(ByVal percent As String, ByVal runs As DataView,
ByVal joints As DataView) As Integer

    Dim ovl_cnt As Int32


    Dim i, j As Integer

    Dim total_joints, total_sections, total_stops As Integer

    Dim running_total, running_sec_total As Integer

    Dim per As Integer

    Dim running_percent, running_tot_percent As Decimal

    Dim runId As Integer

    Dim visitedRuns As New ArrayList

    Dim visitedJoints As New ArrayList

    Dim dupcount As Integer
```

```
'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)



per = Int32.Parse(percent)

running_sec_total = 0



'If (per = 100) Then

'section count is from attributes in funtion so can be different, bummer!

'sec_changed = Replace_Paper()

'If (runs.Table.Rows.Count > 0) Then

'Return Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'Return sec_changed

'Else : Return 0

'End If

'End If



If (runs.Table.Rows.Count > 0) Then
```

```
'pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'for now do it this way

'get this form base not running AGB

total_stops = Int32.Parse(AllStop.Text)

total_joints = Int32.Parse(Joint_CNT.Text)

running_total = 0



For i = 0 To (runs.Table.Rows.Count - 1)

  runId = runs.Item(i).Item("Run_ID")

  If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For

  dupcount = 0

  'Joints can be in more than one run! Have to check that

  'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

  joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND stop = 'S'"

  For j = 0 To joints.Count - 1

    If visitedJoints.Contains(joints(j)("Joint_ID")) Then
```

```
        dupcount = dupcount + 1
      Else
        visitedJoints.Add(joints(j)("Joint_ID"))
      End If

    Next

    joints.RowFilter = Nothing
    running_total = running_total + runs.Item(i).Item("Stops") - dupcount
    running_sec_total = running_sec_total + runs.Item(i).Item("length")
    running_percent = running_total / total_stops * 100.0
    running_tot_percent = running_total / total_joints * 100.0

    'do this early now -- agb
    ' If (running_percent >= per) Then Exit For
```

```
If (running_percent >= per) Then
    running_sec_total = running_sec_total - runs.Item(i).Item("length")
    running_total = running_total - (runs.Item(i).Item("Stops") - dupcount)
    running_percent = running_total / total_stops * 100.0
    running_tot_percent = running_total / total_joints * 100.0
    runId = runs.Item(i).Item("Run_ID")

    ' actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))
    Exit For

End If
```

```
runId = runs.Item(i).Item("Run_ID")
```

```
runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

'use this to debug dups

'runs.Item(i).Item("ptsec") = dupcount



'I dont think this check does anything... AGB

If (Not (visitedRuns.Contains(runId))) Then

   visitedRuns.Add(runId)

   If CheckCommitStops.Checked Then

      Update_Attributes(runId, False)

      If  Equals(VisitedGlbRuns.Items.FindByText(runId.ToString),  Nothing)  Then
VisitedGlbRuns.Items.Add(runId)

   End If

   visited_sections_check(runId, False)
```

```
            End If

            'If (running_percent.Equals(per)) Then Exit For

            'If (running_percent >= per) Then Exit For

        Next
    End If

    ovl_cnt = running_sec_total
```

```
        Return ovl_cnt



    End Function



    Private Function Replace_XLP_percent(ByVal percent As String, ByVal runs As DataView)
As Integer

        Dim i As Integer

        Dim total_sections, XLP_sections, running_total As Integer

        Dim per As Integer

        Dim running_percent, running_tot_percent As Decimal

        Dim runId As Integer



        total_sections = Int32.Parse(Sec_CNT.Text)

        'If percent.Length = 0 Then percent = "100"
```

```
per = Int32.Parse(percent)

' Sum total length of XLP sections from individual sections
For i = 0 To (runs.Table.Rows.Count - 1)
   XLP_sections = XLP_sections + runs.Item(i).Item("length")

Next

If (runs.Table.Rows.Count > 0) Then
   For i = 0 To (runs.Table.Rows.Count - 1)
      runId = runs.Item(i).Item("Run_ID")

      If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For

      running_total = running_total + runs.Item(i).Item("length")
      running_percent = running_total / XLP_sections * 100.0
      running_tot_percent = running_total / total_sections * 100.0
```

```
If (running_percent > per) Then
   running_total = running_total - runs.Item(i).Item("length")
   running_percent = running_total / XLP_sections * 100.0
   running_tot_percent = running_total / total_sections * 100.0
   runId = runs.Item(i).Item("Run_ID")

   actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))
   Exit For
End If

runs.Item(i).Item("tsec") = running_total
runs.Item(i).Item("psec") = Math.Round(running_percent)
```

```
                runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

                If CheckCommitRel.Checked Then

                  Update_Attributes(runId, False)

                  If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)

                End If

                visited_sections_check(runId, False)



              Next

            End If



            Return running_total



          End Function
```

```
Private Function GetCblCnt(ByVal Feeder As String, ByVal Str As String) As Integer

    'AGB added this to emulate Vaibhav's logic in update_attributes stroed proc

    Dim sql As String


    sql = "SELECT cableCount = count(*) FROM EdisonML.dbo.Rerank_OVL WHERE
(from_struct = '" + Str + "' OR to_struct = '" + Str + "') and feeder= '" + Feeder + "' AND 1 = (SELECT count(*)
FROM EdisonML.dbo.Rerank_OVL WHERE (from_struct = '" + Str + "' or to_struct = '" + Str + "') AND insulation
= 'paper' and feeder= '" + Feeder + "' )"


    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)


    Dim dt As DataTable = New DataTable

    myCommand.Fill(dt)

    myCommand = Nothing

    myConnection.Close()

    myConnection = Nothing

    Dim dr As DataRow = dt.Rows(0)
```

```
    Return Int32.Parse(dr("cableCount"))

  End Function



  Private Sub Update_Attributes(ByVal id As String, ByVal isSection As Boolean)

    Dim m_negativeValues As String

    m_negativeValues =
Convert.ToString(ConfigurationManager.AppSettings("negativeAttributes"))

    'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



    If Equals(m_negativeValues, "no") Then



      Update_Attributes_Fix_For_Negative_Attribute_Values(id, isSection)

    Else

      Update_Attributes__Old_With_Negative_Values(id, isSection)

    End If

  End Sub

  Private Sub Update_Attributes__Old_With_Negative_Values(ByVal id As String, ByVal
isSection As Boolean)
```

```
Dim totalJoints As Integer

Dim dr As DataRow

Dim fromtostructs As DataTable

'  Static Uattr As UpdateAttributes = New UpdateAttributes

Dim feeder_id As String

feeder_id = cmbSearchFeeder.SelectedValue

If feeder_id.Length < 5 Then feeder_id = "0" + feeder_id.ToString




If (isSection = False) Then

   'this function is more generic than the nam ovl lets on to

   'attributes = upd.GetAttributeVectors(id, False, feeder_id)



   dr = Uattr.CountMe(id, False)

   fromtostructs = Uattr.GetFromToStruct(id, False)

Else
```

```
'attributes = upd.GetAttributeVectors(id, True, feeder_id)
'
dr = Uattr.CountMe(id, True)
fromtostructs = Uattr.GetFromToStruct(id, True)

End If

Dim k As Integer
Dim dtr As DataRow

'commented on 08/16/2009 for new update attributes code

For k = 0 To (fromtostructs.Rows.Count - 1)
dtr = fromtostructs.Rows(k)
If IsDBNull(dtr("feeder")) = True And IsDBNull(dtr("from_struct")) = True And IsDBNull(dtr("to_struct")) = True Then
'Exit Sub
Continue For
```

```
End If

If (Equals(SectionsSummary.Items.FindByText(dtr("feeder") + "." + dtr("from_struct")
+ "." + dtr("to_struct")), Nothing)) Then

SectionsSummary.Items.Add(dtr("feeder") + "." + dtr("from_struct") + "." +
dtr("to_struct"))

sections_area.InnerText += System.Environment.NewLine + dtr("feeder") + "." +
dtr("from_struct") + "." + dtr("to_struct")

'Else

'commented on 01/27/2009

'duplicate_sections_list = duplicate_sections_list + 1

' duplicate_sections_list = 0

End If

Next

'mean imputation stats fon canonical feeders

'Insulationcnt avg_date

'NULL 256 1979
```

```
'XLP                    45099   1983

'PAPER          24976           1959

'EPR                    58125   1999

'commented on 08/16/2009 for new update attributes code

'Dim k As Integer

'Dim dr As DataRow



'For k = 0 To (attributes.Rows.Count - 1)

'dr = attributes.Rows(k)

'If IsDBNull(dr("feeder")) = True And IsDBNull(dr("from_struct")) = True And IsDBNull(dr("to_struct")) = True Then Exit Sub

'  Continue For

' End If



' If (Equals(SectionsSummary.Items.FindByText(dr("feeder") + "." + dr("from_struct") + "." + dr("to_struct")), Nothing)) Then

'SectionsSummary.Items.Add(dr("feeder") + "." + dr("from_struct") + "." + dr("to_struct"))

'sections_area.InnerText += System.Environment.NewLine + dr("feeder") + "." + dr("from_struct") + "." + dr("to_struct")
```

```
' Else

'commented on 01/27/2009

'duplicate_sections_list = duplicate_sections_list + 1

'  duplicate_sections_list = 0



'  End If

'modifiction for update_attribute ends here



'Updating Counts for cable sections

mon_Paper_GT40.Text = Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) -
Int32.Parse(dr("iPaperGT40")))

mon_Paper_LE40.Text = Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) -
Int32.Parse(dr("iPaperLE40")))

'dont need to add and subtract iSolidLT16 agb

mon_Solid_LT16.Text = Convert.ToString(Int32.Parse(mon_Solid_LT16.Text) +
Int32.Parse(dr("iPaperGT40")) + Int32.Parse(dr("iPaperLE40")) + Int32.Parse(dr("isolidGE16")) +
Int32.Parse(dr("iSolid70_75")))

mon_Solid_GT16.Text = Convert.ToString(Int32.Parse(mon_Solid_GT16.Text) -
Int32.Parse(dr("isolidGE16")))
```

```
mon_Solid_70_75.Text = Convert.ToString(Int32.Parse(mon_Solid_70_75.Text) -
Int32.Parse(dr("iSolid70_75")))

'Updating Counts for joints

'Elastimold Stop - just remove, don't replace

mon_E21GT93.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) -
Int32.Parse(dr("elasGT1993")))

mon_E21LE93.Text = Convert.ToString(Int32.Parse(mon_E21LE93.Text) -
Int32.Parse(dr("elasLE1993")))

'Raychem Stop - just remove, don't replace

mon_R31GT94.Text = Convert.ToString(Int32.Parse(mon_R31GT94.Text) -
Int32.Parse(dr("rayGT1993")))

mon_R31LE93.Text = Convert.ToString(Int32.Parse(mon_R31LE93.Text) -
Int32.Parse(dr("rayLE1993")))

'Other Stop - just remove, don't replace

mon_OtherStop_LT7.Text = Convert.ToString(Int32.Parse(mon_OtherStop_LT7.Text) -
Int32.Parse(dr("oStopLT7")))
```

```
mon_OtherStop7_12.Text = Convert.ToString(Int32.Parse(mon_OtherStop7_12.Text) -
Int32.Parse(dr("oStopBet7A12")))

mon_OtherStop_GT12.Text = Convert.ToString(Int32.Parse(mon_OtherStop_GT12.Text)
- Int32.Parse(dr("oStopGT12")))



'Updating total stops

mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) +
Int32.Parse(mon_E21LE93.Text) + Int32.Parse(mon_R31GT94.Text) + Int32.Parse(mon_R31LE93.Text) +
Int32.Parse(mon_OtherStop_LT7.Text) + Int32.Parse(mon_OtherStop7_12.Text) +
Int32.Parse(mon_OtherStop_GT12.Text))



'Paper - replace

mon_PJ_LT8.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) -
Int32.Parse(dr("papLT8")))

mon_PJ8_16.Text = Convert.ToString(Int32.Parse(mon_PJ8_16.Text) -
Int32.Parse(dr("papBet8A16")))

mon_PJ16_24.Text = Convert.ToString(Int32.Parse(mon_PJ16_24.Text) -
Int32.Parse(dr("papBet16A24")))

mon_PJ24_32.Text = Convert.ToString(Int32.Parse(mon_PJ24_32.Text) -
Int32.Parse(dr("papBet24A32")))
```

```
mon_PJ32_40.Text = Convert.ToString(Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40")))

mon_PJGT40.Text = Convert.ToString(Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40")))

'Updating total paper

mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) + Int32.Parse(mon_PJ8_16.Text) + Int32.Parse(mon_PJ16_24.Text) + Int32.Parse(mon_PJ24_32.Text) + Int32.Parse(mon_PJ32_40.Text) + Int32.Parse(mon_PJGT40.Text))

'Solid - update date

mon_SJ8_16.Text = Convert.ToString(Int32.Parse(mon_SJ8_16.Text) - Int32.Parse(dr("solBet8A16")))

mon_SJ16_24.Text = Convert.ToString(Int32.Parse(mon_SJ16_24.Text) - Int32.Parse(dr("solBet16A24")))

mon_SJ_GT24.Text = Convert.ToString(Int32.Parse(mon_SJ_GT24.Text) - Int32.Parse(dr("solGT24")))

'Finally, adding all to be replaced and date updated to solid
```

```
mon_SJ_LT8.Text = Convert.ToString(Int32.Parse(mon_SJ_LT8.Text) + Int32.Parse(dr("papLT8")) + Int32.Parse(dr("papBet8A16")) + Int32.Parse(dr("papBet16A24")) + Int32.Parse(dr("papBet24A32")) + Int32.Parse(dr("papBet32A40")) + Int32.Parse(dr("papGE40")) + Int32.Parse(dr("solBet8A16")) + Int32.Parse(dr("solBet16A24")) + Int32.Parse(dr("solGT24")) + Int32.Parse(dr("leadWipeCount")) + Int32.Parse(dr("raychem1w1w")) + Int32.Parse(dr("raychem3w1w")) + Int32.Parse(dr("other1w1w")))

' Next

'Updating Section cost text boxes

SectionsChangedSum.Text = SectionsSummary.Items.Count

CostPerSectionSum.Text = "18000"

TotalCostSum.Text = SectionsSummary.Items.Count * 18000

'Updating percentages for cable sections

mon_Percent_Paper_GT40.Text = Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) / Int32.Parse(Sec_CNT.Text) * 100.0)

mon_Percent_Paper_LE40.Text = Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) / Int32.Parse(Sec_CNT.Text) * 100.0)
```

```
mon_Percent_Solid_LT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_LT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_GT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_GT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_70_75.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_70_75.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))



'Updating total solids

mon_AllSolid.Text = Convert.ToString(Int32.Parse(mon_SJ8_16.Text) +
Int32.Parse(mon_SJ16_24.Text) + Int32.Parse(mon_SJ_GT24.Text) + Int32.Parse(mon_SJ_LT8.Text))



'updating total joints

mon_Joint_CNT.Text = Convert.ToString(Int32.Parse(mon_AllStop.Text) +
Int32.Parse(mon_AllPaper.Text) + Int32.Parse(mon_AllSolid.Text))



totalJoints = mon_Joint_CNT.Text
```

```
'Updating percentages

'Elastimold Stop Percentages

mon_Percent_E21GT93.Text = Convert.ToString((Int32.Parse(mon_E21GT93.Text) /
totalJoints) * 100)

mon_Percent_E21LE93.Text = Convert.ToString((Int32.Parse(mon_E21LE93.Text) /
totalJoints) * 100)

'Raychem Stop percentages

mon_Percent_R31GT94.Text = Convert.ToString((Int32.Parse(mon_R31GT94.Text) /
totalJoints) * 100)

mon_Percent_R31LE93.Text = Convert.ToString((Int32.Parse(mon_R31LE93.Text) /
totalJoints) * 100)

'Other Stop percentages

mon_Percent_OtherStop_LT7.Text =
Convert.ToString((Int32.Parse(mon_OtherStop_LT7.Text) / totalJoints) * 100)

mon_Percent_OtherStop7_12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop7_12.Text) / totalJoints) * 100)
```

```
                mon_Percent_OtherStop_GT12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop_GT12.Text) / totalJoints) * 100)


                'Updating total stop percentages

                mon_Percent_ALLStop.Text = Convert.ToString((Int32.Parse(mon_AllStop.Text) /
totalJoints) * 100)


                'Paper percentages

                mon_Percent_PJ_LT8.Text = Convert.ToString((Int32.Parse(mon_PJ_LT8.Text) /
totalJoints) * 100)

                mon_Percent_PJ8_16.Text = Convert.ToString((Int32.Parse(mon_PJ8_16.Text) /
totalJoints) * 100)

                mon_Percent_PJ16_24.Text = Convert.ToString((Int32.Parse(mon_PJ16_24.Text) /
totalJoints) * 100)

                mon_Percent_PJ24_32.Text = Convert.ToString((Int32.Parse(mon_PJ24_32.Text) /
totalJoints) * 100)

                mon_Percent_PJ32_40.Text = Convert.ToString((Int32.Parse(mon_PJ32_40.Text) /
totalJoints) * 100)

                mon_Percent_PJGT40.Text = Convert.ToString((Int32.Parse(mon_PJGT40.Text) /
totalJoints) * 100)
```

```
'Updating total paper percentages

mon_Percent_AllPaper.Text = Convert.ToString((Int32.Parse(mon_AllPaper.Text) /
totalJoints) * 100)

'Solid percentages

mon_Percent_SJ_LT8.Text = Convert.ToString((Int32.Parse(mon_SJ_LT8.Text) /
totalJoints) * 100)

mon_Percent_SJ8_16.Text = Convert.ToString((Int32.Parse(mon_SJ8_16.Text) /
totalJoints) * 100)

mon_Percent_SJ16_24.Text = Convert.ToString((Int32.Parse(mon_SJ16_24.Text) /
totalJoints) * 100)

mon_Percent_SJ_GT24.Text = Convert.ToString((Int32.Parse(mon_SJ_GT24.Text) /
totalJoints) * 100)

'Updating total solids

mon_Percent_AllSolid.Text = Convert.ToString((Int32.Parse(mon_AllSolid.Text) /
totalJoints) * 100)

End Sub
```

```
Private Sub Reset_MTBFs()

    StrategyLoadDropDown.SelectedIndex = 0

    StrategyPILCDropDown.SelectedIndex = 0

    StrategyRelDropDown.SelectedIndex = 0

    RelPercentStops.SelectedIndex = 0

    RelPercentElastimold.SelectedIndex = 0

    RelPercent.SelectedIndex = 0

    SectionsChangedLoad.Text = ""

    SectionsChangedPILC.Text = ""

    SectionsChangedRelStops.Text = ""

    SectionsChangedRelElastimold.Text = ""

    SectionsChangedRel.Text = ""

    actualPercentPILC.Text = ""

    actualTotalCostPILC.Text = ""



    Reset_MTBF_Load()
```

```
    Reset_MTBF_PILC()

    Reset_MTBF_Rel()

        Reset_MTBF_LPW()

End Sub



Private Sub Reset_MTBF_Load()

    RankBaseLoad.Text = ""

    FOTBaseLoad.Text = ""

    MTBFBaseLoad.Text = ""

    RankNewLoad.Text = ""

    RankChangeLoad.Text = ""

    MTBFNewLoad.Text = ""

    MTBFChangeLoad.Text = ""

    FOTNewLoad.Text = ""

    FOTChangeLoad.Text = ""

    TotalCostLoad.Text = ""

    loadStrategyInfo.Visible = False
```

```
End Sub

Private Sub Reset_MTBF_PILC()

    RankBasePILC.Text = ""

    FOTBasePILC.Text = ""

    MTBFBasePILC.Text = ""

    RankNewPILC.Text = ""

    RankChangePILC.Text = ""

    MTBFNewPILC.Text = ""

    MTBFChangePILC.Text = ""

    FOTNewPILC.Text = ""

    FOTChangePILC.Text = ""

    'PercentPILC.SelectedIndex = 0

    'TotalCostPILC.Text = ""

    PILCMTBFInfo.Visible = False

    Panel9.Visible = False

    PILCMTBFTotalCostInfo.Visible = False
```

```
    LoadandPILCPanel.Visible = False

    sectionschangedPILCPanel.Visible = False




End Sub



Private Sub Reset_MTBF_Rel()

    'for xlp tier

    RankBaseRel.Text = ""

    FOTBaseRel.Text = ""

    MTBFBaseRel.Text = ""

    RankNewRel.Text = ""

    RankChangeRel.Text = ""

    MTBFNewRel.Text = ""

    MTBFChangeRel.Text = ""

    FOTNewRel.Text = ""
```

```
FOTChangeRel.Text = ""

'for stopjoints tier

RankBaseRelStops.Text = ""

FOTBaseRelStops.Text = ""

MTBFBaseRelStops.Text = ""

RankNewRelStops.Text = ""

RankChangeRelStops.Text = ""

MTBFNewRelStops.Text = ""

MTBFChangeRelStops.Text = ""

FOTNewRelStops.Text = ""

FOTChangeRelStops.Text = ""

'for elastimold iter

RankBaseRelElast.Text = ""

FOTBaseRelElast.Text = ""

MTBFBaseRelElast.Text = ""

RankNewRelElast.Text = ""

RankChangeRelElast.Text = ""

MTBFNewRelElast.Text = ""
```

```
    MTBFChangeRelElast.Text = ""

    FOTNewRelElast.Text = ""

    FOTChangeRelElast.Text = ""




End Sub



Private Sub Reset_MTBF_LPW()

    RankBaseLPW.Text = ""

    FOTBaseLPW.Text = ""

    MTBFBaseLPW.Text = ""

    RankNewLPW.Text = ""

    RankChangeLPW.Text = ""

    MTBFNewLPW.Text = ""

    MTBFChangeLPW.Text = ""

    FOTNewLPW.Text = ""

    FOTChangeLPW.Text = ""
```

```
End Sub

Public Function rand(ByVal low As Int32, ByVal high As Int32) As Integer

   Static RandomNumGen As New System.Random

   Return RandomNumGen.Next(low, high + 1)

End Function




Private Sub Calculate_New_Load_Runs()

   'modified by shringika on dec 21'2007 included if case

   If cmbSearchFeeder.Items.Count > 0 Then



      If SectionsChangedLoad.Text = "" Then



         'MTBF_NA.Visible = True

         Return

      End If
```

```
loadStrategyInfo.Visible = True

Dim feeder As String

feeder = cmbSearchFeeder.SelectedValue

If feeder.Length < 5 Then feeder = "0" + feeder

Dim pct As Integer

If StrategyLoadDropDown.SelectedIndex = 2 Then

  pct = 100

ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then

  pct = 105

Else

  pct = 110

End If



'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score_CAPTV10_RV1_2008] WHERE [Feeder] = '" & feeder & "' and load_relief_percent = 100 and pilc_percent = 0"
```

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score_CAPTV10_RV1_2008] WHERE [Feeder] = '" & feeder & "' and load_relief_percent =" & pct & " and pilc_percent = 0"

Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [Feeder] = '" & feeder & "' and load_relief_percent =" & pct & " and pilc_percent = 0"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and load_relief_percent = 100 and pilc_percent = 0"

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

```
Dim Rank As Integer

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_LR_NA.Visible = True

   Reset_MTBF_Load()

   Return

End If




Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary




RankBaseLoad.Text = dr("OldRank")

Rank = Int32.Parse(dr("OldRank"))
```

```
'conversion  of rank to MTBF using regression (New 6-08)

' MTBFBaseLoad.Text = (Rank * 0.3386 + 165.96).ToString("F0")

MTBFBaseLoad.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

'MTBFBaseLoad.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseLoad.Text = Convert.ToString(Math.Round(Math.Max((Rank * -0.0095 + 6),
0.0), 0))



'TEMP!!!!!!!!!!!!!

'RankNewLoad.Text = Convert.ToString(rand(Rank, Rank + 100))


RankNewLoad.Text = dr("rank")

'RankNewPILC.Text = "930"


RankChangeLoad.Text = Convert.ToString(Int32.Parse(RankNewLoad.Text) - Rank)

'conversion  of rank to MTBF using regression

MTBFNewLoad.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")
```

```
                MTBFChangeLoad.Text = Convert.ToDouble(dr("delta")).ToString("F0")

                'MTBFNewLoad.Text = (Int32.Parse(RankNewLoad.Text) * 0.3386 +
165.96).ToString("F0")

                ' MTBFChangeLoad.Text =
Convert.ToString(Convert.ToDouble(MTBFNewLoad.Text) - Convert.ToDouble(MTBFBaseLoad.Text))

                FOTNewLoad.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewLoad.Text) * -0.0095 + 6), 0.0), 0))

                FOTChangeLoad.Text = Convert.ToString(Convert.ToDouble(FOTBaseLoad.Text) -
Convert.ToDouble(FOTNewLoad.Text))

                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If

        End Sub
```

```
Private Sub Calculate_New_Load()

'not used now since we do RUNS not sections



'modified by shringika on dec 21'2007 included if case

If cmbSearchFeeder.Items.Count > 0 Then



Dim sql As String = "SELECT * FROM ediprodb2.FeederSuscept.dbo.WhatIf_Ranking
WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("DB2FeederSuscept").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)



Dim dt As DataTable = New DataTable



myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)
```

```
'ABOVE is only temporary

RankBaseLoad.Text = dr("real_rank")

'delete line below when webservice is working



'conversion  of rank to MTBF using regression

MTBFBaseLoad.Text = (Int32.Parse(RankBaseLoad.Text) * 0.311 +
150.0).ToString("F0")

FOTBaseLoad.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankBaseLoad.Text) * -0.0095 + 6), 0.0), 0))



RankNewLoad.Text = "890"

RankChangeLoad.Text = Convert.ToString(Int32.Parse(RankNewLoad.Text) -
Int32.Parse(RankBaseLoad.Text))

'conversion  of rank to MTBF using regression

MTBFNewLoad.Text = (Int32.Parse(RankNewLoad.Text) * 0.311 +
150.0).ToString("F0")



'MTBFNewLoad.Text = Convert.ToString(Int32.Parse(RankNewLoad.Text) * 0.311 +
150.0)
```

```
                MTBFChangeLoad.Text = Convert.ToString(Convert.ToDouble(MTBFNewLoad.Text)
- Convert.ToDouble(MTBFBaseLoad.Text))

                FOTNewLoad.Text = Convert.ToString(Math.Max((Int32.Parse(RankNewLoad.Text) *
-0.0095 + 6), 0.0))

                FOTChangeLoad.Text = Convert.ToString(Convert.ToDouble(FOTBaseLoad.Text) -
Convert.ToDouble(FOTNewLoad.Text))



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If

        End Sub



        Private Sub Calculate_New_PILC()



            sectionschangedPILCPanel.Visible = True



            If RblTotalCostPercentagePILC.SelectedIndex > 1 Then

                sectionschangedPILCPanel.Visible = False
```

```
Dim feeder_Id As String

feeder_Id = New String("X")

feeder_Id = cmbSearchFeeder.SelectedValue

If feeder_Id.Length < 5 Then

   feeder_Id = "0" + feeder_Id



End If



Dim data_new_rank, data_base_rank, base_rank As String

data_base_rank = "feeder=" + feeder_Id.ToString + ";"

data_base_rank += "Static_PaperCable_GT_40 =" + Paper_GT40.Text + ";" +
"Static_PaperCable_LE_40 =" + Paper_LE40.Text + ";" + "Static_SolidCable_LT_16 =" + Solid_LT16.Text +
";"

data_base_rank += "Static_SolidCable_70_75 =" + Solid_70_75.Text + ";" +
"Static_SolidCable_GE_16 =" + Solid_GT16.Text + ";" + "Static_Sections_Total=" + Sec_CNT.Text + ";"

data_base_rank += "Static_E2_1_GT_1993 =" + E21GT93.Text + ";" +
"Static_E2_1_LE_1993 =" + E21LE93.Text + ";" + "Static_R3_1_GT_1993=" + R31GT94.Text + ";"
```

```
data_base_rank += "Static_R3_1_LE_1993 =" + R31LE93.Text + ";" + "Static_Other_Stops_LT_7 =" + OtherStop_LT7.Text + ";" + "Static_Other_Stops_7_12 =" + OtherStop7_12.Text + ";"

data_base_rank += "Static_Other_Stops_GT_12 =" + OtherStop_GT12.Text + ";" + "Static_Stops_Total =" + AllStop.Text + ";" + "Static_PaperJoints_LT_8=" + PJ_LT8.Text + ";"

data_base_rank += "Static_PaperJoints_8_16=" + PJ8_16.Text + ";" + "Static_PaperJoints_16_24 =" + PJ16_24.Text + ";" + "Static_PaperJoints_24_32 =" + PJ24_32.Text + ";"

data_base_rank += "Static_PaperJoints_32_40 =" + PJ32_40.Text + ";" + "Static_PaperJoints_GE_40 =" + PJGT40.Text + ";" + "Static_PaperJoints_Total =" + AllPaper.Text + ";"

data_base_rank += "Static_SolidJoints_LT_8 =" + SJ_LT8.Text + ";" + "Static_SolidJoints_8_16 =" + SJ8_16.Text + ";" + "Static_SolidJoints_16_24 =" + SJ16_24.Text + ";" + "Static_SolidJoints_GT_24 =" + SJ_GT24.Text + ";" + "Static_SolidJoints_Total=" + AllSolid.Text


base_rank =
WebserviceClient.WebserviceClient.proxy_rerank(data_base_rank.ToString)

Dim i As Integer

i = base_rank.IndexOf(",")


RankBasePILC.Text = base_rank.Substring(i + 1)
```

```
Dim score As Double

score = Double.Parse(base_rank.Substring(0, i - 1))



' MTBFBasePILC.Text = -(112.33 * score) + 278.71

'MTBFBasePILC.Text = (1.1638 * score) + 1.6793



Dim pilcmtbf_base As Integer



pilcmtbf_base = Math.Round(1.1733 * score) + 1.6807



If pilcmtbf_base = 0 Then

   MTBFBasePILC.Text = 4999

Else

   MTBFBasePILC.Text = 1 / Math.Round(1.1733 * score + 1.6807, 3) * 365



End If

' MTBFBasePILC.Text = (1.1733 * score) + 1.6807
```

```
data_new_rank = "feeder=" + feeder_Id.ToString + ";"

data_new_rank += "Static_PaperCable_GT_40 =" + mon_Paper_GT40.Text + ";" +
"Static_PaperCable_LE_40 =" + mon_Paper_LE40.Text + ";" + "Static_SolidCable_LT_16 =" +
mon_Solid_LT16.Text + ";"

data_new_rank += "Static_SolidCable_70_75 =" + mon_Solid_70_75.Text + ";"
+ "Static_SolidCable_GE_16 =" + mon_Solid_GT16.Text + ";" + "Static_Sections_Total=" + mon_Sec_CNT.Text
+ ";"

data_new_rank += "Static_E2_1_GT_1993 =" + mon_E21GT93.Text + ";" +
"Static_E2_1_LE_1993 =" + mon_E21LE93.Text + ";" + "Static_R3_1_GT_1993=" + mon_R31GT94.Text + ";"

data_new_rank += "Static_R3_1_LE_1993 =" + mon_R31LE93.Text + ";" +
"Static_Other_Stops_LT_7 =" + mon_OtherStop_LT7.Text + ";" + "Static_Other_Stops_7_12 =" +
mon_OtherStop7_12.Text + ";"

data_new_rank += "Static_Other_Stops_GT_12 =" + mon_OtherStop_GT12.Text + ";"
+ "Static_Stops_Total =" + mon_AllStop.Text + ";" + "Static_PaperJoints_LT_8=" + mon_PJ_LT8.Text + ";"

data_new_rank += "Static_PaperJoints_8_16=" + mon_PJ8_16.Text + ";" +
"Static_PaperJoints_16_24 =" + mon_PJ16_24.Text + ";" + "Static_PaperJoints_24_32 =" + mon_PJ24_32.Text +
";"

data_new_rank += "Static_PaperJoints_32_40 =" + mon_PJ32_40.Text + ";" +
"Static_PaperJoints_GE_40 =" + mon_PJGT40.Text + ";" + "Static_PaperJoints_Total =" + mon_AllPaper.Text +
";"
```

data_new_rank += "Static_SolidJoints_LT_8 =" + mon_SJ_LT8.Text + ";" + "Static_SolidJoints_8_16 =" + mon_SJ8_16.Text + ";" + "Static_SolidJoints_16_24 =" + mon_SJ16_24.Text + ";" + "Static_SolidJoints_GT_24 =" + mon_SJ_GT24.Text + ";" + "Static_SolidJoints_Total=" + mon_AllSolid.Text

Dim new_rank As String

new_rank = WebserviceClient.WebserviceClient.proxy_rerank(data_new_rank.ToString)

Dim j As Integer

j = new_rank.IndexOf(",")

Dim score_new As Double

score_new = Double.Parse(new_rank.Substring(0, j - 1))

RankNewPILC.Text = new_rank.Substring(j + 1)

RankChangePILC.Text = Convert.ToString(Int32.Parse(RankNewPILC.Text) - Int32.Parse(RankBasePILC.Text))

```
' MTBFNewPILC.Text = -(112.33 * score_new) + 278.71

'MTBFNewPILC.Text = (1.1638 * score_new) + 1.6793

'MTBFNewPILC.Text = (1.1733 * score_new) + 1.6807



Dim pilcmtbf_new As Integer



pilcmtbf_new = Math.Round(1.1733 * score_new) + 1.6807



If pilcmtbf_new = 0 Then

   MTBFNewPILC.Text = 4999

Else

   MTBFNewPILC.Text = 1 / Math.Round(1.1733 * score_new + 1.6807, 3) * 365



End If
```

```
MTBFNewPILC.Text = Math.Round(Convert.ToDouble(MTBFNewPILC.Text))

MTBFChangePILC.Text = Convert.ToString(Convert.ToDouble(MTBFNewPILC.Text)
- Convert.ToDouble(MTBFBasePILC.Text))

MTBFBasePILC.Text = Math.Round(Convert.ToDouble(MTBFBasePILC.Text))

MTBFChangePILC.Text = Math.Round(Convert.ToDouble(MTBFChangePILC.Text))

FOTBasePILC.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankBasePILC.Text) * -0.0095 + 6), 0.0), 0))

FOTNewPILC.Text = Convert.ToString(Math.Max((Int32.Parse(RankNewPILC.Text) *
-0.0095 + 6), 0.0))

FOTChangePILC.Text = Convert.ToString(Convert.ToDouble(FOTBasePILC.Text) -
Convert.ToDouble(FOTNewPILC.Text))


FOTBasePILC.Text = Math.Round(Convert.ToDouble(FOTBasePILC.Text))

FOTChangePILC.Text = Math.Round(Convert.ToDouble(FOTChangePILC.Text))

FOTNewPILC.Text = Math.Round(Convert.ToDouble(FOTNewPILC.Text))


selectedStrategy.Text = StrategyPILCDropDown.SelectedItem.Text

sectionsChangedValue.Text = SectionsChangedPILC.Text

totalCostValue.Text = actualTotalCostPILC.Text
```

```
    Panel9.Visible = True




    Exit Sub



End If




Dim cost_percent As Integer

Dim sql As String

lblCostPercentage.Text = PercentPILC.Text

lblCostPercentageValue.Text = actualPercentPILC.Text

'modified by shringika on dec 21'2007 included if case

If cmbSearchFeeder.Items.Count > 0 Then

    MTBF_NA.Visible = False

    If actualPercentPILC.Text = "" Then

        MTBF_NA.Visible = True
```

```
        Return

    End If

    Dim load_strategy As String


    If Session("strategies_load") Is Nothing Then Session("strategies_load") = "0"

    load_strategy = CType(Session("strategies_load"), String)

    sectionschangedPILCPanel.Visible = True


    lblSectionsChangedPILCMTBF.Text = SectionsChangedPILC.Text

    'feeder  network pilc_strategy pilc_percent cost delta NewMTBF       OldMTBF
OldScore OldRank score rank       versionf

    If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

      ' lblTarget.Text = "For Target Percentage "

      lblCostPercentage.Text = PercentPILC.Text


      ' lblStrategy.Text = "and strategy = "

      lblSelectedStrategy.Text = StrategyPILCDropDown.SelectedItem.Text

      ' lblAchieved.Text = " achieved percentage was :"
```

```
PILCMTBFInfo.Visible = True

lblCostPercentageValue.Text = actualPercentPILC.Text

Dim pct As Integer

If StrategyLoadDropDown.SelectedIndex = 2 Then

  pct = 100

ElseIf StrategyLoadDropDown.SelectedIndex = 3 Then

  pct = 105

Else

  pct = 110

End If



If load_strategy.Length > 2 Then

  sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent ='" & PercentPILC.Text & "' and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent = " & pct & ""

  LoadandPILCPanel.Visible = True



  lblLoadPercentagePILCMTBF.Text = StrategyLoadDropDown.SelectedValue
```

```
Else

    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent ='" & PercentPILC.Text & "' and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent= 0"

    LoadandPILCPanel.Visible = False

End If


Else

    PILCMTBFTotalCostInfo.Visible = True

    If Int32.Parse(actualPercentPILC.Text) < 5 Then

      cost_percent = 5

    ElseIf Int32.Parse(actualPercentPILC.Text) < 10 Then

      cost_percent = 10

    ElseIf Int32.Parse(actualPercentPILC.Text) < 15 Then

      cost_percent = 15

    ElseIf Int32.Parse(actualPercentPILC.Text) < 20 Then
```

```
            cost_percent = 20
        ElseIf Int32.Parse(actualPercentPILC.Text) < 25 Then
            cost_percent = 25
        ElseIf Int32.Parse(actualPercentPILC.Text) < 50 Then
            cost_percent = 50
        ElseIf Int32.Parse(actualPercentPILC.Text) < 75 Then
            cost_percent = 75
        ElseIf Int32.Parse(actualPercentPILC.Text) < 90 Then
            cost_percent = 90
        ElseIf Int32.Parse(actualPercentPILC.Text) < 95 Then
            cost_percent = 95
        Else
            cost_percent = 100
        End If
        If load_strategy.Length > 0 Then
            sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent =" & cost_percent & " and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent= 100"
```

```
                LoadandPILCPanel.Visible = True




            Else

                LoadandPILCPanel.Visible = False

                sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' and pilc_percent ='" & cost_percent & "' and pilc_strategy = '" &
StrategyPILCDropDown.SelectedValue & "' and load_relief_percent= 0"

            End If




            'modified on 07/14/2008 for load_relief_percent=100

            ' Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] =
'" & cmbSearchFeeder.SelectedValue & "' and load_relief_percent = 100 and pilc_percent ='" & PercentPILC.Text
& "'"



            lblTotalCostPILC.Text = TotalCostPILC.Text

            lblactualCostPILC.Text = actualTotalCostPILC.Text
```

```
lblSelectedStrategyTotalCost.Text = StrategyPILCDropDown.SelectedItem.Text

lblPILCActualPercent.Text = actualPercentPILC.Text



Dim total_sections, total_cost, cost_per_section As Integer

total_sections = Int32.Parse(Sec_CNT.Text)

total_cost = Int32.Parse(TotalCostPILC.Text)

cost_per_section = Int32.Parse(CostPerSectionPILC.Text)



Dim total_replaced_sections, target_percent, pilc_sections As Integer

pilc_sections = lblPILCTargetPercent.Text

total_replaced_sections = Math.Floor(total_cost / cost_per_section)

target_percent = Math.Floor((total_replaced_sections / pilc_sections) * 100)

lblPILCTargetPercent.Text = target_percent.ToString



End If
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

Dim rank As Integer

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_NA.Visible = True

   Return

End If
```

```
Dim dr As DataRow = dt.Rows(0)

RankBasePILC.Text = dr("OldRank")

rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression

MTBFBasePILC.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBasePILC.Text = Convert.ToString(Math.Round(Math.Max((rank * -0.0095 + 6),
0.0), 0))

RankNewPILC.Text = dr("rank")

'RankNewPILC.Text = "930"

RankChangePILC.Text = Convert.ToString(Int32.Parse(RankNewPILC.Text) - rank)

'conversion  of rank to MTBF using regression

MTBFNewPILC.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")
```

```
                    MTBFChangePILC.Text = Convert.ToDouble(dr("delta")).ToString("F0")


                    FOTNewPILC.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewPILC.Text) * -0.0095 + 6), 0.0), 0))

                    FOTChangePILC.Text = Convert.ToString(Convert.ToDouble(FOTBasePILC.Text) -
Convert.ToDouble(FOTNewPILC.Text))

                    lblCostPercentage.Visible = True

                    lblCostPercentageValue.Visible = True

                    ' If RankBasePILC.Text = "" Or MTBFBasePILC.Text = "" Or FOTBasePILC.Text = ""
Or RankNewPILC.Text = "" Or MTBFNewPILC.Text = "" Or FOTNewPILC.Text = 0 Then

                    'MTBF_NA.Visible = True

                    'RankBasePILC.Text = ""

                    ' MTBFBasePILC.Text = ""

                    ' FOTBasePILC.Text = ""

                    ' RankNewPILC.Text = ""

                    ' MTBFNewPILC.Text = ""

                    ' FOTNewPILC.Text = ""
```

```
        'End If


        If MTBFChangePILC.Text.Length = 0 Then
          MTBF_NA.Visible = True
        Else
          MTBF_NA.Visible = False


        End If
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
      End If
    End Sub
```

```
Private Sub Calculate_New_Rel()

    'modified by shringika on dec 21'2007 included if case

    If cmbSearchFeeder.Items.Count > 0 Then

        If RelPercent.Text = "" Then

            MTBF_Rel_NA.Visible = True

            Return

        End If



        Dim dt As DataTable = New DataTable

        Dim Rank As Integer

        'have to work on this query more



        'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '"
& cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"
```

```
Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and rel_percent = '" & RelPercent.Text & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_Rel_NA.Visible = True

   Return

End If



Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary



RankBaseRel.Text = dr("OldRank")
```

```
Rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression (New 6-08)

'modified on 01/04/2009 for picking up rank from rerank_cost_score

'MTBFBaseRel.Text = (Rank * 0.3386 + 165.96).ToString("F0")



MTBFBaseRel.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseRel.Text  =  Convert.ToString(Math.Round(Math.Max((Rank  *  -0.0095  +  6),
0.0), 0))



'TEMP!!!!!!!!!!!!!



' RankNewRel.Text = Convert.ToString(rand(Rank, Rank + 100))

'modified on 01/04/2009 for picking up rank from rerank_cost_score

RankNewRel.Text = dr("rank")

RankChangeRel.Text = Convert.ToString(Int32.Parse(RankNewRel.Text) - Rank)

'conversion  of rank to MTBF using regression

MTBFNewRel.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")
```

```
MTBFChangeRel.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'added on 01/04/2009 for picking up new mtbf and delta from database , not by calculation

' MTBFNewRel.Text = (Int32.Parse(RankNewRel.Text) * 0.3386 + 165.96).ToString("F0")

'MTBFChangeRel.Text = Convert.ToString(Convert.ToDouble(MTBFNewRel.Text) - Convert.ToDouble(MTBFBaseRel.Text))

FOTNewRel.Text = Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewRel.Text) * -0.0095 + 6), 0.0), 0))

FOTChangeRel.Text = Convert.ToString(Convert.ToDouble(FOTBaseRel.Text) - Convert.ToDouble(FOTNewRel.Text))

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

If MTBFChangeRel.Text.Length = 0 Then

  MTBF_Rel_NA.Visible = True
```

```
Else

    MTBF_Rel_NA.Visible = False



End If



End If

'end if added by shringika on 21dec 2007 as part of validation

End Sub



Protected Sub CalculateNewLoadMTBF_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewLoadMTBF.Click

    MTBF_LR_NA.Visible = False
```

```
        Calculate_New_Load_Runs()

        'Calculate_New_Load()



    End Sub

    Protected Sub CalculateNewPILCMTBF_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewPILCMTBF.Click

        MTBF_NA.Visible = False

        Calculate_New_PILC()



    End Sub

    Protected Sub CalculateNewRelMTBF_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewRelMTBF.Click

        MTBF_Rel_NA.Visible = False

        Calculate_New_Rel()
```

```
End Sub


#Region " Costs "

Protected Sub Initial_Costs()

  Initial_Costs_Load()

  Initial_Costs_PILC()

  Initial_Costs_Rel()


End Sub

Protected Sub CostPerSection_Load_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionLoad.TextChanged

  'TotalCost.Text = "$$$"

  If SectionsChangedLoad.Text <> "" Then

    Total_Cost_Load()

  End If

End Sub

Private Sub Initial_Costs_Load()
```

```
        CostPerSectionLoad.Text = "18000.00"

        TotalCostLoad.Text = ""

        SectionsChangedLoad.Text = ""



    End Sub

    Private Sub Total_Cost_Load()

        TotalCostLoad.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedLoad.Text) *
Convert.ToDouble(CostPerSectionLoad.Text)), 2))

    End Sub



    Protected Sub CostPerSection_PILC_Text_Changed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionPILC.TextChanged

        'TotalCost.Text = "$$$"

        If SectionsChangedPILC.Text <> "" Then

            Total_Cost_PILC()

        End If

    End Sub

    '--modified by shringika on 02/14/2008 for actualtotalcostpilc field
```

```
Private Sub Initial_Costs_PILC()

    CostPerSectionPILC.Text = "18000"

    actualTotalCostPILC.Text = ""

    SectionsChangedPILC.Text = ""

    'TotalCostPILC.Text = "180000"

End Sub

Private Sub Total_Cost_PILC()

    actualTotalCostPILC.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedPILC.Text) *
Convert.ToDouble(CostPerSectionPILC.Text)), 2))

End Sub '*--modification by shringika on 02/14/2008 for actualTotalCostPILC ends

Protected Sub CostPerSection_Rel_Text_Changed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionRel.TextChanged

    'TotalCost.Text = "$$$"
```

```
    If SectionsChangedRel.Text <> "" Then

      Total_Cost_Rel()

    End If

End Sub



Private Sub Initial_Costs_Rel()



    CostPerSectionRel.Text = "18000.00"

    TotalCostRel.Text = ""

    SectionsChangedRel.Text = ""

    CostPerSectionRelElastimold.Text = "18000.00"

    TotalCostRelElastimold.Text = ""

    SectionsChangedRelElastimold.Text = ""



End Sub



Private Sub Total_Cost_Sum()
```

```
                TotalCostSum.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRel.Text) *
Convert.ToDouble(CostPerSectionRel.Text)), 2))

            End Sub


            Protected Sub CostPerSection_Sum_Text_Changed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CostPerSectionSum.TextChanged

              'TotalCost.Text = "$$$"

              If SectionsChangedSum.Text <> "" Then

                Total_Cost_Rel()

              End If

            End Sub


            Private Sub Initial_Costs_Sum()


              CostPerSectionSum.Text = "18000"

              TotalCostSum.Text = ""

              'TotalCostPILC.Text = "180000"

              'SectionsChangedRel.Text = ""
```

```
End Sub

Private Sub Total_Cost_Rel()
  TotalCostRel.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRel.Text) *
Convert.ToDouble(CostPerSectionRel.Text)), 2))
End Sub
#End Region

Private Function getResultsPILCTable() As DataTable
  'gather and validate parameters here
  Dim Feeder As String = cmbSearchFeeder.SelectedValue

  'Get data from the database
  'Dim sql As String = "SELECT [Feeder PILC RANK] AS Feeder_PILC_Rank, [NWK
PILC RANK] AS NWK_PILC_Rank, [PILC Boro Rank] AS Boro_PILC_Rank, description FROM
[Cable_section_rank_pilc_all] WHERE ([Feeder] = '{0}') ORDER BY [Feeder PILC RANK]"
```

```
Dim sql As String = "SELECT * FROM [Rerank_Cable_section_rank_pilc_all] WHERE ([Feeder] = '{0}') ORDER BY [Feeder PILC RANK]"

sql = String.Format(sql, Feeder)

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

Return dt

End Function
```

```
Protected Sub btnFOOrunExcel_click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnovlrunExcel.Click

'work on this later

Return

Dim dt As DataTable = getResultsPILCTable()

If dt Is Nothing Then
  Return
End If

Dim cols As Integer = dt.Columns.Count

'Output excel file
```

```
Response.Clear()

Response.ContentType = "application/xls"

Response.AddHeader("content-disposition", "attachment;
filename=""CapitalPlanning_PILC_Rank.xls""")

Response.Write("<table border='1'>")


Dim i As Integer


'Print Row Headings

Response.Write("<tr>")


Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Feeder Rank</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Network Rank</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Boro Rank</td>")

Response.Write("<td style='FONT-WEIGHT: bold; BACKGROUND-COLOR:
silver'>Description</td>")
```

```
    Response.Write("</tr>")


    'Print Rows
    For Each row As DataRow In dt.Rows
      Response.Write("<tr>")
      For i = 0 To row.ItemArray.Length() - 1
        Response.Write("<td>" & row(i) & "</td>")
      Next
      Response.Write("</tr>")
    Next
    Response.Write("</table>")
    Response.End()
  End Sub
```

```
Protected Sub FMBoro_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles FMBoro.Click

    Dim URL As String

    URL = "feedermaps/" & cmbSearchBorough.SelectedValue & ".html"

    Dim MyScript As New System.Text.StringBuilder
    With MyScript
      .Append("<script type='text/javascript'>")
      .Append("window.open('" & URL & "');")
      .Append("</script>")
    End With

    'Response.Write
    Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
```

```
'Page.ClientScript.RegisterStartupScript(GetType(String), "ClientScript", "alert ('hello');",
True)

End Sub

Protected Sub FMNet_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles FMNet.Click

Dim URL As String

URL = "feedermaps/" & cmbSearchNetwork.SelectedValue.Trim & ".load.html"

Dim MyScript As New System.Text.StringBuilder

With MyScript
```

```
.Append("<script type='text/javascript'>")

.Append("window.open('" & URL & "');")

.Append("</script>")

End With



'Response.Write

Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

End Sub



Protected Sub FMFeeder_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles FMFeeder.Click

Dim URL As String



'URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.html"

URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"



Dim MyScript As New System.Text.StringBuilder
```

```
With MyScript
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Response.Write
Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
End Sub

'Protected Sub CustomValidator1_ServerValidate(ByVal source As Object, ByVal args As
System.Web.UI.WebControls.ServerValidateEventArgs) Handles CustomValidator1.ServerValidate
'  args.IsValid = SectionsChangedLoad.Text.Length() > 0
'End Sub

'Helpers ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
Private Function CalculateCostPILC_Feeder(ByVal Feeder As String, ByVal strategy As
Integer) As Integer

    Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder))

    Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder))

    Dim CNT As Integer



    BindAttributes()

    BindRankingLPW()

    BindFOT()

    'Reset_New_Attributes()

    'added by shringika on 03/11/2008 for LPW step

    'BindAttributes1()

    ' modification ends here 03/11/2008

    If strategy = 1 Then
```

```
    'dv.Sort = "Run_ID ASC"

    dv.Sort = "[Feeder PILC Rank] ASC"

    dv1.Sort = "rank ASC"

    'Replace_paper_percent uses the sort order from above

End If



If strategy = 2 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "load_percentage DESC"

    dv1.Sort = "percentage_load DESC"

End If



If strategy = 3 Then

    'dv.Sort = "Run_ID ASC"

    dv.Sort = "LoadXrank DESC"

    dv1.Sort = "LoadXrank DESC"

End If
```

```
CNT = Replace_Paper_percent(PercentPILC.Text, dv1)

Return CNT



'GVpilcsec.DataSource = dv

'GVpilcsec.DataBind()



'GVpilcruns.DataSource = dv1

'GVpilcruns.DataBind()




'PILC.Visible = False



End Function



'--added by shringika on 17-dec-2007

Protected Sub cmbFeederType_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbFeederType.SelectedIndexChanged
```

```
cmbPercentage.Enabled = False

If cmbFeederType.SelectedItem.Value.Contains("PILC") Then
    cmbPercentage.Items.Clear()
    cmbPercentage.Items.Add("10")
    cmbPercentage.Items.Add("20")
    cmbPercentage.Items.Add("25")
    cmbPercentage.Items.Add("30")
    cmbPercentage.Items.Add("40")
    cmbPercentage.Items.Add("50")
    cmbPercentage.Items.Add("60")
    cmbPercentage.Items.Add("70")
    cmbPercentage.Items.Add("75")
    cmbPercentage.Items.Add("80")
    cmbPercentage.Items.Add("90")
    cmbPercentage.Items.Add("100")
    cmbPercentage.Enabled = True
```

```
End If

If cmbFeederType.SelectedItem.Value.Contains("Load") Then
    cmbPercentage.Items.Clear()
    cmbPercentage.Items.Add("70")
    cmbPercentage.Items.Add("80")
    cmbPercentage.Items.Add("90")
    cmbPercentage.Items.Add("100")
    cmbPercentage.Items.Add("110")
    cmbPercentage.Items.Add("120")
    cmbPercentage.Items.Add("130")
    cmbPercentage.Items.Add("140")
    cmbPercentage.SelectedValue = "100"
    cmbPercentage.Enabled = True

End If
```

```
        BindSearchFeederList()

    End Sub

    Protected Sub cmbPercentage_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmbPercentage.SelectedIndexChanged

        BindSearchFeederList()

    End Sub

    '--modification by shringika on 17-dec-2007 ends

    '--modification by shringika 21-dec-2007 ResetFeederAttributes

    Private Sub ResetFeederAttributes()

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

        If cmbSearchFeeder.Items.Count < 1 Then
```

```
Label2.Text = ""

Feeder.Text = ""

Feeder1.Text = ""

Feeder2.Text = ""

' Feeder3.Text = ""

Feeder4.Text = ""

'Feeder5.Text = ""

Feeder6.Text = ""

Feeder7.Text = ""

feeder11.Text = ""

feeder20.Text = ""

feeder12.Text = ""

Feeder_Name.Text = " "

'SelectedFeederName.TabIndex = ""

SelectedFeederName.TabIndex = 0
```

FLoad.Text = ""

Normal_Rating.Text = ""

Emergency_Rating.Text = ""

Shift.Text = ""

Delta.Text = ""

Emergency_Load.Text = ""

Feeders_4kv.Text = ""

Paper_GT40.Text = ""

Paper_LE40.Text = ""

Solid_LT16.Text = ""

Solid_70_75.Text = ""

Solid_GT16.Text = ""

Percent_Paper_GT40.Text = ""

Percent_Paper_LE40.Text = ""

Percent_Solid_LT16.Text = ""

Percent_Solid_70_75.Text = ""

Percent_Solid_GT16.Text = ""

Sec_CNT.Text = ""

E21GT93.Text = ""

E21LE93.Text = ""

R31GT94.Text = ""

R31LE93.Text = ""

OtherStop_LT7.Text = ""

OtherStop7_12.Text = ""

OtherStop_GT12.Text = ""

AllStop.Text = ""

Percent_ALLStop.Text = ""

PJ_LT8.Text = ""

PJ8_16.Text = ""

PJ16_24.Text = ""

'PJ_GT24.Text = ""

PJ24_32.Text = ""

PJ32_40.Text = ""

PJGT40.Text = ""

AllPaper.Text = ""

Percent_AllPaper.Text = ""

SJ_LT8.Text = ""

SJ8_16.Text = ""

SJ16_24.Text = ""

SJ_GT24.Text = ""

AllSolid.Text = ""

Percent_AllSolid.Text = ""

Percent_E21GT93.Text = ""

Percent_E21LE93.Text = ""

Percent_R31GT94.Text = ""

Percent_R31LE93.Text = ""

Percent_OtherStop_LT7.Text = ""

Percent_OtherStop7_12.Text = ""

Percent_OtherStop_GT12.Text = ""

Percent_PJ_LT8.Text = ""

Percent_PJ8_16.Text = ""

Percent_PJ16_24.Text = ""

'Percent_PJ_GT24.Text = ""

Percent_PJ24_32.Text = ""

Percent_PJ32_40.Text = ""

Percent_PJGT40.Text = ""

Percent_SJ_LT8.Text = ""

Percent_SJ8_16.Text = ""

Percent_SJ16_24.Text = ""

Percent_SJ_GT24.Text = ""

Joint_CNT.Text = ""

XFMR_GT40.Text = ""

XFMR30_39.Text = ""

XFMR20_29.Text = ""

XFMR10_19.Text = ""

XFMR0_9.Text = ""

W_XFMR_GT40.Text = ""

W_XFMR30_39.Text = ""

W_XFMR20_29.Text = ""

W_XFMR10_19.Text = ""

W_XFMR0_9.Text = ""

Percent_XFMR_GT40.Text = ""

```
Percent_XFMR30_39.Text = ""

Percent_XFMR20_29.Text = ""

Percent_XFMR10_19.Text = ""

Percent_XFMR0_9.Text = ""

Percent_W_XFMR_GT40.Text = ""

Percent_W_XFMR30_39.Text = ""

Percent_W_XFMR20_29.Text = ""

Percent_W_XFMR10_19.Text = ""

Percent_W_XFMR0_9.Text = ""

XFMR_CNT_SUMMED.Text = ""

fJeopardy.Text = ""

fCNT.Text = ""

ACTMtbf.Text = ""

'Other stats fill

'StDevMTBF.Text = dr("StDevMTBF")

CountOA.Text = ""

CountCIOA.Text = ""

CountAllOutages.Text = ""
```

```
LastAutoDate.Text = ""

DaysLastOA.Text = ""

LastScheduledWorkDate.Text = ""

LastSuccessfulHipotDate.Text = ""

'all attributes from resetnewattributes set to blank.

mon_Load.Text = ""

mon_Normal_Rating.Text = ""

mon_Emergency_Rating.Text = ""

mon_Shift.Text = ""

mon_Delta.Text = ""

mon_Emergency_Load.Text = ""

mon_4kv.Text = ""

mon_Paper_GT40.Text = ""

mon_Paper_LE40.Text = ""

mon_Solid_LT16.Text = ""

mon_Solid_70_75.Text = ""

mon_Solid_GT16.Text = ""

mon_Percent_Paper_GT40.Text = ""
```

mon_Percent_Paper_LE40.Text = ""

mon_Percent_Solid_LT16.Text = ""

mon_Percent_Solid_70_75.Text = ""

mon_Percent_Solid_GT16.Text = ""

mon_Sec_CNT.Text = ""

mon_E21GT93.Text = ""

mon_E21LE93.Text = ""

mon_R31GT94.Text = ""

mon_R31LE93.Text = ""

mon_OtherStop_LT7.Text = ""

mon_OtherStop7_12.Text = ""

mon_OtherStop_GT12.Text = ""

mon_AllStop.Text = ""

mon_Percent_ALLStop.Text = ""

mon_PJ_LT8.Text = ""

mon_PJ8_16.Text = ""

mon_PJ16_24.Text = ""

'mon_PJ_GT24.Text = ""

```
mon_PJ24_32.Text = ""

mon_PJ32_40.Text = ""

mon_PJGT40.Text = ""

mon_AllPaper.Text = ""

mon_Percent_AllPaper.Text = ""

mon_SJ_LT8.Text = ""

mon_SJ8_16.Text = ""

mon_SJ16_24.Text = ""

mon_SJ_GT24.Text = ""

mon_AllSolid.Text = ""

mon_Percent_AllSolid.Text = ""

mon_Percent_E21GT93.Text = ""

mon_Percent_E21LE93.Text = ""

mon_Percent_R31GT94.Text = ""

mon_Percent_R31LE93.Text = ""

mon_Percent_OtherStop_LT7.Text = ""

mon_Percent_OtherStop7_12.Text = ""

mon_Percent_OtherStop_GT12.Text = ""
```

mon_Percent_PJ_LT8.Text = ""

mon_Percent_PJ8_16.Text = ""

mon_Percent_PJ16_24.Text = ""

' mon_Percent_PJ_GT24.Text = ""

mon_Percent_PJ24_32.Text = ""

mon_Percent_PJ32_40.Text = ""

mon_Percent_PJGT40.Text = ""

mon_Percent_SJ_LT8.Text = ""

mon_Percent_SJ8_16.Text = ""

mon_Percent_SJ16_24.Text = ""

mon_Percent_SJ_GT24.Text = ""

mon_Joint_CNT.Text = ""

mon_XFMR_GT40.Text = ""

mon_XFMR30_39.Text = ""

mon_XFMR20_29.Text = ""

mon_XFMR10_19.Text = ""

mon_XFMR0_9.Text = ""

mon_W_XFMR_GT40.Text = ""

mon_W_XFMR30_39.Text = ""

mon_W_XFMR20_29.Text = ""

mon_W_XFMR10_19.Text = ""

mon_W_XFMR0_9.Text = ""

mon_Percent_XFMR_GT40.Text = ""

mon_Percent_XFMR30_39.Text = ""

mon_Percent_XFMR20_29.Text = ""

mon_Percent_XFMR10_19.Text = ""

mon_Percent_XFMR0_9.Text = ""

mon_Percent_W_XFMR_GT40.Text = ""

mon_Percent_W_XFMR30_39.Text = ""

mon_Percent_W_XFMR20_29.Text = ""

mon_Percent_W_XFMR10_19.Text = ""

mon_Percent_W_XFMR0_9.Text = ""

mon_XFMR_CNT_SUMMED.Text = ""

Jeopardy.Text = ""

CNT.Text = ""

LPW.Text = ""

```
            AVG_LPW.Text = ""

            fAVG_LPW.Text = ""

            fLPW.Text = ""









            Reset_MTBF_Load()

        End If

    End Sub



    Protected Sub CAPTWizard_ActiveStepChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CAPTWizard.ActiveStepChanged

        PILCCablesWorstStopJoints.Visible = False

        UpdateAttributesPILC.Visible = False

        UpdateAttributesRel.Visible = False

        UpdateAttributesLoad.Visible = False
```

```
UpdateAttributesElast.Visible = False

UpdateAttributesStops.Visible = False

PILCCablesWithStopJoints.Visible = False

PILCMTBFInfo.Visible = False

Panel9.Visible = False

PILCMTBFTotalCostInfo.Visible = False

loadStrategyInfo.Visible = False

Runs.Visible = False

downloadtoExcelPanel.Visible = False

ovruns.Visible = False

ovsec.Visible = False

GVRelElastPanel.Visible = False

RelGVElastPanel.Visible = False

XLP.Visible = False

GVRelStopsPanel.Visible = False

MTBF_NA.Visible = False

MTBF_LR_NA.Visible = False

MTBF_Rel_NA.Visible = False
```

```
Reset_MTBF_Load()

Reset_MTBF_PILC()

Reset_MTBF_Rel()

Runs_Select.Visible = False

sectionschangedPILCPanel.Visible = False



'lblTarget.Text = ""

'lblCostPercentage.Text = ""

'lblAchieved.Text = ""

'lblCostPercentageValue.Text = ""




'SectionsChangedPILC.Text = ""

' PILCCalculationReset()



TargetCostCheck.Visible = False

'SectionsChangedPILC.Text = ""

' actualTotalCostPILC.Text = ""
```

```
' actualPercentPILC.Text = ""

' GVPILCStopJoints.Visible = False

' Runs.Visible = False

' UpdateAttributesPILC.Visible = False

'  CommitChangesPILC.Visible = False

'CheckCommitPILC.Checked = False

'PILCCablesWithStopJoints.Visible = False

' GVPILCStopJoints.Visible = False

' End If

CalculateNewLoadMTBF.Enabled = True

' If CAPTWizard.ActiveStepIndex = 2 Then

'If StrategyLoadDropDown.SelectedIndex > 2 Then CalculateNewLoadMTBF.Enabled =
False

'End If

'If CAPTWizard.ActiveStepIndex.Equals(4) Then Exit Sub



'RblTotalCostPercentagePILC.SelectedIndex = 1
```

```
End Sub

'--method added on 2-jan-2008 ends here.



Protected Sub CAPTWizard_SideBarButtonClick(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.WizardNavigationEventArgs) Handles CAPTWizard.SideBarButtonClick

  If cmbSearchFeeder.Items.Count < 1 Then

    e.Cancel = True

  End If

  If e.NextStepIndex.Equals(9) Then

    BindAttributes1()

  End If



End Sub
```

```
'        Protected Sub RblCostPercentage_SelectedIndexChanged(ByVal sender As Object,
ByVal e 'As System.EventArgs) Handles RblCostPercentageRel.SelectedIndexChanged

'        If RblCostPercentageRel.SelectedIndex > 0 Then

'          RelTotalCost.Enabled = False

'          RelPercent.Enabled = True

'     Else

'          RelTotalCost.Enabled = True

'        RelPercent.Enabled = False

'

'

'        End If

'

'     End Sub

'     '*--added by shringika to include totalcost/percent radio button selectioncase
```

```
Protected Sub RblTotalCostPercentagePILC_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles RblTotalCostPercentagePILC.SelectedIndexChanged

    PercentPILC.SelectedIndex = 0

    TotalCostPILC.Text = ""

    actualTotalCostPILC.Text = ""

    SectionsChangedPILC.Text = ""

    actualPercentPILC.Text = ""

    Runs.Visible = False

    downloadtoExcelPanel.Visible = False

    PILCCablesWithStopJoints.Visible = False

    PILCCablesWorstStopJoints.Visible = False

    Runs_Select.Visible = False

    CostPercentPanel.Visible = True

    If RblTotalCostPercentagePILC.SelectedIndex < 1 Then

      'targetcost case

      TotalCostPILC.Enabled = True

      PercentPILC.Enabled = False
```

```
    End If

    If RblTotalCostPercentagePILC.SelectedIndex < 2 Then
      ' percent case
      TargetCostCheck.Visible = False
      TotalCostPILC.Enabled = False
      PercentPILC.Enabled = True

    Else
      'select runs case
      CostPercentPanel.Visible = False
      TargetCostCheck.Visible = False
      TotalCostPILC.Enabled = False
      PercentPILC.Enabled = False
      CalculateCostPILC_SelectRuns()

    End If
  End Sub
```

```
'---modification by shringika to include stop joints calculation in tier2

Private Sub StopJointsCalculation()


  Dim dv1 As DataView = New DataView(GetRunStopTable(Feeder.Text))

  Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))

  'runstop.Visible = True

  If RblTotalCostPercentagePILC.SelectedIndex > 0 Then

    If CheckCommitPILC.Checked Then

      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - duplicate_sections_list)

    Else

      'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2) - visited_sections_count)

      SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_percent_PILC(PercentPILC.Text, dv1, dv2))
```

```
                End If

                Total_Cost_PILC()

            Else


                If CheckCommitPILC.Checked Then

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - duplicate_sections_list)

                Else

                    'SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2) - visited_sections_count)

                    SectionsChangedPILC.Text =
Convert.ToString(Replace_Stops_TotalCost_PILC(TotalCostPILC.Text, dv1, dv2))

                End If

                Total_Cost_PILC()


            End If

            If SectionsChangedPILC.Text = "0" Then UpdateAttributesPILC.Visible = False
```

```
' Code for hierarchgrid


Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PILCJoint_Sections")

dt1 = dv1.ToTable()

dt2 = dv2.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("PILCJoint_Sections", dcc1, dcc2, False)

dss.Relations.Add(drr)


'bind the DataSet to the HierarGrid
```

```
GVPILCStopJoints.DataSource = dss

GVPILCStopJoints.DataMember = dss.Tables(0).TableName()

GVPILCStopJoints.DataBind()



Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



If Equals(m_deltaMTBF, "no") Then

   GVPILCStopJoints.Columns(9).Visible = False

End If

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVPILCStopJoints.Items.Count) - 1

   If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVPILCStopJoints.Items(visited_run_grid).Cells(1).Text),     Nothing))
Then
```

```
                GVPILCStopJoints.Items(visited_run_grid).ForeColor = Drawing.Color.Red


            End If

        Next


        PILCCablesWithStopJoints.Visible = True

        GVPILCStopJoints.Visible = True

        downloadtoExcelPanel.Visible = True




    End Sub


    Private Function Replace_Stops_percent_PILC(ByVal percent As String, ByVal runs As
DataView, ByVal joints As DataView) As Integer


        Dim ovl_cnt As Int32
```

```
Dim i, j As Integer

Dim total_joints, total_sections, total_stops As Integer

Dim running_total, running_sec_total As Integer

Dim per As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer

Dim visitedRuns As New ArrayList

Dim visitedJoints As New ArrayList

Dim dupcount As Integer



'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)

'If percent.Length = 0 Then

'percent = "100"
```

```
' End If

per = Int32.Parse(percent)



running_sec_total = 0

actualPercentPILC.Text = "0"



'If (per = 100) Then

'actualPercentPILC.Text = "100"

'section count is from attributes in funtion so can be different, bummer!

' sec_changed = Replace_Paper()

'

'If (runs.Table.Rows.Count > 0) Then

'Return Int32.Parse(runs.Item(0).Item("ovl_cnt"))

'

'Return sec_changed

'Else : Return 0

'End If

'End If
```

```
If (runs.Table.Rows.Count > 0) Then

  'pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))

  'for now do it this way

  'total_stops = Int32.Parse(mon_AllStop.Text)

  'total_joints = Int32.Parse(mon_Joint_CNT.Text)

  total_stops = Int32.Parse(AllStop.Text)

  total_joints = Int32.Parse(Joint_CNT.Text)

  running_total = 0



  For i = 0 To (runs.Table.Rows.Count - 1)



    runId = runs.Item(i).Item("Run_ID")

    If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
Continue For
```

```
dupcount = 0

'Joints can be in more than one run! Have to check that

'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND stop = 'S'"

For j = 0 To joints.Count - 1

   If visitedJoints.Contains(joints(j)("Joint_ID")) Then

      dupcount = dupcount + 1

   Else

      visitedJoints.Add(joints(j)("Joint_ID"))

   End If



   ' If visitedGlbJoints.Contains(joints(j)("Joint_ID")) Then running_total =
running_total - 1



Next
```

```
joints.RowFilter = Nothing

running_total = running_total + runs.Item(i).Item("Stops") - dupcount

running_sec_total = running_sec_total + runs.Item(i).Item("length")

running_percent = running_total / total_stops * 100.0

running_tot_percent = running_total / total_joints * 100.0

actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))



'do it early now....

If (running_percent >= per) Then

   running_sec_total = running_sec_total - runs.Item(i).Item("length")

   running_total = running_total - (runs.Item(i).Item("Stops") - dupcount)

   running_percent = running_total / total_stops * 100.0

   running_tot_percent = running_total / total_joints * 100.0

   runId = runs.Item(i).Item("Run_ID")



   actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

   Exit For
```

```
End If

runId = runs.Item(i).Item("Run_ID")

runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)

actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

'use this to debug dups

'runs.Item(i).Item("ptsec") = dupcount

'I dont think this check does anything... AGB

If (Not (visitedRuns.Contains(runId))) Then

   visitedRuns.Add(runId)

   If CheckCommitPILC.Checked Then
```

```
                    Update_Attributes(runId, False)

                    If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)


                  End If

                  visited_sections_check(runId, False)

                End If


                'If (running_percent >= per) Then Exit For


              Next

            End If


            ovl_cnt = running_sec_total
```

```
        Return ovl_cnt



    End Function



    Protected Sub GVPILCStopJoints_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles GVPILCStopJoints.SelectedIndexChanged

        Dim dt As DataTable

        Dim Run As Integer

        Dim feeder As String

        feeder = cmbSearchFeeder.SelectedValue

        Run = Int32.Parse(GVPILCStopJoints.SelectedItem.Cells(1).Text)



        dt = GetSectionInRunTable(Run, feeder)



        Dim struct As String

        struct = dt.Rows(0).Item("from_struct")

        Session("struct") = struct
```

```
Dim URL As String

URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"

Dim MyScript As New System.Text.StringBuilder
With MyScript
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
```

```
ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub



Private Sub GVpilcstopjoints_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVPILCStopJoints.TemplateSelection

e.TemplateFilename = "Templates\\RelJointSections.ascx"

End Sub



Protected Sub GVPILCStopJointTypes_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles GVPILCStopJointTypes.SelectedIndexChanged

Dim dt As DataTable

Dim Run As Integer

Dim feeder As String

feeder = cmbSearchFeeder.SelectedValue

Run = Int32.Parse(GVPILCStopJointTypes.SelectedItem.Cells(2).Text)
```

```
dt = GetSectionInRunTable(Run, feeder)

Dim struct As String

struct = dt.Rows(0).Item("from_struct")

Session("struct") = struct

Dim URL As String

URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"

Dim MyScript As New System.Text.StringBuilder

With MyScript

   .Append("<script type='text/javascript'>")

   .Append("window.open('" & URL & "');")

   .Append("</script>")
```

```
        End With

        'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

        ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

        'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

    End Sub

    Private Sub GVpilcstopjointtypes_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVPILCStopJointTypes.TemplateSelection

        e.TemplateFilename = "Templates\\RelJointSections.ascx"

    End Sub
```

```
Protected Sub PercentPILC_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PercentPILC.SelectedIndexChanged

    PILCCalculationReset()

End Sub



Protected Sub CommitChangesPILC_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesPILC.Click

    TargetCostCheck.Visible = False



    CheckCommitPILC.Checked = True

    If RblTotalCostPercentagePILC.SelectedIndex = 2 Then

      CalculateCostPILC_SelectRunsCommitChanges()

      CheckCommitPILC.Checked = False
```

```
            'CommitChangesPILC.Enabled = False

            UpdateAttributesPILC.Visible = False

            Session("strategies_pilc") = StrategyPILCDropDown.SelectedValue



            Exit Sub

        End If



        CalculateCostPILC_CommitChanges()

        CheckCommitPILC.Checked = False

        'CommitChangesPILC.Enabled = False

        UpdateAttributesPILC.Visible = False

        Session("strategies_pilc") = StrategyPILCDropDown.SelectedValue




    End Sub



    Protected Sub CalculateCostPILC_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostPILC.Click
```

```
        CheckCommitPILC.Checked = False

        TargetCostCheck.Visible = False


        If RblTotalCostPercentagePILC.SelectedIndex > 1 Then

            CalculateCostPILC_SelectRunsCommitChanges()

            CostPercentPanel.Visible = True

            UpdateAttributesPILC.Visible = True

            CommitChangesPILC.Visible = True

            downloadtoExcelPanel.Visible = True

            Exit Sub

        End If


        CalculateCostPILC_CommitChanges()

    End Sub


    Protected Sub TotalCostPILC_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TotalCostPILC.TextChanged

        PILCCalculationReset()
```

```
End Sub

Private Sub PILCCalculationReset()

  If cmbSearchFeeder.Items.Count > 0 Then



    TargetCostCheck.Visible = False

    SectionsChangedPILC.Text = ""

    actualTotalCostPILC.Text = ""

    actualPercentPILC.Text = ""

    GVPILCStopJoints.Visible = False

    Runs.Visible = False

    downloadtoExcelPanel.Visible = False

    UpdateAttributesPILC.Visible = False

    CommitChangesPILC.Visible = False

    CheckCommitPILC.Checked = False

    PILCCablesWithStopJoints.Visible = False

    GVPILCStopJoints.Visible = False
```

```
RankBasePILC.Text = ""

FOTBasePILC.Text = ""

MTBFBasePILC.Text = ""

RankNewPILC.Text = ""

RankChangePILC.Text = ""

MTBFNewPILC.Text = ""

MTBFChangePILC.Text = ""

FOTNewPILC.Text = ""

FOTChangePILC.Text = ""



PILCCablesWorstStopJoints.Visible = False
```

```
        End If

    End Sub

    Protected Sub CommitChangesLoad_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesLoad.Click

        CheckCommitLoad.Checked = True

        CalculateCostLoad_CommitChanges()

        Session("strategies_load") = StrategyLoadDropDown.SelectedValue

        CheckCommitLoad.Checked = False

        'CommitChangesLoad.Enabled = False

        UpdateAttributesLoad.Visible = False


    End Sub
```

```
'Protected Sub CommitChangesRel_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesPILC.Click

' CheckCommitRel.Checked = True

' CalculateCostRel_CommitChanges()

'  CheckCommitRel.Checked = False

'CommitChangesRel.Enabled = False

'  UpdateAttributesRel.Visible = False




' End Sub
```

```
Protected Sub CommitChangesRel_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesRel.Click

   CheckCommitRel.Checked = True

   CalculateCostRel_CommitChanges()

   CheckCommitRel.Checked = False
```

'CommitChangesRel.Enabled = False

UpdateAttributesRel.Visible = False

End Sub

Private Function Replace_Stops_TotalCost_PILC(ByVal totalcost As String, ByVal runs As DataView, ByVal joints As DataView) As Integer

'needs changes for PILC sections replacement selection tier's StopJointsStrategy based on toal Cost

'added on 06/23/2008 shringika

Dim ovl_cnt As Int32

Dim i, j, k As Integer

Dim total_sections, pilc_sections_count, running_total, running_sec_total, total_stops, total_joints As Integer

Dim total_cost As Integer

Dim running_percent, running_tot_percent As Decimal

Dim runId As Integer

Dim cost_per_section As Integer

Dim visitedRuns As New ArrayList

```
Dim visitedJoints As New ArrayList

Dim dupcount As Integer




total_stops = Int32.Parse(AllStop.Text)

total_joints = Int32.Parse(Joint_CNT.Text)

running_total = 0

'Dim dv As DataView = New DataView(GetRunLoadTable(Feeder.Text))



total_sections = Int32.Parse(Sec_CNT.Text)

total_cost = Int32.Parse(totalcost)

cost_per_section = Int32.Parse(CostPerSectionPILC.Text)



Dim total_replaced_sections As Integer

total_replaced_sections = Math.Floor(total_cost / cost_per_section)



pilc_sections_count = 0
```

```
If (runs.Table.Rows.Count > 0) Then

    'count the number of pilc sections by running loop earlier so that it can be used for
getting running percent

    'commented by shringika on 06/24/2008

    ' pilc_sections = Int32.Parse(runs.Item(0).Item("ovl_cnt"))

    'why it come to 154 when it should be 48 for '4x40' need to check

    For k = 0 To (runs.Table.Rows.Count - 1)

      pilc_sections_count = pilc_sections_count + runs.Item(k).Item("length")

    Next



    lblsections.Text = pilc_sections_count



    For i = 0 To (runs.Table.Rows.Count - 1)
```

"""""inclusiomn of modifications by Albert into this code for duplicate joints 06/26/2008

runId = runs.Item(i).Item("Run_ID")

If Not Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then Continue For

dupcount = 0

'Joints can be in more than one run! Have to check that

'assumption for PILC runs a joint has to be a stop joint to be in multiple runs

joints.RowFilter = "Run_ID =" & Convert.ToString(runId) & "AND stop = 'S'"

For j = 0 To joints.Count - 1

If visitedJoints.Contains(joints(j)("Joint_ID")) Then

dupcount = dupcount + 1

Else

visitedJoints.Add(joints(j)("Joint_ID"))

End If

```
Next



joints.RowFilter = Nothing

"""""""""""""""modifcation for joints ends here

running_total = running_total + runs.Item(i).Item("stops") - dupcount

'total # of stops

running_sec_total = running_sec_total + runs.Item(i).Item("length")

'total sections

If (running_sec_total > total_replaced_sections) Then

   'running_total = running_total + runs.Item(i).Item("Stops") - dupcount
```

```
    running_total = running_total - (runs.Item(i).Item("stops") - dupcount)

    running_sec_total = running_sec_total - runs.Item(i).Item("length")

    running_percent = running_total / total_stops * 100.0

    actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

    Exit For

End If



running_percent = running_total / total_stops * 100.0

running_tot_percent = running_sec_total / total_sections * 100.0

runId = runs.Item(i).Item("Run_ID")

runs.Item(i).Item("tsec") = running_total

runs.Item(i).Item("sec") = running_sec_total

runs.Item(i).Item("psec") = Math.Round(running_percent)

runs.Item(i).Item("ptsec") = Math.Round(running_tot_percent)



If (Not (visitedRuns.Contains(runId))) Then
```

```
                visitedRuns.Add(runId)

                If CheckCommitPILC.Checked Then

                  Update_Attributes(runId, False)

                  If Equals(VisitedGlbRuns.Items.FindByText(runId.ToString), Nothing) Then
VisitedGlbRuns.Items.Add(runId)

                End If

                visited_sections_check(runId, False)

              End If

              actualPercentPILC.Text = Convert.ToString(Math.Round(running_percent))

              If (running_sec_total.Equals(total_replaced_sections)) Then Exit For

            Next

          End If
```

```
        ovl_cnt = running_sec_total

        'actualPercentPILC.Text = Math.Round(running_total / pilc_sections) * 100.0




        Return ovl_cnt



        'Might want to move this to a separate sub

        'SectionsChangedPILC.Text = Convert.ToString(ovl_cnt)

        'Total_Cost_PILC()

    End Function



    'Protected Sub RelPercent_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RelPercent.SelectedIndexChanged

        ' If cmbSearchFeeder.Items.Count > 0 Then




        '  SectionsChangedRel.Text = ""

        '  TotalCostRel.Text = ""
```

```
' ELAS.Visible = False

' runstop.Visible = False

'  XLP.Visible = False



'   UpdateAttributesRel.Visible = False

'   CommitChangesRel.Visible = False

'  CheckCommitRel.Checked = False



'End If

' End Sub



Private Sub Total_Cost_RelElastimold()

   TotalCostRelElastimold.Text                                                    =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRelElastimold.Text)                *
Convert.ToDouble(CostPerSectionRelElastimold.Text)), 2))

End Sub
```

```
Protected Sub CaculateCostRelElastimold_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CaculateCostRelElastimold.Click

    CheckCommitElast.Checked = False

    visited_sections_count = 0

    CalculateCostRelElastimold_CommitChanges()

End Sub

Private Sub CalculateCostRelElastimold_CommitChanges()

    If cmbSearchFeeder.Items.Count > 0 Then



        BindAttributes()

        BindRankingLPW()

        BindFOT()

        ' Reset_New_Attributes()

        UpdateAttributesElast.Visible = True

        CommitChangesElast.Visible = True

        GVRelElastPanel.Visible = True
```

```
RelGVElastPanel.Visible = True

RelGVElast.Visible = True

GVRelElastStops.Visible = True



'ELASTIMOLDS

Dim dv1 As DataView = New DataView(GetElastimoldTable(Feeder.Text))

Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))



' SectionsChangedRelElastimold.Text =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2))

'Total_Cost_RelElastimold()



If CheckCommitElast.Checked Then

  SectionsChangedRelElastimold.Text =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2) - duplicate_sections_list)

  Total_Cost_RelElastimold()
```

```
                Else

                    '                                   SectionsChangedRelElastimold.Text =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2) - visited_sections_count)

                    SectionsChangedRelElastimold.Text                                   =
Convert.ToString(Replace_Elastimold_percent(RelPercentElastimold.Text, dv1, dv2))

                    Total_Cost_RelElastimold()

                End If



                If SectionsChangedRelElastimold.Text = "0" Then UpdateAttributesElast.Visible =
False



                ' Code for hierarchgrid



                Dim dss As DataSet = New DataSet()

                Dim dt1 As DataTable = New DataTable("RelRuns")

                Dim dt2 As DataTable = New DataTable("RelJoint_Sections")



                dt1 = dv1.ToTable()

                dt2 = dv2.ToTable()
```

```
dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("RelElastJoint_Sections", dcc1, dcc2,
False)

dss.Relations.Add(drr)



'bind the DataSet to the HierarGrid

GVRelElastStops.DataSource = dss

GVRelElastStops.DataMember = dss.Tables(0).TableName()

GVRelElastStops.DataBind()

Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);
```

```
If Equals(m_deltaMTBF, "no") Then
    GVRelElastStops.Columns(9).Visible = False
End If

Dim visited_run_grid As Integer
For visited_run_grid = 0 To (GVRelElastStops.Items.Count) - 1
    If Not (Equals(VisitedGlbRuns.Items.FindByText(GVRelElastStops.Items(visited_run_grid).Cells(1).Text), Nothing)) Then
        GVRelElastStops.Items(visited_run_grid).ForeColor = Drawing.Color.Red
    End If
Next
```

```
        End If

    End Sub



    Protected Sub GVRelElast_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVRelElastStops.TemplateSelection

        e.TemplateFilename = "Templates\\RelJointSections.ascx"

    End Sub



    Protected Sub CommitChangesElast_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesElast.Click

        CheckCommitElast.Checked = True

        CalculateCostRelElastimold_CommitChanges()

        CheckCommitElast.Checked = False

        'CalculateCostRelElastimold_CommitChanges()



        'CommitChangesRelElast.Enabled = False

        UpdateAttributesElast.Visible = False
```

```
End Sub

Protected Sub CalculateNewRelMTBFElast_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewRelMTBFElast.Click
    MTBF_NA_RelElast.Visible = False
    Calculate_New_RelElast()

End Sub

Private Sub Calculate_New_RelElast()
    'modified by shringika on dec 21'2007 included if case
    If cmbSearchFeeder.Items.Count > 0 Then

        Dim dt As DataTable = New DataTable
        Dim Rank As Integer
        'have to work on this query more
```

```
'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '"
& cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '"
& cmbSearchFeeder.SelectedValue & "'and rel_percent =  '" & RelPercentElastimold.Text & "' "

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 60

myCommand.Fill(dt)

'return if there is no MTBF data computed

'future could call webservice?

If (dt.Rows.Count = 0) Then

  MTBF_NA_RelElast.Visible = True

  Exit Sub

End If
```

```
Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary



RankBaseRelElast.Text = dr("OldRank")

Rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression (New 6-08)

'MTBFBaseRelElast.Text = (Rank * 0.3386 + 165.96).ToString("F0")

'modified on 01/04/2009 for picking up rank from rerank_cost_score



'MTBFBaseRel.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

MTBFBaseRelElast.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseRelElast.Text = Convert.ToString(Math.Round(Math.Max((Rank * -0.0095 +
6), 0.0), 0))



'TEMP!!!!!!!!!!!!
```

```
' RankNewRelElast.Text = Convert.ToString(rand(Rank, Rank + 100))

'modified on 01/04/2009 for picking up rank from rerank_cost_score

RankNewRelElast.Text = dr("rank")



'RankNewPILC.Text = "930"



RankChangeRelElast.Text = Convert.ToString(Int32.Parse(RankNewRelElast.Text) -
Rank)

'added on 01/04/2009 for picking up new mtbf and delta from database , not by
calculation

MTBFNewRelElast.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

MTBFChangeRelElast.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'conversion of rank to MTBF using regression

'MTBFNewRel.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

'MTBFChangeRel.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'commented on 01/04/2009 for picking up values from database

'MTBFNewRelElast.Text = (Int32.Parse(RankNewRelElast.Text) * 0.3386 +
165.96).ToString("F0")
```

```
'MTBFChangeRelElast.Text =
Convert.ToString(Convert.ToDouble(MTBFNewRelElast.Text) - Convert.ToDouble(MTBFBaseRelElast.Text))

FOTNewRelElast.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewRelElast.Text) * -0.0095 + 6), 0.0), 0))

FOTChangeRelElast.Text =
Convert.ToString(Convert.ToDouble(FOTBaseRelElast.Text) - Convert.ToDouble(FOTNewRelElast.Text))


myCommand = Nothing

myConnection.Close()

myConnection = Nothing

If MTBFChangeRelElast.Text.Length = 0 Then

  MTBF_NA_RelElast.Visible = True

Else

  MTBF_NA_RelElast.Visible = False


End If
```

```
        End If

        'end if added by shringika on 21dec 2007 as part of validation

    End Sub



    Protected Sub GVRelStops_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GVRelStops.SelectedIndexChanged

        Dim dt As DataTable

        Dim Run As Integer

        Dim feeder As String

        feeder = cmbSearchFeeder.SelectedValue

        Run = Int32.Parse(GVRelStops.SelectedItem.Cells(1).Text)



        dt = GetSectionInRunTable(Run, feeder)



        Dim struct As String

        struct = dt.Rows(0).Item("from_struct")

        Session("struct") = struct
```

```
Dim URL As String

URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"

Dim MyScript As New System.Text.StringBuilder
With MyScript
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
```

```
ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub



Protected Sub GVRelStops_TemplateSelection(ByVal sender As Object, ByVal e As
DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles GVRelStops.TemplateSelection

e.TemplateFilename = "Templates\\RelJointSections.ascx"

End Sub



Protected Sub CalculateCostRelStops_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateCostRelStops.Click

CheckCommitStops.Checked = False

CalculateCostRelStops_CommitChanges()

End Sub

Private Sub CalculateCostRelStops_CommitChanges()

If cmbSearchFeeder.Items.Count > 0 Then
```

```
BindAttributes()

BindRankingLPW()

BindFOT()

'Reset_New_Attributes()



CommitChangesStops.Visible = True

GVRelStops.Visible = True

UpdateAttributesStops.Visible = True

GVRelStopsPanel.Visible = True

'STOP JOINTS

Dim dv1 As DataView = New DataView(GetRunStopTable(Feeder.Text))

Dim dv2 As DataView = New DataView(GetRunJointsTable(Feeder.Text))

'runstop.Visible = True




If CheckCommitStops.Checked Then
```

```
                SectionsChangedRelStops.Text =
Convert.ToString(Replace_Stops_percent(RelPercentStops.Text, dv1, dv2) - duplicate_sections_list)

            Else

                ' SectionsChangedRelStops.Text =
Convert.ToString(Replace_Stops_percent(RelPercentStops.Text, dv1, dv2) - visited_sections_count)

                SectionsChangedRelStops.Text =
Convert.ToString(Replace_Stops_percent(RelPercentStops.Text, dv1, dv2))

            End If

            If SectionsChangedRelStops.Text = "0" Then UpdateAttributesStops.Visible = False



            Total_Cost_RelStops()



            ' Code for hierarchgrid



            Dim dss As DataSet = New DataSet()

            Dim dt1 As DataTable = New DataTable("RelRuns")

            Dim dt2 As DataTable = New DataTable("RelJoint_Sections")

            dt1 = dv1.ToTable()

            dt2 = dv2.ToTable()
```

```
dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn

'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("RelJoint_Sections", dcc1, dcc2, False)

dss.Relations.Add(drr)



'bind the DataSet to the HierarGrid

GVRelStops.DataSource = dss

GVRelStops.DataMember = dss.Tables(0).TableName()

GVRelStops.DataBind()

Dim m_deltaMTBF As String

m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);
```

```
If Equals(m_deltaMTBF, "no") Then

  GVRelStops.Columns(9).Visible = False

End If


Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVRelStops.Items.Count) - 1

  If Not (Equals(VisitedGlbRuns.Items.FindByText(GVRelStops.Items(visited_run_grid).Cells(1).Text), Nothing)) Then

    GVRelStops.Items(visited_run_grid).ForeColor = Drawing.Color.Red

  End If

Next

End If
```

```
        'end if added on 21-dec-2007 by shringika

    End Sub

    Private Sub Total_Cost_RelStops()

        TotalCostRelStops.Text =
Convert.ToString(Math.Round((Convert.ToDouble(SectionsChangedRelStops.Text) *
Convert.ToDouble(CostperSectionRelStops.Text)), 2))

    End Sub


    Protected Sub CommitChangesStops_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CommitChangesStops.Click

        UpdateAttributesStops.Visible = False



        CommitChangesStops.Visible = False

        CheckCommitStops.Checked = True

        GVRelStopsPanel.Visible = False

        CalculateCostRelStops_CommitChanges()

        'CommitChangesRel.Enabled = False

        UpdateAttributesStops.Visible = False
```

```
End Sub

Protected Sub CalculateNewRelMTBFStops_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CalculateNewRelMTBFStops.Click
    MTBF_NA_RelStops.Visible = False
    Calculate_New_RelStops()
End Sub
Private Sub Calculate_New_RelStops()

    'modified by shringika on dec 21'2007 included if case
    If cmbSearchFeeder.Items.Count > 0 Then

      MTBF_NA_RelStops.Visible = False
      If SectionsChangedRelStops.Text = "" Then
        MTBF_NA_RelStops.Visible = True
        Return
```

```
End If

Dim feeder_id As String

If cmbSearchFeeder.SelectedValue.Length < 5 Then

   feeder_id = "0" + cmbSearchFeeder.SelectedValue

Else

   feeder_id = cmbSearchFeeder.SelectedValue

End If



If Session("strategies_pilc") Is Nothing Then Session("strategies_pilc") = "0"

Dim pilc_strategy As String = CType(Session("strategies_pilc"), String)



If Session("strategies_load") Is Nothing Then Session("strategies_load") = "0"

Dim load_strategy As String = CType(Session("strategies_load"), String)



Dim dt As DataTable = New DataTable

Dim Rank As Integer

Dim sql As String

'have to work on this query more
```

'Dim sql As String = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and pilc_percent = 100"

If load_strategy.Length < 2 Then

If pilc_strategy.Length < 2 Then

sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "' and rel_percent = '" & RelPercentStops.Text & "'"

Else

sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "' AND pilc_strategy='" & pilc_strategy & "' AND pilc_percent= '" & PercentPILC.SelectedValue & "' AND rel_percent = '" & RelPercentStops.SelectedValue & "'"

End If

Else

If pilc_strategy.Length < 2 Then

sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" & cmbSearchFeeder.SelectedValue & "'and rel_percent = '" & RelPercentStops.Text & "' AND load_relief_percent='" & StrategyLoadDropDown.SelectedValue & "'"

```
                Else

                    sql = "SELECT * FROM [Rerank_Cost_Score] WHERE [sFeeder] = '" &
cmbSearchFeeder.SelectedValue & "' AND pilc_strategy='" & pilc_strategy & "' AND pilc_percent= '" &
PercentPILC.SelectedValue & "' AND rel_percent = '" & RelPercentStops.SelectedValue & "' AND
load_relief_percent='100'"

                    ' & StrategyLoadDropDown.SelectedValue & """

                End If



            End If



            ' TextBox1.Text = sql



            Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

            Dim myCommand As New SqlDataAdapter(sql, myConnection)

            myCommand.SelectCommand.CommandTimeout = 180

            myCommand.Fill(dt)

            'return if there is no MTBF data computed
```

```
'future could call webservice?

If (dt.Rows.Count = 0) Then

   MTBF_NA_RelStops.Visible = True

   Return

End If



Dim dr As DataRow = dt.Rows(0)

'ABOVE is only temporary



RankBaseRelStops.Text = dr("OldRank")

Rank = Int32.Parse(dr("OldRank"))

'conversion  of rank to MTBF using regression (New 6-08)

'MTBFBaseRelStops.Text = (Rank * 0.3386 + 165.96).ToString("F0")

MTBFBaseRelStops.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")



'MTBFBaseRel.Text = Convert.ToDouble(dr("OldMTBF")).ToString("F0")

FOTBaseRelStops.Text = Convert.ToString(Math.Round(Math.Max((Rank * -0.0095 +
6), 0.0), 0))
```

'modified on 01/04/2009 for picking up rank from rerank_cost_score

RankNewRelStops.Text = dr("rank")

'RankNewRelStops.Text = Convert.ToString(rand(Rank, Rank + 100))

RankChangeRelStops.Text = Convert.ToString(Int32.Parse(RankNewRelStops.Text) - Rank)

'conversion of rank to MTBF using regression

'MTBFNewRel.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

'MTBFChangeRel.Text = Convert.ToDouble(dr("delta")).ToString("F0")

'added on 01/04/2009 for picking up new mtbf and delta from database , not by calculation

```
MTBFNewRelStops.Text = Convert.ToDouble(dr("NewMTBF")).ToString("F0")

MTBFChangeRelStops.Text = Convert.ToDouble(dr("delta")).ToString("F0")



'MTBFNewRelStops.Text = (Int32.Parse(RankNewRelStops.Text) * 0.3386 +
165.96).ToString("F0")

'MTBFChangeRelStops.Text =
Convert.ToString(Convert.ToDouble(MTBFNewRelStops.Text) - Convert.ToDouble(MTBFBaseRelStops.Text))

FOTNewRelStops.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankNewRelStops.Text) * -0.0095 + 6), 0.0), 0))

FOTChangeRelStops.Text =
Convert.ToString(Convert.ToDouble(FOTBaseRelStops.Text) - Convert.ToDouble(FOTNewRelStops.Text))



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

If MTBFChangeRelStops.Text.Length = 0 Then

  MTBF_NA_RelStops.Visible = True

Else

  MTBF_NA_RelStops.Visible = False
```

```
            End If

        End If

        'end if added by shringika on 21dec 2007 as part of validation

    End Sub

    Private Sub BindAttributes1()

        Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

        ' Dim myCommand As New SqlCommand

        ' myCommand.CommandType = CommandType.StoredProcedure

        ' myCommand.CommandText = " Exec UpdateBanksOff_Current"
```

```
'    myCommand.Connection = myConnection



'   myCommand = Nothing

'   myConnection.Close()

'  myConnection = Nothing
```

OpenMainsLTE2_Feeder()

OpenMainsGT2_Feeder()

OpenMainsLTE2_NBY_FCTR_GTE10()

OpenMainsLTE2_NBY_FCTR_GTE5LT10()

OpenMainsGT2_NBY_FCTR_GTE10()

OpenMainsGT2_NBY_FCTR_GTE5LT10()

BanksOff_NBY_FCTR_GTE5LT10()

BanksOff_Feeder()

BanksOff_NBY_FCTR_GTE10()

NetRms_Fuse_Vault_Scan_Feeder()

NetRms_Fuse_NBY_FCTR_GTE10()

NetRms_Fuse_NBY_FCTR_GTE5LT10()

NetRms_SwitchesOpen_Vault_Scan_Feeder()

NetRms_SwitchesOpen_NBY_FCTR_GTE5LT10()

NetRms_SwitchesOpen_NBY_FCTR_GTE10()

NetRms_ClosedDNU_Vault_Scan_Feeder()

NetRms_ClosedDNU_GTE10()

NetRms_ClosedDNU_GTE5LT10()

'PlannedBanksOff_Feeder()

' PlannedBanksOff_Feeder_GTE10()

'PlannedBanksOff_Feeder_GTE5LT10()

NetRms_Missing_Vault_Scan_Feeder()

NetRms_Missing_Vault_GTE5LT10()

NetRms_Missing_Vault_GTE10()

' PlannedOpenMainsLTE2_Feeder()

'PlannedOpenMainsGT2_Feeder()

'PlannedOpenMainsLTE2_GTE10()

'PlannedOpenMainsGT2_GTE10()

' PlannedOpenMainsGT2_GTE5LT10()

' PlannedOpenMainsLTE2_GTE5LT10()

NetRms_ClosedOld_Vault_Scan_Feeder()

NetRms_ClosedUNR_Vault_Scan_Feeder()

NetRms_ClosedUNR_Vault_Scan_GTE5LT10()

NetRms_ClosedUNR_Vault_Scan_GTE10()

NetRms_ClosedOld_Vault_Scan_GTE5LT10()

NetRms_ClosedOld_Vault_Scan_GTE10()

Total_LPW_Calc()

TargetInitialValues_LPW()

'Target_Total_LPW_Calc()

'LPW_Cost_Calculation()

```
'populating the vault details



'code from show vault detail button



VaultDetailsPanel.Visible = True

Check_Target_Values()




Target_Total_LPW_Calc()

LPW_Cost_Calculation()

' LPW_grid()









GridviewdataBindBanksOff()
```

GridViewdataBindOpenSwitches()

GridViewdataBindFuse()

GridViewDataBindClosedOld()

GridViewDataBindClosedUNR()

GridViewDataBindClosedDNU()

GridViewDataBindOpenMainsLTE2()

GridViewDataBindOpenMainsGT2()

GridviewdataBindMissing()

GridviewdataBindBanksOff_NbyFactorGTE10()

GridViewdataBindOpenSwitches_NbyFactorGTE10()

GridViewdataBindFuse_NbyFactorGTE10()

GridViewDataBindClosedOld_NbyFactorGTE10()

GridViewDataBindClosedUNR_NbyFactorGTE10()

GridViewDataBindClosedDNU_NbyFactorGTE10()

GridViewDataBindOpenMainsLTE2_NbyFactorGTE10()

GridViewDataBindOpenMainsGT2_NbyFactorGTE10()

GridviewdataBindMissing_NbyFactorGTE10()

```
GridviewdataBindBanksOff_NbyFactorGTE5LT10()

GridViewdataBindOpenSwitches_NbyFactorGTE5LT10()

GridViewdataBindFuse_NbyFactorGTE5LT10()

GridViewDataBindClosedOld_NbyFactorGTE5LT10()

GridViewDataBindClosedUNR_NbyFactorGTE5LT10()

GridViewDataBindClosedDNU_NbyFactorGTE5LT10()

GridViewDataBindMissing_NbyFactorGTE5LT10()

GridViewDataBindOpenMainsLTE2_NbyFactorGTE5LT10()

GridViewDataBindOpenMainsGT2_NbyFactorGTE5LT10()

IRlpw.Text = fLPW.Text



IR_Scandatetime()




End Sub
```

```
Private Sub lpwreset()

    Total_LPW_Calc()




    TargetInitialValues_LPW()

    'Target_Total_LPW_Calc()

    'LPW_Cost_Calculation()




    'populating the vault details


    'code from show vault detail button


    VaultDetailsPanel.Visible = True
```

' Check_Target_Values()

Target_Total_LPW_Calc()

LPW_Cost_Calculation()

' LPW_grid()

GridviewdataBindBanksOff()

GridViewdataBindOpenSwitches()

GridViewdataBindFuse()

GridViewDataBindClosedOld()

GridViewDataBindClosedUNR()

GridViewDataBindClosedDNU()

GridViewDataBindOpenMainsLTE2()

GridViewDataBindOpenMainsGT2()

GridviewdataBindMissing()

GridviewdataBindBanksOff_NbyFactorGTE10()

GridViewdataBindOpenSwitches_NbyFactorGTE10()

GridViewdataBindFuse_NbyFactorGTE10()

GridViewDataBindClosedOld_NbyFactorGTE10()

GridViewDataBindClosedUNR_NbyFactorGTE10()

GridViewDataBindClosedDNU_NbyFactorGTE10()

GridViewDataBindOpenMainsLTE2_NbyFactorGTE10()

GridViewDataBindOpenMainsGT2_NbyFactorGTE10()

GridViewDataBindMissing_NbyFactorGTE10()

GridviewdataBindBanksOff_NbyFactorGTE5LT10()

GridViewdataBindOpenSwitches_NbyFactorGTE5LT10()

```
    GridViewdataBindFuse_NbyFactorGTE5LT10()

    GridViewDataBindClosedOld_NbyFactorGTE5LT10()

    GridViewDataBindClosedUNR_NbyFactorGTE5LT10()

    GridViewDataBindClosedDNU_NbyFactorGTE5LT10()

    GridViewDataBindOpenMainsLTE2_NbyFactorGTE5LT10()

    GridViewDataBindOpenMainsGT2_NbyFactorGTE5LT10()

    GridViewDataBindMissing_NbyFactorGTE5LT10()

    IRlpw.Text = fLPW.Text



    IR_Scandatetime()

End Sub



'*--modification on 02/14/2008 ends here



'*--added by shringika for LPW calculation



Private Sub BanksOff_NBY_FCTR_GTE10()
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue



If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

BanksOff_GTE10.Text = drcheck(dr, "bo_count")
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub BanksOff_NBY_FCTR_GTE5LT10()



    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String
```

```
Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then
   Feeder_name = "0" + Feeder_name
End If


'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
sql = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER = '" & Feeder_name & "'AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"


Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
        Dim myCommand As New SqlDataAdapter(sql, myConnection)

        myCommand.SelectCommand.CommandTimeout = 120

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        BanksOff_GTE5LT10.Text = drcheck(dr, "bo_count")









        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub
```

```
Private Sub BanksOff_Feeder()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



      Dim Feeder_name As String

      Feeder_name = cmbSearchFeeder.SelectedValue



      If Feeder_name.Length < 5 Then

        Feeder_name = "0" + Feeder_name

      End If



      'Dim sql As String = "SELECT count(*) as bo_count FROM
[Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" &
cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH] where Feeder_ID = '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

BanksOffFeederCurrent.Text = drcheck(dr, "bo_count")
```

```
        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsLTE2_Feeder()

    'transformer on feeder

    Dim sql As String



    If cmbSearchFeeder.Items.Count > 0 Then



        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue



        If Feeder_name.Length < 5 Then
```

```
Feeder_name = "0" + Feeder_name

End If

' If cmbSearchFeeder.SelectedValue.Contains("B") Or
cmbSearchFeeder.SelectedValue.Contains("Q") Then

'Dim sql As String = "SELECT count(*) as bo_count FROM
[Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" &
cmbSearchFeeder.SelectedValue & "'"

' sql = "SELECT count(*) as om_count FROM
[Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name
& "'"

'commented on 10/12/2009 based on lpw discussion with Albert for including all vault
entries

' sql = "SELECT count(*) as om_count FROM
[Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name
& "'"

sql = "SELECT count(*) as om_count FROM
[Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name
& "'"
```

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSXW_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name & "'"

' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSM_3MONTH_LTE2] where LVSD_Feeder='" & Feeder_name & "'"

' End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

```
        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        OpenMainsLTE2Current.Text = drcheck(dr, "om_count")









        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsGT2_Feeder()

    'transformer on feeder

    Dim sql As String
```

```
If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    ' If cmbSearchFeeder.SelectedValue.Contains("B") Or
cmbSearchFeeder.SelectedValue.Contains("Q") Then

    'Dim sql As String = "SELECT count(*) as bo_count FROM
[Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" &
cmbSearchFeeder.SelectedValue & "'"

    'commented on 10/12/2009 for including new open mains definition
```

'sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

' sql = "SELECT (case when vault_entries is not null then vault_entries else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSXW_3MONTH_GT2] where LVSD_FEEDER='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSM_3MONTH_GT2] where LVSD_Feeder='" & Feeder_name & "'"

```
'End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

OpenMainsGT2Current.Text = drcheck(dr, "om_count")



myCommand = Nothing

myConnection.Close()
```

```
        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsGT2_NBY_FCTR_GTE5LT10()

    'transformer on feeder

    Dim sql As String



    If cmbSearchFeeder.Items.Count > 0 Then



        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue



        If Feeder_name.Length < 5 Then

            Feeder_name = "0" + Feeder_name

        End If
```

'If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

'commented on 10/12/2009 for new openmains count criteria

'sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id = '" & Feeder_name & "')"

' sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id = '" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id = '" & Feeder_name & "')"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

```
' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSXW_3MONTH_GT2OM_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OM_NETRMSM_3MONTH_GT2OM_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "'"

' End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)
```

```
        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        OMGT2_GTE5LT10.Text = drcheck(dr, "om_count")




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub

Private Sub OpenMainsGT2_NBY_FCTR_GTE10()

    'transformer on feeder

    Dim sql As String



    If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If


'commented on 10/12/2009 for new om count criteria

'sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] WHERE LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in( Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

' sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end ) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] WHERE LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in( Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] WHERE LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in( Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

OMGT2_GTE10.Text = drcheck(dr, "om_count")







myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If
```

```
End Sub

Private Sub OpenMainsLTE2_NBY_FCTR_GTE5LT10()
  'transformer on feeder
  Dim sql As String

  If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String
    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then
      Feeder_name = "0" + Feeder_name
    End If
```

'commenetd on 10/12/2009 for new om criteria

' sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id= '" & Feeder_name & "')"

' sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id= '" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id= '" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

```
        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        OMLTE2_GTE5LT10.Text = drcheck(dr, "om_count")









        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub OpenMainsLTE2_NBY_FCTR_GTE10()
```

```
'transformer on feeder

Dim sql As String


If cmbSearchFeeder.Items.Count > 0 Then


  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue


  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If


  'commenetd on 10/12/2009 for new om criteria

  'sql = "SELECT count(*) as om_count as om_count FROM
[Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" &
Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM
[Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

'sql = "SELECT (case when sum(vault_entries) is not null then sum(vault_entries) else 0 end) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

sql = "SELECT count(*) as om_count FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

OMLTE2_GTE10.Text = drcheck(dr, "om_count")

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If

End Sub

Private Sub NetRms_Fuse_Vault_Scan_Feeder()
    'transformer on feeder
    Dim sql As String
    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
    If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If


sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE LVS_VAULT_CONDITION like '%D%' AND LVS_FEEDER_NO='" & Feeder_name & "'"


Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180
```

```
        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)



        FuseFeederCurrent.Text = drcheck(dr, "fuse_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub NetRms_Fuse_NBY_FCTR_GTE10()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



Fuse_GTE10.Text = drcheck(dr, "fuse_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub
```

```
Private Sub NetRms_Fuse_NBY_FCTR_GTE5LT10()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



      Dim Feeder_name As String

      Feeder_name = cmbSearchFeeder.SelectedValue



      If Feeder_name.Length < 5 Then

        Feeder_name = "0" + Feeder_name

      End If



      ' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then
```

sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

'sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_Fuse_NBY_FCTR_GTE5LT10] WHERE LVS_FEEDER_NO='" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) AS fuse_count FROM [Transformer_OM_BO].[dbo].[NETRMSM_Fuse_NBY_FCTR_GTE5LT10] WHERE LVS_FEEDER_NO='" & Feeder_name & "'"

'End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)



        Fuse_GTE5LT10.Text = drcheck(dr, "fuse_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub NetRms_SwitchesOpen_Vault_Scan_Feeder()

    'transformer on feeder
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    ' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

    'sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE LVS_NWP_STATUS_FLAG like '%B%' AND LVS_FEEDER_NO = '" & Feeder_name & "'"
```

sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE ([Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%B%' OR [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%A%') AND LVS_FEEDER_NO = '" & Feeder_name & "'"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

'sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_2WEEK_VAULTSCAN] WHERE LVS_NWP_STATUS_FLAG like '%B%' AND LVS_FEEDER_NO='" & Feeder_name & "'"

' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

' sql = "SELECT count(*) AS open_count FROM [Transformer_OM_BO].[dbo].[NETRMSM_2WEEK_VAULTSCAN] WHERE LVS_NWP_STATUS_FLAG like '%B%' AND LVS_FEEDER_NO='" & Feeder_name & "'"

' End If

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

```
        Dim myCommand As New SqlDataAdapter(sql, myConnection)
        myCommand.SelectCommand.CommandTimeout = 180
        Dim dt As DataTable = New DataTable
        myCommand.Fill(dt)

        Dim dr As DataRow = dt.Rows(0)

        OpenSwitchesFeederCurrent.Text = drcheck(dr, "open_count")

        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If

End Sub

Private Sub NetRms_SwitchesOpen_NBY_FCTR_GTE10()
    'transformer on feeder
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    sql = "SELECT count(*) AS opens_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE10] WHERE LVSD_FEEDER= '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



OpenSwitches_GTE10.Text = drcheck(dr, "opens_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub

Private Sub NetRms_SwitchesOpen_NBY_FCTR_GTE5LT10()
```

```
'transformer on feeder

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then



  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue



  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If



  sql = "SELECT count(*) AS opens_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE5LT10] WHERE
```

```
LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number
FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)


                Dim dr As DataRow = dt.Rows(0)


                OpenSwitches_GTE5LT10.Text = drcheck(dr, "opens_count")


                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing
```

```
    End If


End Sub



Private Sub NetRms_ClosedDNU_Vault_Scan_Feeder()
    'transformer on feeder
    Dim sql As String
    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
    If cmbSearchFeeder.Items.Count > 0 Then


        Dim Feeder_name As String
        Feeder_name = cmbSearchFeeder.SelectedValue


        If Feeder_name.Length < 5 Then
            Feeder_name = "0" + Feeder_name
```

End If

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_Failed_Flags]='U' AND LVS_VAULT_CONDITION NOT LIKE '%B%' AND LVS_VAULT_CONDITION NOT LIKE '%A%'AND LVS_FEEDER_NO='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

```
            ClosedDNUCurrent.Text = drcheck(dr, "closed_count")



            myCommand = Nothing

            myConnection.Close()

            myConnection = Nothing

        End If



    End Sub



    Private Sub NetRms_ClosedDNU_GTE10()

        'transformer on feeder

        Dim sql As String

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

        If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue


If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If


sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in(Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"


Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)
```

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)



        ClosedDNU_GTE10.Text = drcheck(dr, "closed_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub

Private Sub NetRms_ClosedDNU_GTE5LT10()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "'AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



ClosedDNU_GTE5LT10.Text = drcheck(dr, "closed_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub
```

```
Private Sub PlannedBanksOff_Feeder()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



      Dim Feeder_name As String

      Feeder_name = cmbSearchFeeder.SelectedValue



      If Feeder_name.Length < 5 Then

        Feeder_name = "0" + Feeder_name

      End If
```

```
sql = "SELECT count(*) AS pbo_count FROM [Transformer_OM_BO].[dbo].[PlannedBanksOff] where Feeder_ID='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

'PlannedBanksOff.Text = drcheck(dr, "pbo_count")


myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If
```

```
End Sub

Private Sub PlannedBanksOff_Feeder_GTE10()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If
```

```
sql = "SELECT count(*) AS pbo_count FROM [Transformer_OM_BO].[dbo].[PlannedBanksOff_GTE10] where Feeder_ID='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

'  PlannedBanksOff_GTE10.Text = drcheck(dr, "pbo_count")



myCommand = Nothing

myConnection.Close()
```

```
        myConnection = Nothing
      End If

    End Sub

    Private Sub PlannedBanksOff_Feeder_GTE5LT10()
      'transformer on feeder
      Dim sql As String
      '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
      If cmbSearchFeeder.Items.Count > 0 Then

        Dim Feeder_name As String
        Feeder_name = cmbSearchFeeder.SelectedValue

        If Feeder_name.Length < 5 Then
          Feeder_name = "0" + Feeder_name
```

```
End If

sql = "SELECT count(*) AS pbo_count FROM [Transformer_OM_BO].[dbo].[PlannedBanksOff_GTE5LT10] where Feeder_ID='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

' PlannedBanksOff_GTE5LT10.Text = drcheck(dr, "pbo_count")
```

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
      End If

    End Sub

    Private Sub NetRms_Missing_Vault_Scan_Feeder()
      'transformer on feeder
      Dim sql As String
      '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
      If cmbSearchFeeder.Items.Count > 0 Then

        Dim Feeder_name As String
        Feeder_name = cmbSearchFeeder.SelectedValue
```

```
If Feeder_name.Length < 5 Then
  Feeder_name = "0" + Feeder_name
End If

' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE (year([LVS_vault_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

'ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_2WEEK_VAULTSCAN] WHERE (year([LVS_NWK_SCAN_TIME_DATE])< 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then
```

```
' sql = "SELECT count(*) AS missing_count FROM
[Transformer_OM_BO].[dbo].[NETRMSM_2WEEK_VAULTSCAN] WHERE
(year([LVS_NWK_SCAN_TIME_DATE])< 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

' End If


Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

MissingFeederCurrent.Text = drcheck(dr, "missing_count")


myCommand = Nothing

myConnection.Close()

myConnection = Nothing
```

```
End If

End Sub

Private Sub NetRms_Missing_Vault_GTE5LT10()
  'transformer on feeder
  Dim sql As String
  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
  If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String
    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then
      Feeder_name = "0" + Feeder_name
```

End If

sql = "SELECT count(distinct([LVSD_VAULT_NUMBER])) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

' If cmbSearchFeeder.SelectedValue.Contains("B") Or cmbSearchFeeder.SelectedValue.Contains("Q") Then

'sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSBQ_2WEEK_VAULTSCAN] LVS INNER JOIN [Transformer_OM_BO].[dbo].[LVSD_STATIC_DATA_CURRENT_GTE5LT10] LVSD ON LVSD.[LVSD_FEEDER]=LVS.[LVS_FEEDER_NO] AND LVSD.[LVSD_VAULT_NUMBER]=LVS.[LVS_VAULT_NUMBER] AND (year([LVS_VAULT_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

"ElseIf cmbSearchFeeder.SelectedValue.Contains("X") Then

' sql = "SELECT count(*) AS missing_count FROM [Transformer_OM_BO].[dbo].[NETRMSXW_2WEEK_VAULTSCAN] LVS INNER JOIN [Transformer_OM_BO].[dbo].[LVSD_STATIC_DATA_CURRENT_GTE5LT10] LVSD ON LVSD.[LVSD_FEEDER]=LVS.[LVS_FEEDER_NO] AND

```
LVSD.[LVSD_VAULT_NUMBER]=LVS.[LVS_VAULT_NUMBER] AND
(year([LVS_VAULT_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

                ' ElseIf cmbSearchFeeder.SelectedValue.Contains("M") Then

                ' sql = "SELECT count(*) AS missing_count FROM
[Transformer_OM_BO].[dbo].[NETRMSM_2WEEK_VAULTSCAN] LVS INNER JOIN
[Transformer_OM_BO].[dbo].[LVSD_STATIC_DATA_CURRENT_GTE5LT10] LVSD ON
LVSD.[LVSD_FEEDER]=LVS.[LVS_FEEDER_NO] AND
LVSD.[LVSD_VAULT_NUMBER]=LVS.[LVS_VAULT_NUMBER] AND
(year([LVS_VAULT_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

                ' End If

                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable
```

```
            myCommand.Fill(dt)



            Dim dr As DataRow = dt.Rows(0)

            'NB: Grid views monitor Feeder to re-query!!!

            Missing_GTE5LT10.Text = drcheck(dr, "missing_count")



            myCommand = Nothing

            myConnection.Close()

            myConnection = Nothing

        End If



    End Sub



    Private Sub NetRms_Missing_Vault_GTE10()

        'transformer on feeder

        Dim sql As String

        '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found
```

```
If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If

    sql = "SELECT count(distinct([LVSD_VAULT_NUMBER])) AS missing_count
FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE10] WHERE LVSD_FEEDER =
'" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number  FROM
[Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

    Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

    Dim myCommand As New SqlDataAdapter(sql, myConnection)

    myCommand.SelectCommand.CommandTimeout = 180
```

```
        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        Missing_GTE10.Text = drcheck(dr, "missing_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub PlannedOpenMainsGT2_Feeder()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POM_NETRMALL_3MONTH_GT2] where LVSD_Feeder='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        'PlannedOMGT2.Text = drcheck(dr, "pom_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub

Private Sub PlannedOpenMainsLTE2_Feeder()

    'transformer on feeder

    Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POM_NETRMALL_3MONTH_LTE2] where LVSD_Feeder='" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

```
        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        'PlannedOMLTE2.Text = drcheck(dr, "pom_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub PlannedOpenMainsLTE2_GTE10()

    'transformer on feeder
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue



If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If



sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "'"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
        Dim myCommand As New SqlDataAdapter(sql, myConnection)

        myCommand.SelectCommand.CommandTimeout = 180

        Dim dt As DataTable = New DataTable

        myCommand.Fill(dt)



        Dim dr As DataRow = dt.Rows(0)

        'NB: Grid views monitor Feeder to re-query!!!

        'PlannedOMLTE2_GTE10.Text = drcheck(dr, "pom_count")



        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If



End Sub



Private Sub PlannedOpenMainsGT2_GTE10()
```

```
'transformer on feeder

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

If cmbSearchFeeder.Items.Count > 0 Then



  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue



  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If



  sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMGT2_NETRMSALL_3MONTH_GTE10] where LVSD_Feeder='" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

' PlannedOMGT2_GTE10.Text = drcheck(dr, "pom_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub
```

```
Private Sub PlannedOpenMainsGT2_GTE5LT10()

    'transformer on feeder

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then



      Dim Feeder_name As String

      Feeder_name = cmbSearchFeeder.SelectedValue



      If Feeder_name.Length < 5 Then

        Feeder_name = "0" + Feeder_name

      End If



      sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "'"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)

                'NB: Grid views monitor Feeder to re-query!!!

                ' PlannedOMGT2_GTE5LT10.Text = drcheck(dr, "pom_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub PlannedOpenMainsLTE2_GTE5LT10()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS pom_count FROM [Transformer_OM_BO].[dbo].[POMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_Feeder='" & Feeder_name & "'"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)

                'NB: Grid views monitor Feeder to re-query!!!

                'PlannedOMLTE2_GTE5LT10.Text = drcheck(dr, "pom_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub NetRms_ClosedOld_Vault_Scan_Feeder()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_FEEDER_NO]='" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)



Dim dr As DataRow = dt.Rows(0)



ClosedOld.Text = drcheck(dr, "closed_count")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If



End Sub
```

```
Private Sub NetRms_ClosedUNR_Vault_Scan_Feeder()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_FEEDER_NO]='" & Feeder_name & "'"
```

```
                Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim dt As DataTable = New DataTable

                myCommand.Fill(dt)



                Dim dr As DataRow = dt.Rows(0)



                ClosedUNR.Text = drcheck(dr, "closed_count")



                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

            End If



        End Sub
```

```
Private Sub NetRms_ClosedOld_Vault_Scan_GTE10()

  'transformer on feeder

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue



    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER]
```

= '" & Feeder_name & "'AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

ClosedOld_GTE10.Text = drcheck(dr, "closed_count")

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

End If

```
End Sub


Private Sub NetRms_ClosedOld_Vault_Scan_GTE5LT10()
  'transformer on feeder
  Dim sql As String
  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found
  If cmbSearchFeeder.Items.Count > 0 Then


    Dim Feeder_name As String
    Feeder_name = cmbSearchFeeder.SelectedValue


    If Feeder_name.Length < 5 Then
      Feeder_name = "0" + Feeder_name
    End If
```

```
sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)


Dim dr As DataRow = dt.Rows(0)


ClosedOld_GTE5LT10.Text = drcheck(dr, "closed_count")


myCommand = Nothing

myConnection.Close()
```

```
        myConnection = Nothing

      End If


    End Sub

    Private Sub NetRms_ClosedUNR_Vault_Scan_GTE10()

      'transformer on feeder

      Dim sql As String

      '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

      If cmbSearchFeeder.Items.Count > 0 Then


        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue


        If Feeder_name.Length < 5 Then

          Feeder_name = "0" + Feeder_name

        End If
```

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

ClosedUNR_GTE10.Text = drcheck(dr, "closed_count")

myCommand = Nothing

myConnection.Close()

```
        myConnection = Nothing

      End If


    End Sub


    Private Sub NetRms_ClosedUNR_Vault_Scan_GTE5LT10()

      'transformer on feeder

      Dim sql As String

      '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

      If cmbSearchFeeder.Items.Count > 0 Then


        Dim Feeder_name As String

        Feeder_name = cmbSearchFeeder.SelectedValue


        If Feeder_name.Length < 5 Then

          Feeder_name = "0" + Feeder_name
```

End If

sql = "SELECT count(*) AS closed_count FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER]='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

ClosedUNR_GTE5LT10.Text = drcheck(dr, "closed_count")

```
        myCommand = Nothing
        myConnection.Close()
        myConnection = Nothing
    End If

End Sub

Private Sub TargetInitialValues_LPW()
    Target_OpenSwitchesFeeder.Text = OpenSwitchesFeederCurrent.Text
    Target_OpenSwitches_GTE5LT10.Text = OpenSwitches_GTE5LT10.Text
    Target_OpenSwitches_GTE10.Text = OpenSwitches_GTE10.Text
    Target_BanksOffFeederCurrent.Text = BanksOffFeederCurrent.Text
    Target_BanksOff_GTE10.Text = BanksOff_GTE10.Text
    Target_BanksOff_GTE5LT10.Text = BanksOff_GTE5LT10.Text
    Target_FuseFeederCurrent.Text = FuseFeederCurrent.Text
    Target_Fuse_GTE10.Text = Fuse_GTE10.Text
    Target_Fuse_GTE5LT10.Text = Fuse_GTE5LT10.Text
    Target_MissingFeederCurrent.Text = MissingFeederCurrent.Text
```

```
Target_Missing_GTE10.Text = Missing_GTE10.Text

Target_Missing_GTE5LT10.Text = Missing_GTE5LT10.Text

Target_ClosedOld.Text = ClosedOld.Text

Target_ClosedOld_GTE10.Text = ClosedOld_GTE10.Text

Target_ClosedOld_GTE5LT10.Text = ClosedOld_GTE5LT10.Text

Target_ClosedUNR.Text = ClosedUNR.Text

Target_ClosedUNR_GTE10.Text = ClosedUNR_GTE10.Text

Target_ClosedUNR_GTE5LT10.Text = ClosedUNR_GTE5LT10.Text

Target_ClosedDNUCurrent.Text = ClosedDNUCurrent.Text

Target_ClosedDNU_GTE10.Text = ClosedDNU_GTE10.Text

Target_ClosedDNU_GTE5LT10.Text = ClosedDNU_GTE5LT10.Text

Target_OpenMainsLTE2Current.Text = OpenMainsLTE2Current.Text

Target_OMLTE2_GTE10.Text = OMLTE2_GTE10.Text

Target_OMLTE2_GTE5LT10.Text = OMLTE2_GTE5LT10.Text

Target_OpenMainsGT2Current.Text = OpenMainsGT2Current.Text

Target_OMGT2_GTE10.Text = OMGT2_GTE10.Text

Target_OMGT2_GTE5LT10.Text = OMGT2_GTE5LT10.Text

' Target_PlannedOMLTE2.Text = PlannedOMLTE2.Text
```

```
' Target_PlannedOMLTE2_GTE10.Text = PlannedOMLTE2_GTE10.Text

' Target_PlannedOMLTE2_GTE5LT10.Text = PlannedOMLTE2_GTE5LT10.Text

' Target_PlannedOMGT2.Text = PlannedOMGT2.Text

' Target_PlannedOMGT2_GTE10.Text = PlannedOMGT2_GTE10.Text

' Target_PlannedOMGT2_GTE5LT10.Text = PlannedOMGT2_GTE5LT10.Text

' Target_PlannedBanksOff.Text = PlannedBanksOff.Text

' Target_PlannedBanksOff_GTE10.Text = PlannedBanksOff_GTE10.Text

' Target_PlannedBanksOff_GTE5LT10.Text = PlannedBanksOff_GTE5LT10.Text


End Sub


Private Sub Total_LPW_Calc()


Sum_SwitchesOpen.Text = 5 * Int32.Parse(OpenSwitchesFeederCurrent.Text) + 5 *
Int32.Parse(OpenSwitches_GTE10.Text) + 3 * Int32.Parse(OpenSwitches_GTE5LT10.Text)


Sum_BanksOff.Text = 5 * Int32.Parse(BanksOffFeederCurrent.Text) + 5 *
Int32.Parse(BanksOff_GTE10.Text) + 3 * Int32.Parse(BanksOff_GTE5LT10.Text)
```

Sum_Fuse.Text = 5 * Int32.Parse(FuseFeederCurrent.Text) + 5 * Int32.Parse(Fuse_GTE10.Text) + 3 * Int32.Parse(Fuse_GTE5LT10.Text)

Sum_Missing.Text = Int32.Parse(MissingFeederCurrent.Text) + Int32.Parse(Missing_GTE10.Text) + Int32.Parse(Missing_GTE5LT10.Text)

Sum_ClosedOld.Text = Int32.Parse(ClosedOld.Text) + Int32.Parse(ClosedOld_GTE10.Text) + Int32.Parse(ClosedOld_GTE5LT10.Text)

Sum_ClosedUNR.Text = Int32.Parse(ClosedUNR.Text) + Int32.Parse(ClosedUNR_GTE10.Text) + Int32.Parse(ClosedUNR_GTE5LT10.Text)

Sum_ClosedDNU.Text = Int32.Parse(ClosedDNUCurrent.Text) + Int32.Parse(ClosedDNU_GTE10.Text) + Int32.Parse(ClosedDNU_GTE5LT10.Text)

Sum_OpenMainsLTE2.Text = Int32.Parse(OpenMainsLTE2Current.Text) + Int32.Parse(OMLTE2_GTE10.Text) + Int32.Parse(OMLTE2_GTE5LT10.Text)

Sum_OpenMainsGT2.Text = 2 * (Int32.Parse(OpenMainsGT2Current.Text) + Int32.Parse(OMGT2_GTE10.Text) + Int32.Parse(OMGT2_GTE5LT10.Text))

' Sum_PlannedOpenMainsLTE2.Text = Int32.Parse(PlannedOMLTE2.Text) + Int32.Parse(PlannedOMLTE2_GTE10.Text) + Int32.Parse(PlannedOMLTE2_GTE5LT10.Text)

'Sum_PlannedOpenMainsGT2.Text = 2 * (Int32.Parse(PlannedOMGT2.Text) + Int32.Parse(PlannedOMGT2_GTE10.Text) + Int32.Parse(PlannedOMGT2_GTE5LT10.Text))

' Sum_PlannedBanksOff.Text = 3 * (Int32.Parse(PlannedBanksOff.Text) + Int32.Parse(PlannedBanksOff_GTE10.Text) + Int32.Parse(PlannedBanksOff_GTE5LT10.Text))

TotalLPWCurrent.Text = Int32.Parse(Sum_SwitchesOpen.Text) + Int32.Parse(Sum_BanksOff.Text) + Int32.Parse(Sum_Fuse.Text) + Int32.Parse(Sum_Missing.Text) + Int32.Parse(Sum_ClosedOld.Text) + Int32.Parse(Sum_ClosedUNR.Text) + Int32.Parse(Sum_ClosedDNU.Text) + Int32.Parse(Sum_OpenMainsLTE2.Text) + Int32.Parse(Sum_OpenMainsGT2.Text)

' + Int32.Parse(Sum_PlannedOpenMainsLTE2.Text) + Int32.Parse(Sum_PlannedOpenMainsGT2.Text) + Int32.Parse(Sum_PlannedBanksOff.Text)

Transformer_Current.Text = 5 * (Int32.Parse(OpenSwitchesFeederCurrent.Text) + Int32.Parse(BanksOffFeederCurrent.Text) + Int32.Parse(FuseFeederCurrent.Text)) + Int32.Parse(MissingFeederCurrent.Text) + Int32.Parse(ClosedOld.Text) + Int32.Parse(ClosedUNR.Text) + Int32.Parse(ClosedDNUCurrent.Text) + Int32.Parse(OpenMainsLTE2Current.Text) + 2 * Int32.Parse(OpenMainsGT2Current.Text)

'+ Int32.Parse(PlannedOMLTE2.Text) + 2 * Int32.Parse(PlannedOMGT2.Text) + 3 * Int32.Parse(PlannedBanksOff.Text)

```
Transformer_NBY_GTE10.Text = 5 * (Int32.Parse(OpenSwitches_GTE10.Text) +
Int32.Parse(BanksOff_GTE10.Text) + Int32.Parse(Fuse_GTE10.Text)) + Int32.Parse(Missing_GTE10.Text) +
Int32.Parse(ClosedOld_GTE10.Text) + Int32.Parse(ClosedUNR_GTE10.Text) +
Int32.Parse(ClosedDNU_GTE10.Text) + Int32.Parse(OMLTE2_GTE10.Text) + 2 *
Int32.Parse(OMGT2_GTE10.Text)

' + Int32.Parse(PlannedOMLTE2_GTE10.Text) + 2 *
Int32.Parse(PlannedOMGT2_GTE10.Text) + 3 * Int32.Parse(PlannedBanksOff_GTE10.Text)

Transformer_NBY_GTE5LT10.Text = 3 * (Int32.Parse(OpenSwitches_GTE5LT10.Text)
+ Int32.Parse(BanksOff_GTE5LT10.Text) + Int32.Parse(Fuse_GTE5LT10.Text)) +
Int32.Parse(Missing_GTE5LT10.Text) + Int32.Parse(ClosedOld_GTE5LT10.Text) +
Int32.Parse(ClosedUNR_GTE5LT10.Text) + Int32.Parse(ClosedDNU_GTE5LT10.Text) +
Int32.Parse(OMLTE2_GTE5LT10.Text) + 2 * Int32.Parse(OMGT2_GTE5LT10.Text)

' + Int32.Parse(PlannedOMLTE2_GTE5LT10.Text) + 2 *
Int32.Parse(PlannedOMGT2_GTE5LT10.Text) + 3 * Int32.Parse(PlannedBanksOff_GTE5LT10.Text)

End Sub
```

```
Private Sub Target_Total_LPW_Calc()

Target_Sum_SwitchesOpen.Text = 5 * Int32.Parse(Target_OpenSwitchesFeeder.Text) + 5 * Int32.Parse(Target_OpenSwitches_GTE10.Text) + 3 * Int32.Parse(Target_OpenSwitches_GTE5LT10.Text)

Target_Sum_BanksOff.Text = 5 * Int32.Parse(Target_BanksOffFeederCurrent.Text) + 5 * Int32.Parse(Target_BanksOff_GTE10.Text) + 3 * Int32.Parse(Target_BanksOff_GTE5LT10.Text)

Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 * Int32.Parse(Target_Fuse_GTE10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)

Target_Sum_Missing.Text = Int32.Parse(Target_MissingFeederCurrent.Text) + Int32.Parse(Target_Missing_GTE10.Text) + Int32.Parse(Target_Missing_GTE5LT10.Text)

Target_Sum_ClosedOld.Text = Int32.Parse(Target_ClosedOld.Text) + Int32.Parse(Target_ClosedOld_GTE10.Text) + Int32.Parse(Target_ClosedOld_GTE5LT10.Text)

Target_Sum_ClosedUNR.Text = Int32.Parse(Target_ClosedUNR.Text) + Int32.Parse(Target_ClosedUNR_GTE10.Text) + Int32.Parse(Target_ClosedUNR_GTE5LT10.Text)
```

Target_Sum_ClosedDNU.Text = Int32.Parse(Target_ClosedDNUCurrent.Text) + Int32.Parse(Target_ClosedDNU_GTE10.Text) + Int32.Parse(Target_ClosedDNU_GTE5LT10.Text)

Target_Sum_OpenMainsLTE2.Text = Int32.Parse(Target_OpenMainsLTE2Current.Text) + Int32.Parse(Target_OMLTE2_GTE10.Text) + Int32.Parse(Target_OMLTE2_GTE5LT10.Text)

Target_Sum_OpenMainsGT2.Text = 2 * (Int32.Parse(Target_OpenMainsGT2Current.Text) + Int32.Parse(Target_OMGT2_GTE10.Text) + Int32.Parse(Target_OMGT2_GTE5LT10.Text))

'Target_Sum_PlannedOpenMainsLTE2.Text = Int32.Parse(Target_PlannedOMLTE2.Text) + Int32.Parse(Target_PlannedOMLTE2_GTE10.Text) + Int32.Parse(Target_PlannedOMLTE2_GTE5LT10.Text)

' Target_Sum_PlannedOpenMainsGT2.Text = 2 * (Int32.Parse(Target_PlannedOMGT2.Text) + Int32.Parse(Target_PlannedOMGT2_GTE10.Text) + Int32.Parse(Target_PlannedOMGT2_GTE5LT10.Text))

' Target_Sum_PlannedBanksOff.Text = 3 * (Int32.Parse(Target_PlannedBanksOff.Text) + Int32.Parse(Target_PlannedBanksOff_GTE10.Text) + Int32.Parse(Target_PlannedBanksOff_GTE5LT10.Text))

Target_TotalLPWCurrent.Text = Int32.Parse(Target_Sum_SwitchesOpen.Text) + Int32.Parse(Target_Sum_BanksOff.Text) + Int32.Parse(Target_Sum_Fuse.Text) + Int32.Parse(Target_Sum_Missing.Text) + Int32.Parse(Target_Sum_ClosedOld.Text) + Int32.Parse(Target_Sum_ClosedUNR.Text) + Int32.Parse(Target_Sum_ClosedDNU.Text) + Int32.Parse(Target_Sum_OpenMainsLTE2.Text) + Int32.Parse(Target_Sum_OpenMainsGT2.Text)

'+ Int32.Parse(Target_Sum_PlannedOpenMainsLTE2.Text) +
Int32.Parse(Target_Sum_PlannedOpenMainsGT2.Text) + Int32.Parse(Target_Sum_PlannedBanksOff.Text)

Target_Transformer_Current.Text = 5 * (Int32.Parse(Target_OpenSwitchesFeeder.Text) +
Int32.Parse(Target_BanksOffFeederCurrent.Text) + Int32.Parse(Target_FuseFeederCurrent.Text)) +
Int32.Parse(Target_MissingFeederCurrent.Text) + Int32.Parse(Target_ClosedOld.Text) +
Int32.Parse(Target_ClosedUNR.Text) + Int32.Parse(Target_ClosedDNUCurrent.Text) +
Int32.Parse(Target_OpenMainsLTE2Current.Text) + 2 * Int32.Parse(Target_OpenMainsGT2Current.Text)

' + Int32.Parse(Target_PlannedOMLTE2.Text) + 2 *
Int32.Parse(Target_PlannedOMGT2.Text) + 3 * Int32.Parse(Target_PlannedBanksOff.Text)

Target_Transformer_NBY_GTE10.Text = 5 *
(Int32.Parse(Target_OpenSwitches_GTE10.Text) + Int32.Parse(Target_BanksOff_GTE10.Text) +
Int32.Parse(Target_Fuse_GTE10.Text)) + Int32.Parse(Target_Missing_GTE10.Text) +
Int32.Parse(Target_ClosedOld_GTE10.Text) + Int32.Parse(Target_ClosedUNR_GTE10.Text) +
Int32.Parse(Target_ClosedDNU_GTE10.Text) + Int32.Parse(Target_OMLTE2_GTE10.Text) + 2 *
Int32.Parse(Target_OMGT2_GTE10.Text)

' + Int32.Parse(Target_PlannedOMLTE2_GTE10.Text) + 2 *
Int32.Parse(Target_PlannedOMGT2_GTE10.Text) + 3 * Int32.Parse(Target_PlannedBanksOff_GTE10.Text)

```
target_Transformer_NBY_GTE5LT10.Text = 3 *
(Int32.Parse(Target_OpenSwitches_GTE5LT10.Text) + Int32.Parse(Target_BanksOff_GTE5LT10.Text) +
Int32.Parse(Target_Fuse_GTE5LT10.Text)) + Int32.Parse(Target_Missing_GTE5LT10.Text) +
Int32.Parse(Target_ClosedOld_GTE5LT10.Text) + Int32.Parse(Target_ClosedUNR_GTE5LT10.Text) +
Int32.Parse(Target_ClosedDNU_GTE5LT10.Text) + Int32.Parse(Target_OMLTE2_GTE5LT10.Text) + 2 *
Int32.Parse(Target_OMGT2_GTE5LT10.Text)

' + Int32.Parse(Target_PlannedOMLTE2_GTE5LT10.Text) + 2 *
Int32.Parse(Target_PlannedOMGT2_GTE5LT10.Text) + 3 *
Int32.Parse(Target_PlannedBanksOff_GTE5LT10.Text)



End Sub

Private Sub LPW_Cost_Calculation()

closeOpenSwitches.Text = Int32.Parse(OpenSwitchesFeederCurrent.Text) -
Int32.Parse(Target_OpenSwitchesFeeder.Text) + Int32.Parse(OpenSwitches_GTE10.Text) -
Int32.Parse(Target_OpenSwitches_GTE10.Text) + Int32.Parse(OpenSwitches_GTE5LT10.Text) -
Int32.Parse(Target_OpenSwitches_GTE5LT10.Text)

closeOpenSwitches_TotalCost.Text = Int32.Parse(unitCost_CloseOpenSwitches.Text) *
Int32.Parse(closeOpenSwitches.Text)

closeBanksOff.Text = Int32.Parse(BanksOffFeederCurrent.Text) -
Int32.Parse(Target_BanksOffFeederCurrent.Text) + Int32.Parse(BanksOff_GTE10.Text) -
```

Int32.Parse(Target_BanksOff_GTE10.Text) + Int32.Parse(BanksOff_GTE5LT10.Text) - Int32.Parse(Target_BanksOff_GTE5LT10.Text)

closeBanksOff_TotalCost.Text = Int32.Parse(unitCost_CloseBanksOff.Text) * Int32.Parse(closeBanksOff.Text)

repairOpenFuse.Text = Int32.Parse(FuseFeederCurrent.Text) - Int32.Parse(Target_FuseFeederCurrent.Text) + Int32.Parse(Fuse_GTE10.Text) - Int32.Parse(Target_Fuse_GTE10.Text) + Int32.Parse(Fuse_GTE5LT10.Text) - Int32.Parse(Target_Fuse_GTE5LT10.Text)

repairOpenFuse_TotalCost.Text = Int32.Parse(unitCost_RepairOpenFuse.Text) * Int32.Parse(repairOpenFuse.Text)

repairOpenMains.Text = Int32.Parse(OpenMainsLTE2Current.Text) - Int32.Parse(Target_OpenMainsLTE2Current.Text) + Int32.Parse(OMLTE2_GTE10.Text) - Int32.Parse(Target_OMLTE2_GTE10.Text) + Int32.Parse(OMLTE2_GTE5LT10.Text) - Int32.Parse(Target_OMLTE2_GTE5LT10.Text) + Int32.Parse(OpenMainsGT2Current.Text) - Int32.Parse(Target_OpenMainsGT2Current.Text) + Int32.Parse(OMGT2_GTE10.Text) - Int32.Parse(Target_OMGT2_GTE10.Text) + Int32.Parse(OMGT2_GTE5LT10.Text) - Int32.Parse(Target_OMGT2_GTE5LT10.Text)

RepairOpenMains_TotalCost.Text = Int32.Parse(repairOpenMains.Text) * Int32.Parse(unitCost_RepairOpenMains.Text)

```
FixReporting.Text = Int32.Parse(ClosedDNUCurrent.Text) -
Int32.Parse(Target_ClosedDNUCurrent.Text) + Int32.Parse(ClosedUNR.Text) -
Int32.Parse(Target_ClosedUNR.Text) + Int32.Parse(ClosedOld.Text) - Int32.Parse(Target_ClosedOld.Text) +
Int32.Parse(MissingFeederCurrent.Text) - Int32.Parse(Target_MissingFeederCurrent.Text) +
Int32.Parse(ClosedDNU_GTE10.Text) - Int32.Parse(Target_ClosedDNU_GTE10.Text) +
Int32.Parse(ClosedUNR_GTE10.Text) - Int32.Parse(Target_ClosedUNR_GTE10.Text) +
Int32.Parse(ClosedOld_GTE10.Text) - Int32.Parse(Target_ClosedOld_GTE10.Text) +
Int32.Parse(Missing_GTE10.Text) - Int32.Parse(Target_Missing_GTE10.Text) +
Int32.Parse(ClosedDNU_GTE5LT10.Text) - Int32.Parse(Target_ClosedDNU_GTE5LT10.Text) +
Int32.Parse(ClosedUNR_GTE5LT10.Text) - Int32.Parse(Target_ClosedUNR_GTE5LT10.Text) +
Int32.Parse(ClosedOld_GTE5LT10.Text) - Int32.Parse(Target_ClosedOld_GTE5LT10.Text) +
Int32.Parse(Missing_GTE5LT10.Text) - Int32.Parse(Target_Missing_GTE5LT10.Text)

FixReporting_TotalCost.Text = Int32.Parse(FixReporting.Text) *
Int32.Parse(unitCost_FixReporting.Text)

totalNumberActions.Text = Int32.Parse(closeOpenSwitches.Text) +
Int32.Parse(closeBanksOff.Text) + Int32.Parse(repairOpenFuse.Text) + Int32.Parse(repairOpenMains.Text) +
Int32.Parse(FixReporting.Text)

upgrade_TotalCost.Text = Int32.Parse(closeOpenSwitches_TotalCost.Text) +
Int32.Parse(closeBanksOff_TotalCost.Text) + Int32.Parse(repairOpenFuse_TotalCost.Text) +
Int32.Parse(RepairOpenMains_TotalCost.Text) + Int32.Parse(FixReporting_TotalCost.Text)

End Sub
```

```
Private Function numericCheck(ByVal item As String) As String

    Dim num As Integer

    Dim result As Boolean = Int32.TryParse(item, num)

    If result Then

        If num < 0 Then Return "0" Else Return num

    Else : Return "0"

    End If


End Function
```

```
Private Sub Check_Target_Values()

    Target_OpenSwitchesFeeder.Text = numericCheck(Target_OpenSwitchesFeeder.Text)

    Target_OpenSwitches_GTE10.Text = numericCheck(Target_OpenSwitches_GTE10.Text)

    Target_OpenSwitches_GTE5LT10.Text = numericCheck(Target_OpenSwitches_GTE5LT10.Text)

    Target_BanksOffFeederCurrent.Text = numericCheck(Target_BanksOffFeederCurrent.Text)

    Target_BanksOff_GTE10.Text = numericCheck(Target_BanksOff_GTE10.Text)

    Target_BanksOff_GTE5LT10.Text = numericCheck(Target_BanksOff_GTE5LT10.Text)

    Target_FuseFeederCurrent.Text = numericCheck(Target_FuseFeederCurrent.Text)

    Target_Fuse_GTE5LT10.Text = numericCheck(Target_Fuse_GTE5LT10.Text)

    Target_Fuse_GTE10.Text = numericCheck(Target_Fuse_GTE10.Text)

    Target_MissingFeederCurrent.Text = numericCheck(Target_MissingFeederCurrent.Text)
```

```
Target_Missing_GTE5LT10.Text = numericCheck(Target_Missing_GTE5LT10.Text)

Target_Missing_GTE10.Text = numericCheck(Target_Missing_GTE10.Text)

Target_OpenMainsGT2Current.Text =
numericCheck(Target_OpenMainsGT2Current.Text)

Target_OMGT2_GTE5LT10.Text = numericCheck(Target_OMGT2_GTE5LT10.Text)

Target_OMGT2_GTE10.Text = numericCheck(Target_OMGT2_GTE10.Text)

Target_OpenMainsLTE2Current.Text =
numericCheck(Target_OpenMainsLTE2Current.Text)

Target_OMLTE2_GTE5LT10.Text = numericCheck(Target_OMLTE2_GTE5LT10.Text)

Target_OMLTE2_GTE10.Text = numericCheck(Target_OMLTE2_GTE10.Text)

Target_ClosedOld.Text = numericCheck(Target_ClosedOld.Text)

Target_ClosedOld_GTE5LT10.Text =
numericCheck(Target_ClosedOld_GTE5LT10.Text)

Target_ClosedOld_GTE10.Text = numericCheck(Target_ClosedOld_GTE10.Text)

Target_ClosedDNUCurrent.Text = numericCheck(Target_ClosedDNUCurrent.Text)

Target_ClosedDNU_GTE5LT10.Text =
numericCheck(Target_ClosedDNU_GTE5LT10.Text)

Target_ClosedDNU_GTE10.Text = numericCheck(Target_ClosedDNU_GTE10.Text)

Target_ClosedUNR.Text = numericCheck(Target_ClosedUNR.Text)
```

```
Target_ClosedUNR_GTE5LT10.Text = numericCheck(Target_ClosedUNR_GTE5LT10.Text)

Target_ClosedUNR_GTE10.Text = numericCheck(Target_ClosedUNR_GTE10.Text)

End Sub

Protected Sub LPW_Reset_Click(ByVal sender As Object, ByVal e As System.EventArgs)

'Protected Sub LPW_Submit_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LPW_Submit.Click

'Check_Target_Values()
```

```
'Target_Total_LPW_Calc()

'LPW_Cost_Calculation()

'' LPW_grid()

'GridviewdataBindBanksOff()

'GridViewdataBindOpenSwitches()

'GridViewdataBindFuse()

'GridViewDataBindClosedOld()

'GridViewDataBindClosedUNR()

'GridViewDataBindClosedDNU()

'GridViewDataBindOpenMainsLTE2()

'GridViewDataBindOpenMainsGT2()

' BindAttributes1()

reset_visitedLPWSessionVariables()

lpwreset()



End Sub

Private Sub GridviewdataBindBanksOff()
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT Vault_id, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH] where Feeder_ID = '" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "BO_NETRMSALL_3MONTH")



GV_BanksOff.DataSource = ds

GV_BanksOff.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewdataBindOpenSwitches()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name
```

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_vault_number,LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE ([Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%B%' OR [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT].[LVS_VAULT_CONDITION] like '%A%') AND LVS_FEEDER_NO = '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

```
myCommand.Fill(ds, "NetRMS_OpenSwitches")



GV_OpenSwitches.DataSource = ds

GV_OpenSwitches.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If









End Sub

Private Sub GridViewdataBindFuse()



  Dim sql As String
```

```
'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE LVS_VAULT_CONDITION like '%D%' AND LVS_FEEDER_NO='" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_Fuse")



GV_Fuse.DataSource = ds

GV_Fuse.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewDataBindClosedOld()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  '  If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT] WHERE [LVS_FEEDER_NO]= '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT")

```
        GV_ClosedOld.DataSource = ds

        GV_ClosedOld.DataBind()




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

        '  End If




End Sub


Private Sub GridViewDataBindClosedUNR()


    Dim sql As String
```

```
'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT] where lvs_feeder_no = '" & Feeder_name & "'"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT")




GV_ClosedUNR.DataSource = ds

GV_ClosedUNR.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewDataBindClosedDNU()


  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

  '  If cmbSearchFeeder.Items.Count > 0 Then



  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvs_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT] where lvs_Feeder_no = '" & Feeder_name & "'"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT")

GV_ClosedDNU.DataSource = ds

GV_ClosedDNU.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If



End Sub

Private Sub GridViewDataBindOpenMainsLTE2()



  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in(select nearest_vault_id from [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_LTE2] where LVSD_FEEDER='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180
```

```
        Dim ds As DataSet = New DataSet

        myCommand.Fill(ds, "OpenMainsLTE2")

        GV_OpenMainsLTE2.DataSource = ds



        GV_OpenMainsLTE2.DataBind()




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

        ' End If


End Sub


Private Sub GridViewDataBindOpenMainsGT2()
```

```
Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH] where LVSD_FEEDER='" & Feeder_name & "'
```

```
and nearest_vault_id in (select nearest_vault_id from [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_gt2] where LVSD_FEEDER='" & Feeder_name & "')"

                Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim ds As DataSet = New DataSet

                myCommand.Fill(ds, "OpenMainsGT2")

                GV_OpenMainsGT2.DataSource = ds



                GV_OpenMainsGT2.DataBind()




                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

                ' End If
```

```
End Sub

Private Sub GridviewdataBindMissing()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name
```

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

' sql = "SELECT Vault_id, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH] where Feeder_ID = '" & Feeder_name & "'"

sql = "SELECT lvs_vault_number, lvs_vault_scan_time_date FROM [Transformer_OM_BO].[dbo].[LVS_LAST_VAULT_SCAN_CURRENT] WHERE (year([LVS_vault_SCAN_TIME_DATE]) < 1970) AND (LVS_FEEDER_NO='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "Missing")

```
GV_Missing.DataSource = ds

GV_Missing.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If









End Sub



Private Sub GridviewdataBindBanksOff_NbyFactorGTE10()



  Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT LVSD_NBY_Vault_Number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "BO_NETRMSALL_3MONTHGTE10")




GV_BanksOff_NbyFactorGTE10.DataSource = ds

GV_BanksOff_NbyFactorGTE10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewdataBindOpenSwitches_NbyFactorGTE10()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE10] WHERE LVSD_FEEDER= '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_OpenSwitches_GTE10")

GV_OpenSwitches_NbyFactorGTE10.DataSource = ds

```
GV_OpenSwitches_NbyFactorGTE10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub

Private Sub GridViewdataBindFuse_NbyFactorGTE10()



Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE10] WHERE LVSD_FEEDER ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_FuseGTE10")



GV_Fuse_NbyFactorGTE10.DataSource = ds

GV_Fuse_NbyFactorGTE10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

'  End If



End Sub
```

```
Private Sub GridViewDataBindClosedOld_NbyFactorGTE10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    ' If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If



    'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM
[Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER]
= '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number  FROM
[Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds,
"CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")

GV_ClosedOld_NbyFactorGTE10.DataSource = ds

GV_ClosedOld_NbyFactorGTE10.DataBind()
```

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If

End Sub

Private Sub GridViewDataBindClosedUNR_NbyFactorGTE10()

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE10] WHERE [LVSD_FEEDER] ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number  FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)
```

```
                Dim myCommand As New SqlDataAdapter(sql, myConnection)

                myCommand.SelectCommand.CommandTimeout = 180

                Dim ds As DataSet = New DataSet

                myCommand.Fill(ds,
"CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")




                GV_ClosedUNR_NbyFactorGTE10.DataSource = ds

                GV_ClosedUNR_NbyFactorGTE10.DataBind()




                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

                ' End If
```

```
End Sub

Private Sub GridViewDataBindClosedDNU_NbyFactorGTE10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no
overloaded feeder for selected criteria found

    '  If cmbSearchFeeder.Items.Count > 0 Then


    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")

GV_ClosedDNU_NbyFactorGTE10.DataSource = ds

GV_ClosedDNU_NbyFactorGTE10.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If

End Sub

Private Sub GridViewDataBindMissing_NbyFactorGTE10()

Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number  FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"



Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)
```

```
myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "Missing_LVS_LAST_VAULT_SCAN_CURRENT_GTE10")

GV_Missing_NbyFactorGTE10.DataSource = ds

GV_Missing_NbyFactorGTE10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub
```

```
Private Sub GridViewDataBindOpenMainsLTE2_NbyFactorGTE10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    '  If cmbSearchFeeder.Items.Count > 0 Then



    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

       Feeder_name = "0" + Feeder_name

    End If



    'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "OpenMainsLTE2GTE10")

GV_OpenMainsLTE2_NbyFactorGTE10.DataSource = ds

GV_OpenMainsLTE2_NbyFactorGTE10.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If



End Sub


Private Sub GridViewDataBindOpenMainsGT2_NbyFactorGTE10()


  Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "OpenMainsGT2GTE10")

GV_OpenMainsGT2_NbyFactorGTE10.DataSource = ds



GV_OpenMainsGT2_NbyFactorGTE10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridviewdataBindBanksOff_NbyFactorGTE5LT10()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT LVSD_NBY_Vault_Number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[BO_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "BO_NETRMSALL_3MONTHGTE5LT10")

GV_BanksOff_NbyFactorGTE5LT10.DataSource = ds

GV_BanksOff_NbyFactorGTE5LT10.DataBind()

```
myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If



End Sub



Private Sub GridViewdataBindOpenSwitches_NbyFactorGTE5LT10()



Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then
```

```
Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_SwitchesOpen_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER= '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_OpenSwitchesGTE5LT10")



GV_OpenSwitches_NbyFactorGTE5LT10.DataSource = ds

GV_OpenSwitches_NbyFactorGTE5LT10.DataBind()



myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewdataBindFuse_NbyFactorGTE5LT10()


  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  '  If cmbSearchFeeder.Items.Count > 0 Then


  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then

    Feeder_name = "0" + Feeder_name

  End If


  'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"
```

```
sql = "SELECT lvsd_nby_vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Fuse_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "NetRMS_FuseGTE5LT10")


GV_Fuse_NbyFactorGTE5LT10.DataSource = ds

GV_Fuse_NbyFactorGTE5LT10.DataBind()


myCommand = Nothing

myConnection.Close()
```

```
myConnection = Nothing

' End If

End Sub

Private Sub GridViewDataBindClosedOld_NbyFactorGTE5LT10()

  Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String
```

```
Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

  Feeder_name = "0" + Feeder_name

End If
```

```
'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDOLD_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)
```

```
                myCommand.SelectCommand.CommandTimeout = 180

                Dim ds As DataSet = New DataSet

                myCommand.Fill(ds,
"CLOSEDOLD_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")




                GV_ClosedOld_NbyFactorGTE5LT10.DataSource = ds

                GV_ClosedOld_NbyFactorGTE5LT10.DataBind()




                myCommand = Nothing

                myConnection.Close()

                myConnection = Nothing

                ' End If
```

```
End Sub


Private Sub GridViewDataBindClosedUNR_NbyFactorGTE5LT10()


    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    '  If cmbSearchFeeder.Items.Count > 0 Then


    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name

    End If
```

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[CLOSEDUNR_LVS_LAST_VAULT_SCAN_GTE5LT10] WHERE [LVSD_FEEDER] ='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDUNR_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")

```
GV_ClosedUNR_NbyFactorGTE5LT10.DataSource = ds

GV_ClosedUNR_NbyFactorGTE5LT10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub

Private Sub GridViewDataBindClosedDNU_NbyFactorGTE5LT10()



  Dim sql As String

'--modified by shringika on dec 20 -2007 included if section to include the case when no
```

overloaded feeder for selected criteria found

```
' If cmbSearchFeeder.Items.Count > 0 Then



Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

   Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number,LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_ClosedDNU_NBY_FCTR_GTE5LT10] WHERE LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"
```

```
Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "CLOSEDDNU_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")

GV_ClosedDNU_NbyFactorGTE5LT10.DataSource = ds

GV_ClosedDNU_NbyFactorGTE5LT10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If
```

```
End Sub

Private Sub GridViewDataBindMissing_NbyFactorGTE5LT10()

    Dim sql As String

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    '  If cmbSearchFeeder.Items.Count > 0 Then

    Dim Feeder_name As String

    Feeder_name = cmbSearchFeeder.SelectedValue

    If Feeder_name.Length < 5 Then

      Feeder_name = "0" + Feeder_name
```

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_Vault_number, LVS_VAULT_SCAN_TIME_DATE FROM [Transformer_OM_BO].[dbo].[NETRMSALL_Missing_NBY_FCTR_GTE5LT10] WHERE [LVSD_FEEDER] = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 180

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "MISSING_LVS_LAST_VAULT_SCAN_CURRENT_GTE5LT10")

```
GV_Missing_NbyFactorGTE5LT10.DataSource = ds

GV_Missing_NbyFactorGTE5LT10.DataBind()




myCommand = Nothing

myConnection.Close()

myConnection = Nothing

' End If




End Sub

Private Sub GridViewDataBindOpenMainsLTE2_NbyFactorGTE5LT10()



  Dim sql As String
```

'--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

' If cmbSearchFeeder.Items.Count > 0 Then

Dim Feeder_name As String

Feeder_name = cmbSearchFeeder.SelectedValue

If Feeder_name.Length < 5 Then

Feeder_name = "0" + Feeder_name

End If

'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMLTE2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim ds As DataSet = New DataSet

myCommand.Fill(ds, "OpenMainsLTE2GTE5LT10")

GV_OpenMainsLTE2_NbyFactorGTE5LT10.DataSource = ds

GV_OpenMainsLTE2_NbyFactorGTE5LT10.DataBind()

myCommand = Nothing

myConnection.Close()

myConnection = Nothing

```
' End If

End Sub

Private Sub GridViewDataBindOpenMainsGT2_NbyFactorGTE5LT10()

  Dim sql As String

  '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

  ' If cmbSearchFeeder.Items.Count > 0 Then

  Dim Feeder_name As String

  Feeder_name = cmbSearchFeeder.SelectedValue

  If Feeder_name.Length < 5 Then
```

```
Feeder_name = "0" + Feeder_name

End If



'Dim sql As String = "SELECT count(*) as bo_count FROM [Transformer_OM_BO].[dbo].[BO_NETRMSBQ_3MONTH_NBY_FCTR_GTE5LT10] where Feeder_ID='" & cmbSearchFeeder.SelectedValue & "'"

sql = "SELECT lvsd_nby_vault_number,table_datetime FROM [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER = '" & Feeder_name & "' AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

'sql = "SELECT nearest_vault_id,table_datetime FROM [Transformer_OM_BO].[dbo].[OM_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "' and nearest_vault_id in (select lvsd_nby_vault_number from [Transformer_OM_BO].[dbo].[OMGT2_NETRMSALL_3MONTH_GTE5LT10] where LVSD_FEEDER='" & Feeder_name & "') AND LVSD_Vault_Number in (Select Vault_id as LVSD_Vault_Number FROM [Transformer_OM_BO].[dbo].[TargetVault_withCondition] where feeder_id='" & Feeder_name & "')"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300
```

```
        Dim ds As DataSet = New DataSet

        myCommand.Fill(ds, "OpenMainsGT2GTE5LT10")

        GV_OpenMainsGT2_NbyFactorGTE5LT10.DataSource = ds



        GV_OpenMainsGT2_NbyFactorGTE5LT10.DataBind()




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

        ' End If


    End Sub
```

```
Protected Sub RelPercent_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles RelPercent.SelectedIndexChanged

    RelXLPCalculationReset()

End Sub



Private Sub BindAttributes_Precomp()

    '--modified by shringika on dec 20 -2007 included if section to include the case when no overloaded feeder for selected criteria found

    If cmbSearchFeeder.Items.Count > 0 Then

        Dim fdrvalue As String

        fdrvalue = cmbSearchFeeder.SelectedValue

        'If cmbSearchFeeder.SelectedValue.Length < 5 Then

        'fdrvalue = "0" + cmbSearchFeeder.SelectedValue

        'End If
```

```
Dim sql As String = "SELECT * FROM Rerank_Attributes_PrecomMap_2008 WHERE
Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

Dim myConnection As New
SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

Dim dr As DataRow = dt.Rows(0)

'NB: Grid views monitor Feeder to re-query!!!

Feeder.Text = cmbSearchFeeder.SelectedValue

Feeder1.Text = cmbSearchFeeder.SelectedValue

Feeder2.Text = cmbSearchFeeder.SelectedValue
```

```
Feeder4.Text = cmbSearchFeeder.SelectedValue

' Feeder5.Text = cmbSearchFeeder.SelectedValue

Feeder6.Text = cmbSearchFeeder.SelectedValue

Feeder7.Text = cmbSearchFeeder.SelectedValue

feeder11.Text = cmbSearchFeeder.SelectedValue

feeder12.Text = cmbSearchFeeder.SelectedValue

feeder20.Text = cmbSearchFeeder.SelectedValue

Feeder_Name.Text = cmbSearchFeeder.SelectedValue

SelectedFeederName.Text = cmbSearchFeeder.SelectedValue




FLoad.Text = drcheck(dr, "mon_Load")

Normal_Rating.Text = drcheck(dr, "mon_Normal_Rating")

Emergency_Rating.Text = drcheck(dr, "mon_Emergency_Rating")

Shift.Text = drcheck(dr, "mon_Shift")

Delta.Text = drcheck(dr, "mon_Delta")

Emergency_Load.Text = drcheck(dr, "mon_Emergency_Load")

Feeders_4kv.Text = drcheck(dr, "mon_4kv")
```

```
Paper_GT40.Text = drcheck(dr, "mon_Paper_GT40")

Paper_LE40.Text = drcheck(dr, "mon_Paper_LE40")

'changed by shringika to exclude mon_solid_LT16 and mon_solid_GT16

Solid_LT16.Text = drcheck(dr, "mon_Solid_LT16")

Solid_70_75.Text = drcheck(dr, "mon_Solid_70_75")

Solid_GT16.Text = drcheck(dr, "mon_Solid_GT16")

Percent_Paper_GT40.Text = drcheck(dr, "mon_percent_Paper_GT40")

Percent_Paper_LE40.Text = drcheck(dr, "mon_percent_Paper_LE40")

Percent_Solid_LT16.Text = drcheck(dr, "mon_percent_Solid_LT16")

Percent_Solid_70_75.Text = drcheck(dr, "mon_percent_Solid_70_75")

Percent_Solid_GT16.Text = drcheck(dr, "mon_percent_Solid_GT16")

Sec_CNT.Text = drcheck(dr, "mon_Sec_CNT")

E21GT93.Text = drcheck(dr, "mon_E21GT93")

E21LE93.Text = drcheck(dr, "mon_E21LE93")

R31GT94.Text = drcheck(dr, "mon_R31GT94")

R31LE93.Text = drcheck(dr, "mon_R31LE93")

OtherStop_LT7.Text = drcheck(dr, "mon_OtherStop_LT7")

OtherStop7_12.Text = drcheck(dr, "mon_OtherStop7_12")
```

```
OtherStop_GT12.Text = drcheck(dr, "mon_OtherStop_GT12")

AllStop.Text = drcheck(dr, "mon_AllStop")

Percent_ALLStop.Text = drcheck(dr, "mon_percent_AllStop")

PJ_LT8.Text = drcheck(dr, "mon_PJ_LT8")

PJ8_16.Text = drcheck(dr, "mon_PJ8_16")

PJ16_24.Text = drcheck(dr, "mon_PJ16_24")

' PJ_GT24.Text = drcheck(dr, "mon_PJ_GT24")

PJ24_32.Text = drcheck(dr, "mon_PJ24-32")

PJ32_40.Text = drcheck(dr, "mon_PJ32-40")

PJGT40.Text = drcheck(dr, "mon_PJGT40")

AllPaper.Text = drcheck(dr, "mon_AllPaper")

Percent_AllPaper.Text = drcheck(dr, "mon_Percent_AllPaper")

SJ_LT8.Text = drcheck(dr, "mon_SJ_LT8")

SJ8_16.Text = drcheck(dr, "mon_SJ8_16")

SJ16_24.Text = drcheck(dr, "mon_SJ16_24")

SJ_GT24.Text = drcheck(dr, "mon_SJ_GT24")

AllSolid.Text = drcheck(dr, "mon_AllSolid")

Percent_AllSolid.Text = drcheck(dr, "mon_percent_AllSolid")
```

Percent_E21GT93.Text = drcheck(dr, "mon_percent_E21GT93")

Percent_E21LE93.Text = drcheck(dr, "mon_percent_E21LE93")

Percent_R31GT94.Text = drcheck(dr, "mon_percent_R31GT94")

Percent_R31LE93.Text = drcheck(dr, "mon_percent_R31LE93")

Percent_OtherStop_LT7.Text = drcheck(dr, "mon_percent_OtherStop_LT7")

Percent_OtherStop7_12.Text = drcheck(dr, "mon_percent_OtherStop7_12")

Percent_OtherStop_GT12.Text = drcheck(dr, "mon_percent_OtherStop_GT12")

Percent_PJ_LT8.Text = drcheck(dr, "mon_percent_PJ_LT8")

Percent_PJ8_16.Text = drcheck(dr, "mon_percent_PJ8_16")

Percent_PJ16_24.Text = drcheck(dr, "mon_percent_PJ16_24")

'Percent_PJ_GT24.Text = drcheck(dr, "mon_percent_PJ_GT24")

Percent_PJ24_32.Text = drcheck(dr, "mon_percent_PJ24-32")

Percent_PJ32_40.Text = drcheck(dr, "mon_percent_PJ32-40")

Percent_PJGT40.Text = drcheck(dr, "mon_percent_PJGT40")

Percent_SJ_LT8.Text = drcheck(dr, "mon_percent_SJ_LT8")

Percent_SJ8_16.Text = drcheck(dr, "mon_percent_SJ8_16")

Percent_SJ16_24.Text = drcheck(dr, "mon_percent_SJ16_24")

Percent_SJ_GT24.Text = drcheck(dr, "mon_percent_SJ_GT24")

```
Joint_CNT.Text = drcheck(dr, "mon_Joint_CNT")

XFMR_GT40.Text = drcheck(dr, "mon_XFMR_GT40")

XFMR30_39.Text = drcheck(dr, "mon_XFMR30_39")

XFMR20_29.Text = drcheck(dr, "mon_XFMR20_29")

XFMR10_19.Text = drcheck(dr, "mon_XFMR10_19")

XFMR0_9.Text = drcheck(dr, "mon_XFMR0_9")

W_XFMR_GT40.Text = drcheck(dr, "mon_W_XFMR_GT40")

W_XFMR30_39.Text = drcheck(dr, "mon_W_XFMR30_39")

W_XFMR20_29.Text = drcheck(dr, "mon_W_XFMR20_29")

W_XFMR10_19.Text = drcheck(dr, "mon_W_XFMR10_19")

W_XFMR0_9.Text = drcheck(dr, "mon_W_XFMR0_9")

Percent_XFMR_GT40.Text = drcheck(dr, "mon_percent_XFMR_GT40")

Percent_XFMR30_39.Text = drcheck(dr, "mon_percent_XFMR30_39")

Percent_XFMR20_29.Text = drcheck(dr, "mon_percent_XFMR20_29")

Percent_XFMR10_19.Text = drcheck(dr, "mon_percent_XFMR10_19")

Percent_XFMR0_9.Text = drcheck(dr, "mon_percent_XFMR0_9")

Percent_W_XFMR_GT40.Text = drcheck(dr, "mon_percent_W_XFMR_GT40")

Percent_W_XFMR30_39.Text = drcheck(dr, "mon_percent_W_XFMR30_39")
```

```
        Percent_W_XFMR20_29.Text = drcheck(dr, "mon_percent_W_XFMR20_29")

        Percent_W_XFMR10_19.Text = drcheck(dr, "mon_percent_W_XFMR10_19")

        Percent_W_XFMR0_9.Text = drcheck(dr, "mon_percent_W_XFMR0_9")

        XFMR_CNT_SUMMED.Text = drcheck(dr, "mon_XFMR_CNT_SUMMED")

        fJeopardy.Text = drcheck(dr, "Jeopardy")

        fCNT.Text = drcheck(dr, "CNT")




        myCommand = Nothing

        myConnection.Close()

        myConnection = Nothing

    End If

End Sub


Private Sub visited_sections_check(ByVal id As String, ByVal isSection As Boolean)


    Dim dvCablesInRun As DataView
```

```
Dim currYear As Integer

Dim j As Integer

Dim cableyear As String

Dim fromStr As String

Dim toStr As String

Dim feederId As String

currYear = 2008



'VS: New added

Dim insulation As String




If (isSection = False) Then

   'this function is more generic than the nam ovl lets on to

   dvCablesInRun = New DataView(GetOVLLoadTableForRun(id))

Else

   dvCablesInRun = New DataView(GetCableById(id))
```

```
End If

'SectionsSummary.Items.Add(dvCablesInRun.Item(0).Item("from_struct"))

For j = 0 To (dvCablesInRun.Table.Rows.Count - 1)
  If (dvCablesInRun.Item(j).Item("INSTALLATION_DATE").Equals(DBNull.Value))
Then

    cableYear = 1966
  Else
    cableYear = dvCablesInRun.Item(j).Item("INSTALLATION_DATE")
  End If

  feederId = dvCablesInRun.Item(j).Item("Feeder")
  ' if (isDBNull(dvCablesInRun.Item(j).Item("insulation")) then insulation 07/30/2008
what should we set if insulation = NULL Shringika
  If dvCablesInRun.Item(j).Item("insulation").Equals(DBNull.Value) Then
```

```
    insulation = "paper"
Else
    insulation = dvCablesInRun.Item(j).Item("insulation")
End If

If IsDBNull(dvCablesInRun.Item(j).Item("from_struct")) Then
    fromStr = 100000000
Else

    fromStr = dvCablesInRun.Item(j).Item("from_struct")
End If

If IsDBNull(dvCablesInRun.Item(j).Item("to_struct")) Then
    toStr = 100000000
Else
```

```
            toStr = dvCablesInRun.Item(j).Item("to_struct")

        End If




        'sections_area set up

        '

        ' If sections_area.InnerText.Contains(feederId + "." + fromStr.ToString + "." +
toStr.ToString) = False Then

        'lbldup.Text = "x"

        ' Else

        ' visited_sections_count = visited_sections_count + 1



        ' End If



        'vs2220:new changes, not yet reflected in vs2-rerank
```

```
'commented on 01/27/2009 for excluding visited sections_couint

'    If Not (Equals(SectionsSummary.Items.FindByText("No Sections"), Nothing)) Then
SectionsSummary.Items.Remove("No Sections")

' If (Equals(SectionsSummary.Items.FindByText(feederId + "." + fromStr.ToString + "."
+ toStr.ToString), Nothing)) Then

'lbldup.Text = "x"

' Else

' visited_sections_count = visited_sections_count + 1



' End If


Next


End Sub


Private Function GetRunsWithWorstStopTable(ByVal Feeder As String) As DataTable


If Feeder.Length < 5 Then
```

```
Feeder = "0" + Feeder.ToString

End If

' Dim sql As String = "SELECT [Run_ID],[rank] as worst_joint_rank, [elastimold],[RAYCHEM] ,[other_stops],[length], [stops] FROM [Rerank_PILC_Runs_Joints_Types_2008] WHERE feeder = '" & Feeder & "' and stops > 0 order by elastimold desc,raychem desc, other_stops desc "

'commented for deltra mtbf

Dim sql As String = "SELECT Runs.[Run_ID],Runs.[rank] as worst_joint_rank,Runs.[elastimold],Runs.[RAYCHEM] ,Runs.[other_stops],runs.[length], Runs.[stops],cast([delta] as decimal(4,1)) as delta FROM [Rerank_PILC_Runs_Joints_Types_2008] AS Runs INNER JOIN EdisonML.dbo.bulk_rerank_PILCRunID_precom AS Runs_Precomp ON Runs.Feeder= Runs_Precomp.feeder AND Runs.Run_Id=Runs_Precomp.Run_Id WHERE Runs.feeder = '" & cmbSearchFeeder.SelectedValue & "' and stops > 0 order by elastimold desc,raychem desc, other_stops desc"

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("EdisonML").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

'takes 30 secs to do!

myCommand.SelectCommand.CommandTimeout = 180
```

```
Dim dt As DataTable = New DataTable


myCommand.Fill(dt)


Dim totalColumn As DataColumn = New DataColumn
Dim perColumn As DataColumn = New DataColumn
Dim pertotColumn As DataColumn = New DataColumn
Dim secColumn As DataColumn = New DataColumn
With totalColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "tsec"
   '   .Expression = "sum(length)"
End With
With perColumn
   .DataType = System.Type.GetType("System.Decimal")
   .ColumnName = "psec"
End With
With pertotColumn
```

```
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "ptsec"
End With
With secColumn
    .DataType = System.Type.GetType("System.Decimal")
    .ColumnName = "sec"
End With
With dt.Columns
    .Add(totalColumn)
    .Add(perColumn)
    .Add(pertotColumn)
    .Add(secColumn)
End With




myCommand = Nothing
myConnection.Close()
```

```
        myConnection = Nothing


        Return dt

    End Function


    Protected Sub RelPercentElastimold_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles RelPercentElastimold.SelectedIndexChanged

        RelElastimoldCalculationReset()

    End Sub


    Private Sub RelElastimoldCalculationReset()

        If cmbSearchFeeder.Items.Count > 0 Then


            TargetCostCheck.Visible = False
```

```
SectionsChangedRelElastimold.Text = ""

TotalCostRelElastimold.Text = ""

'actualPercentPILC.Text = ""



UpdateAttributesElast.Visible = False

CommitChangesElast.Visible = False

CheckCommitElast.Checked = False

RankBaseRelElast.Text = ""

FOTBaseRelElast.Text = ""

MTBFBaseRelElast.Text = ""

RankNewRelElast.Text = ""

RankChangeRelElast.Text = ""

MTBFNewRelElast.Text = ""

MTBFChangeRelElast.Text = ""

FOTNewRelElast.Text = ""

FOTChangeRelElast.Text = ""



GVRelElastPanel.Visible = False
```

```
        RelGVElastPanel.Visible = False

        GVRelElastStops.Visible = False

        RelGVElast.Visible = False




    End If



End Sub

Private Sub RelXLPCalculationReset()

    If cmbSearchFeeder.Items.Count > 0 Then



        SectionsChangedRel.Text = ""

        TotalCostRel.Text = ""

        'actualPercentPILC.Text = ""



        UpdateAttributesRel.Visible = False
```

```
CommitChangesRel.Visible = False

CheckCommitRel.Checked = False

RankBaseRel.Text = ""

FOTBaseRel.Text = ""

MTBFBaseRel.Text = ""

RankNewRel.Text = ""

RankChangeRel.Text = ""

MTBFNewRel.Text = ""

MTBFChangeRel.Text = ""

FOTNewRel.Text = ""

FOTChangeRel.Text = ""



XLP.Visible = False

GridViewXLP.Visible = False




End If
```

```
End Sub

Private Function feeder_withzeroes_fix(ByVal feeder As String) As String

  If feeder.Length < 5 Then
    feeder = "0" + feeder.ToString
  End If
  Return feeder
End Function

Private Function NewAttributes_Check(ByVal item As String) As String
  If IsDBNull(item) Then Return "9999" Else
  If Int32.Parse(item) < 0 Then Return "0" Else Return item

End Function
```

```
Private Sub Update_Attributes_Fix_For_Negative_Attribute_Values(ByVal id As String,
ByVal isSection As Boolean)

    Dim totalJoints As Integer
    'Dim attributes As DataTable
    Dim fromtostructs As DataTable
    ' Dim dr As DataView
    Dim dr As DataRow
    Dim feeder_id As String
    feeder_id = cmbSearchFeeder.SelectedValue
    If feeder_id.Length < 5 Then feeder_id = "0" + feeder_id
    'Uattr.ChangeFeeder(feeder_id)

    If (isSection = False) Then
      'this function is more generic than the nam ovl lets on to
      'attributes = upd.GetAttributeVectors(id, False, feeder_id)

      dr = Uattr.CountMe(id, False)
```

```
    fromtostructs = Uattr.GetFromToStruct(id, False)

Else

    'attributes = upd.GetAttributeVectors(id, True, feeder_id)


    dr = Uattr.CountMe(id, True)

    fromtostructs = Uattr.GetFromToStruct(id, True)


End If


'mean imputation stats fon canonical feeders

'Insulationcnt avg_date

'NULL                    256     1979

'XLP                     45099   1983

'PAPER           24976           1959

'EPR                     58125   1999


Dim k As Integer

Dim dtr As DataRow
```

```
Dim mon_AllStop_forJointstotal, mon_AllPaper_ForJointsTotal As String

'commented on 08/16/2009 for new update attributes code


For k = 0 To (fromtostructs.Rows.Count - 1)

  dtr = fromtostructs.Rows(k)

  If IsDBNull(dtr("feeder")) = True And IsDBNull(dtr("from_struct")) = True And
IsDBNull(dtr("to_struct")) = True Then

    'Exit Sub

    Continue For

  End If


  If (Equals(SectionsSummary.Items.FindByText(dtr("feeder") + "." + dtr("from_struct")
+ "." + dtr("to_struct")), Nothing)) Then

    SectionsSummary.Items.Add(dtr("feeder") + "." + dtr("from_struct") + "." +
dtr("to_struct"))

    sections_area.InnerText += System.Environment.NewLine + dtr("feeder") + "." +
dtr("from_struct") + "." + dtr("to_struct")

    'Else

    'commented on 01/27/2009
```

```
'duplicate_sections_list = duplicate_sections_list + 1

' duplicate_sections_list = 0


End If

Next

'Updating Counts for cable sections

mon_Paper_GT40.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) - Int32.Parse(dr("iPaperGT40"))))

mon_Paper_LE40.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) - Int32.Parse(dr("iPaperLE40"))))

'dont need to add and subtract iSolidLT16 agb

mon_Solid_LT16.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Solid_LT16.Text) + Int32.Parse(dr("iPaperGT40")) +
Int32.Parse(dr("iPaperLE40")) + Int32.Parse(dr("isolidGE16")) + Int32.Parse(dr("iSolid70_75"))))

mon_Solid_GT16.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Solid_GT16.Text) - Int32.Parse(dr("isolidGE16"))))

mon_Solid_70_75.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_Solid_70_75.Text) - Int32.Parse(dr("iSolid70_75"))))
```

'Updating Counts for joints

'first updating total stop and paper joints for getting total joint counts at the end

mon_AllStop_forJointstotal = Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) + Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) + Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) + Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))

mon_AllPaper_ForJointsTotal = Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) + Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) + Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) + Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))

'commented on 04/05/2009

' mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) + Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) + Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) + Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))

```
'mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) - Int32.Parse(dr("papLT8")) + Int32.Parse(mon_PJ8_16.Text) - Int32.Parse(dr("papBet8A16")) + Int32.Parse(mon_PJ16_24.Text) - Int32.Parse(dr("papBet16A24")) + Int32.Parse(mon_PJ24_32.Text) - Int32.Parse(dr("papBet24A32")) + Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40")) + Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40")))

'modificat6ion on 04/05/2009 ends here


'Elastimold Stop - just remove, don't replace


mon_E21GT93.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_E21GT93.Text) - Int32.Parse(dr("elasGT1993"))))

mon_E21LE93.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993"))))


'Raychem Stop - just remove, don't replace

mon_R31GT94.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993"))))

mon_R31LE93.Text = NewAttributes_Check(Convert.ToString(Int32.Parse(mon_R31LE93.Text) - Int32.Parse(dr("rayLE1993"))))
```

```
'Other Stop - just remove, don't replace

mon_OtherStop_LT7.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7"))))

mon_OtherStop7_12.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_OtherStop7_12.Text) -
Int32.Parse(dr("oStopBet7A12"))))

mon_OtherStop_GT12.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_OtherStop_GT12.Text) -
Int32.Parse(dr("oStopGT12"))))



'Updating total stops

'enabled on 04/05/2009

mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) +
Int32.Parse(mon_E21LE93.Text) + Int32.Parse(mon_R31GT94.Text) + Int32.Parse(mon_R31LE93.Text) +
Int32.Parse(mon_OtherStop_LT7.Text) + Int32.Parse(mon_OtherStop7_12.Text) +
Int32.Parse(mon_OtherStop_GT12.Text))

'mon_AllStop.Text = Convert.ToString(Int32.Parse(mon_E21GT93.Text) -
Int32.Parse(dr("elasGT1993")) + Int32.Parse(mon_E21LE93.Text) - Int32.Parse(dr("elasLE1993")) +
Int32.Parse(mon_R31GT94.Text) - Int32.Parse(dr("rayGT1993")) + Int32.Parse(mon_R31LE93.Text) -
```

```
Int32.Parse(dr("rayLE1993")) + Int32.Parse(mon_OtherStop_LT7.Text) - Int32.Parse(dr("oStopLT7")) +
Int32.Parse(mon_OtherStop7_12.Text) - Int32.Parse(dr("oStopBet7A12")) +
Int32.Parse(mon_OtherStop_GT12.Text) - Int32.Parse(dr("oStopGT12")))


                'Paper - replace

                mon_PJ_LT8.Text =
NewAttributes_Check((Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) - Int32.Parse(dr("papLT8")))))

                mon_PJ8_16.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ8_16.Text) - Int32.Parse(dr("papBet8A16"))))

                mon_PJ16_24.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ16_24.Text) - Int32.Parse(dr("papBet16A24"))))

                mon_PJ24_32.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ24_32.Text) - Int32.Parse(dr("papBet24A32"))))

                mon_PJ32_40.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40"))))

                mon_PJGT40.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40"))))


                'Updating total paper

                '               mon_AllPaper.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) + Int32.Parse(mon_PJ8_16.Text) +
```

```
Int32.Parse(mon_PJ16_24.Text) + Int32.Parse(mon_PJ24_32.Text) + Int32.Parse(mon_PJ32_40.Text) +
Int32.Parse(mon_PJGT40.Text)))

' mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) -
Int32.Parse(dr("papLT8")) + Int32.Parse(mon_PJ8_16.Text) - Int32.Parse(dr("papBet8A16")) +
Int32.Parse(mon_PJ16_24.Text) - Int32.Parse(dr("papBet16A24")) + Int32.Parse(mon_PJ24_32.Text) -
Int32.Parse(dr("papBet24A32")) + Int32.Parse(mon_PJ32_40.Text) - Int32.Parse(dr("papBet32A40")) +
Int32.Parse(mon_PJGT40.Text) - Int32.Parse(dr("papGE40")))

mon_AllPaper.Text = Convert.ToString(Int32.Parse(mon_PJ_LT8.Text) +
Int32.Parse(mon_PJ8_16.Text) + Int32.Parse(mon_PJ16_24.Text) + Int32.Parse(mon_PJ24_32.Text) +
Int32.Parse(mon_PJ32_40.Text) + Int32.Parse(mon_PJGT40.Text))

'Solid - update date

mon_SJ8_16.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ8_16.Text) - Int32.Parse(dr("solBet8A16"))))

mon_SJ16_24.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ16_24.Text) - Int32.Parse(dr("solBet16A24"))))

mon_SJ_GT24.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ_GT24.Text) - Int32.Parse(dr("solGT24"))))

'Finally, adding all to be replaced and date updated to solid
```

```
                mon_SJ_LT8.Text =
NewAttributes_Check(Convert.ToString(Int32.Parse(mon_SJ_LT8.Text) + Int32.Parse(dr("papLT8")) +
Int32.Parse(dr("papBet8A16")) + Int32.Parse(dr("papBet16A24")) + Int32.Parse(dr("papBet24A32")) +
Int32.Parse(dr("papBet32A40")) + Int32.Parse(dr("papGE40")) + Int32.Parse(dr("solBet8A16")) +
Int32.Parse(dr("solBet16A24")) + Int32.Parse(dr("solGT24")) + Int32.Parse(dr("leadWipeCount")) +
Int32.Parse(dr("raychem1w1w")) + Int32.Parse(dr("raychem3w1w")) + Int32.Parse(dr("other1w1w"))))

                ' Next

                'Updating Section cost text boxes

                SectionsChangedSum.Text = SectionsSummary.Items.Count

                CostPerSectionSum.Text = "18000"

                TotalCostSum.Text = SectionsSummary.Items.Count * 18000

                'Updating percentages for cable sections

                mon_Percent_Paper_GT40.Text = Convert.ToString(Int32.Parse(mon_Paper_GT40.Text) /
Int32.Parse(Sec_CNT.Text) * 100.0)
```

```
mon_Percent_Paper_LE40.Text = Convert.ToString(Int32.Parse(mon_Paper_LE40.Text) /
Int32.Parse(Sec_CNT.Text) * 100.0)

mon_Percent_Solid_LT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_LT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_GT16.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_GT16.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

mon_Percent_Solid_70_75.Text =
Convert.ToString(Math.Round(Int32.Parse(mon_Solid_70_75.Text) / Int32.Parse(Sec_CNT.Text) * 100.0, 1))

'Updating total solids

mon_AllSolid.Text = Convert.ToString(Int32.Parse(mon_SJ8_16.Text) +
Int32.Parse(mon_SJ16_24.Text) + Int32.Parse(mon_SJ_GT24.Text) + Int32.Parse(mon_SJ_LT8.Text))

'updating total joints

mon_Joint_CNT.Text = Convert.ToString(Int32.Parse(mon_AllStop.Text) +
Int32.Parse(mon_AllPaper.Text) + Int32.Parse(mon_AllSolid.Text))
```

```
totalJoints = mon_Joint_CNT.Text



'Updating percentages

'Elastimold Stop Percentages

mon_Percent_E21GT93.Text = Convert.ToString((Int32.Parse(mon_E21GT93.Text) /
totalJoints) * 100)

mon_Percent_E21LE93.Text = Convert.ToString((Int32.Parse(mon_E21LE93.Text) /
totalJoints) * 100)



'Raychem Stop percentages

mon_Percent_R31GT94.Text = Convert.ToString((Int32.Parse(mon_R31GT94.Text) /
totalJoints) * 100)

mon_Percent_R31LE93.Text = Convert.ToString((Int32.Parse(mon_R31LE93.Text) /
totalJoints) * 100)



'Other Stop percentages

mon_Percent_OtherStop_LT7.Text =
Convert.ToString((Int32.Parse(mon_OtherStop_LT7.Text) / totalJoints) * 100)
```

```
                mon_Percent_OtherStop7_12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop7_12.Text) / totalJoints) * 100)

                mon_Percent_OtherStop_GT12.Text =
Convert.ToString((Int32.Parse(mon_OtherStop_GT12.Text) / totalJoints) * 100)


                'Updating total stop percentages

                mon_Percent_ALLStop.Text = Convert.ToString((Int32.Parse(mon_AllStop.Text) /
totalJoints) * 100)


                'Paper percentages

                mon_Percent_PJ_LT8.Text = Convert.ToString((Int32.Parse(mon_PJ_LT8.Text) /
totalJoints) * 100)

                mon_Percent_PJ8_16.Text = Convert.ToString((Int32.Parse(mon_PJ8_16.Text) /
totalJoints) * 100)

                mon_Percent_PJ16_24.Text = Convert.ToString((Int32.Parse(mon_PJ16_24.Text) /
totalJoints) * 100)

                mon_Percent_PJ24_32.Text = Convert.ToString((Int32.Parse(mon_PJ24_32.Text) /
totalJoints) * 100)

                mon_Percent_PJ32_40.Text = Convert.ToString((Int32.Parse(mon_PJ32_40.Text) /
totalJoints) * 100)
```

```
                mon_Percent_PJGT40.Text = Convert.ToString((Int32.Parse(mon_PJGT40.Text) /
totalJoints) * 100)

                'Updating total paper percentages

                mon_Percent_AllPaper.Text = Convert.ToString((Int32.Parse(mon_AllPaper.Text) /
totalJoints) * 100)

                'mon_Percent_AllPaper.Text =
Convert.ToString((Int32.Parse(mon_AllPaper_ForJointsTotal) / totalJoints) * 100)

                'Solid percentages

                mon_Percent_SJ_LT8.Text = Convert.ToString((Int32.Parse(mon_SJ_LT8.Text) /
totalJoints) * 100)

                mon_Percent_SJ8_16.Text = Convert.ToString((Int32.Parse(mon_SJ8_16.Text) /
totalJoints) * 100)

                mon_Percent_SJ16_24.Text = Convert.ToString((Int32.Parse(mon_SJ16_24.Text) /
totalJoints) * 100)

                mon_Percent_SJ_GT24.Text = Convert.ToString((Int32.Parse(mon_SJ_GT24.Text) /
totalJoints) * 100)

                'Updating total solids
```

```
mon_Percent_AllSolid.Text = Convert.ToString((Int32.Parse(mon_AllSolid.Text) / totalJoints) * 100)

End Sub

Protected Sub CalculateLPWMTBF_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles CalculateLPWMTBF.Click

If TotalLPWCurrent.Text = "" Then Exit Sub

'Dim myService As Maxim = New Maxim()

Dim feeder_Id As String

feeder_Id = New String("X")

feeder_Id = cmbSearchFeeder.SelectedValue

If feeder_Id.Length < 5 Then

  feeder_Id = "0" + feeder_Id

End If
```

```
Dim rms_weight, target_rms_weight As String

rms_weight = Convert.ToString(Int32.Parse(Sum_ClosedOld.Text) + Int32.Parse(Sum_ClosedUNR.Text))

Dim data_new_rank, data_base_rank, base_rank As String


data_base_rank = "feeder=" + feeder_Id.ToString + ";"

data_base_rank += "LPW_SUM_Avg_last_04_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_04_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_04_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_45_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_45_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_45_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_24_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_24_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_24_hrs=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_07_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_07_days=" + TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_07_days=" + TotalLPWCurrent.Text + ";" + "LPW_RMS_Avg_last_04_hrs=" + rms_weight + ";" + "LPW_RMS_Min_last_04_hrs=" + rms_weight + ";" + "LPW_RMS_Max_last_04_hrs=" + rms_weight + ";" + "LPW_RMS_Avg_last_45_days=" + rms_weight + ";" + "LPW_SUM_Min_last_45_days=" + rms_weight + ";" + "LPW_SUM_Max_last_45_days=" + rms_weight + ";" + "LPW_RMS_Avg_last_24_hrs=" + rms_weight + ";" + "LPW_RMS_Min_last_24_hrs=" + rms_weight + ";" + "LPW_RMS_Max_last_24_hrs=" + rms_weight + ";" + "LPW_RMS_Avg_last_07_days=" + rms_weight + ";" + "LPW_RMS_Min_last_07_days=" + rms_weight + ";" + "LPW_RMS_Max_last_07_days=" + rms_weight

'base_rank = myService.WSRerank(data_base_rank)

'base_rank = proxy.rerank(data_base_rank.ToString)
```

'data_base_rank =

"feeder=02M43;Static_Feeder_Jeopardy=5.12;Static_Feeder_Load=365.0;Static_Emergency_Proj_Load=598.0;Stat
ic_Normal_rating=418.0;Static_Emergency_rating=613.0;Static_Shift=4.17;Static_Four_KV=0;Static_OA_CNT=0;
Static_delta=56.29;Static_TRANSFORMERS=18.0;Static_Feeds_TA=0;Static_Num_Shunt=0;Static_hipot_OA=3;
Static_hipot_freq_Paper_gte_40=0.377000004053;Static_hipot_freq_Paper_Other=0.0719999969006;Static_hipot_f
req_Solid_70_75=0.00600000005215;Static_hipot_freq_Solid_LT16=0.0430000014603;Static_hipot_freq_Solid_G
TE16=0.0359999984503;Static_hipot_freq_E21_pre93=0.0;Static_hipot_freq_Other_stops=0.122000001371;Static
_hipot_freq_R31_post93=0.25799998641;Static_hipot_freq_Paper_ns=0.234999999404;Static_hipot_freq_Solid_ns
=0.01600000076;Static_hipot_freq_Sum=1.22000002861;Static_hipot_freq_Norm_Sum=0.127000004053;Static_hi
pot_Norm_Shift=0.173999994993;Static_hipot_Norm_Delta=0.910000026226;Static_hipot_Norm_OA=0.5;Static_
hipot_freq_Wtd_Nor=0.389999985695;Static_hipot_GT_half=1;Static_hipot_Norm_WT_Nrm=0.683000028133;St
atic_hipot_freq_E21_post93=0.0;Static_hipot_freq_R31_pre93=0.0560000017285;Static_PaperC_GE_40=41;Static
_PaperOther=9;Static_PaperC_Total=50;Static_Solid_LT_16=40;Static_Solid_70_75=2;Static_Solid_Total=58;Stat
ic_Solid_GE_16=16;Static_total_sections=108;Static_Percent_PaperC_GE_40=37.962962963;Static_Percent_Pape
rOther=8.33333333333;Static_Percent_PaperC_Total=46.2962962963;Static_Percent_Solid_LT_16=37.037037037
;Static_Percent_Solid_70_75=1.85185185185;Static_Percent_Solid_GE_16=14.8148148148;Static_Percent_Solid_
Total=53.7037037037;Static_E2_1_LE_1993=0;Static_E2_1_GT_1993=0;Static_E2_1_Total=0;Static_R3_1_LE_1
993=2;Static_R3_1_GT_1993=12;Static_R3_1_Total=14;Static_Other_Stops_LT_7=3;Static_Other_Stops_7_12=2
;Static_Other_Stops_GT_12=10;Static_Other_Stops_Total=15;Static_Total_Stops=29;Static_PAPER_LT_8=0;Stati
c_PAPER_8_16=0;Static_PAPER_16_24=2;Static_PAPER_24_32=1;Static_PAPER_32_40=5;Static_PAPER_GT
_40=24;Static_PAPER_Total=32;Static_SolidJoints_LT_8=29;Static_SolidJoints_8_16=5;Static_SolidJoints_16_24
=4;Static_SolidJoints_GT_24=0;Static_SolidJoints_Total=38;Static_total_joints=99;Static_Percent_E2_1_LE_1993
=0.0;Static_Percent_E2_1_GT_1993=0.0;Static_Percent_E2_1_Total=0.0;Static_Percent_R3_1_LE_1993=2.02020
20202;Static_Percent_R3_1_GT_1993=12.1212121212;Static_Percent_R3_1_Total=14.1414141414;Static_Percent
_Other_Stops_LT_7=3.0303030303;Static_Percent_Other_Stops_7_12=2.0202020202;Static_Percent_Other_Stops
_GT_12=10.101010101;Static_Percent_Other_Stops_Total=15.1515151515;Static_Percent_Total_Stops=29.29292
92929;Static_Percent_PAPER_LT_8=0.0;Static_Percent_PAPER_8_16=0.0;Static_Percent_PAPER_16_24=2.0202

020202;Static_Percent_PAPER_24_32=1.0101010101;Static_Percent_PAPER_32_40=5.05050505051;Static_Perce
nt_PAPER_GT_40=24.2424242424;Static_Percent_PAPER_Total=32.3232323232;Static_Percent_SolidJoints_LT
_8=29.2929292929;Static_Percent_SolidJoints_8_16=5.05050505051;Static_Percent_SolidJoints_16_24=4.040404
0404;Static_Percent_SolidJoints_GT_24=0.0;Static_Percent_SolidJoints_Total=38.3838383838;Static_Transformer
_0_9=6;Static_Transformer_10_19=3;Static_Transformer_20_29=1;Static_Transformer_30_39=3;Static_Transform
er_GE_40=2;Static_Transformer_ALL=15;Static_West_Transformer_0_9=0;Static_West_Transformer_10_19=2;St
atic_West_Transformer_20_29=0;Static_West_Transformer_30_39=0;Static_West_Transformer_GE_40=0;Static_
West_Transformer_ALL=2;Static_total_transformers=17;Static_Percent_Transformer_0_9=35.2941176471;Static_
Percent_Transformer_10_19=17.6470588235;Static_Percent_Transformer_20_29=5.88235294118;Static_Percent_
Transformer_30_39=17.6470588235;Static_Percent_Transformer_GE_40=11.7647058824;Static_Percent_Transfor
mer_ALL=88.2352941176;Static_Percent_West_Transformer_0_9=0.0;Static_Percent_West_Transformer_10_19=
11.7647058824;Static_Percent_West_Transformer_20_29=0.0;Static_Percent_West_Transformer_30_39=0.0;Static
_Percent_West_Transformer_GE_40=0.0;Static_Percent_West_Transformer_ALL=11.7647058824;Static_SH1_CO
UNT=0;Static_SH1_TOTAL=0.0;Static_SH3_COUNT=1;Static_SH3_TOTAL=15.0;Static_SH5_COUNT=5;Static
_SH5_TOTAL=46.41;Static_SH10_COUNT=7;Static_SH10_TOTAL=57.83;Static_SH16_COUNT=16;Static_SH
16_TOTAL=61.77;Static_SH32_COUNT=27;Static_SH32_TOTAL=63.42;Static_SH1_AVG=0.0;Static_SH3_AV
G=15.0;Static_SH5_AVG=3.0;Static_SH10_AVG=8.26142857143;Static_SH16_AVG=3.860625;Static_SH32_AV
G=2.34888888889;Static_Feeder_CNT=28;Static_Sum_LPW_20050921=30.0;Static_RMS_LPW_20050921=10.5;
Static_Avg_LPW_20050921=1.76470588235;Static_Sum_LPW_20050824=25.5;Static_RMS_LPW_20050824=6.
5;Static_Avg_LPW_20050824=1.5;Static_Sum_LPW_20050816=43.0;Static_RMS_LPW_20050816=12.0;Static_S
um_LPW_20050802=33.0;Static_RMS_LPW_20050802=9.0;Static_Avg_LPW_20050802=1.94117647059;Static_
Avg_LPW_20050816=2.52941176471;Static_Sum_LPW_20050614=49.0;Static_RMS_LPW_20050614=16.0;Stati
c_Avg_LPW_20050614=3.0625;Static_Sections=110;Static_CSR_LE_5percent=0;Static_CSR_LE_10percent=0;St
atic_CSR_LE_15percent=0;Static_CSR_LE_20percent=0;Static_CSR_LE_25percent=1;Static_CSR_LE_30percent
=11;Static_Percent_CSR_LE_5percent=0.0;Static_Percent_CSR_LE_10percent=0.0;Static_Percent_CSR_LE_15pe
rcent=0.0;Static_Percent_CSR_LE_20percent=0.0;Static_Percent_CSR_LE_25percent=0.90909090909;Static_Perc
ent_CSR_LE_30percent=10.0;Static_joints=99;Static_JR_LE_5percent=17;Static_JR_LE_10percent=29;Static_JR_

LE_15percent=35;Static_JR_LE_20percent=38;Static_JR_LE_25percent=56;Static_JR_LE_30percent=61;Static_Percent_JR_LE_5percent=17.1717171717;Static_Percent_JR_LE_10percent=29.2929292929;Static_Percent_JR_LE_15percent=35.3535353535;Static_Percent_JR_LE_20percent=38.3838383838;Static_Percent_JR_LE_25percent=56.5656565657;Static_Percent_JR_LE_30percent=61.6161616162;Static_mon_AllStop=29;Static_mon_percent_AllStop=29.2929292929;Static_mon_AllPaper=32;Static_mon_percent_AllPaper=32.3232323232;Static_mon_AllSolid=34;Static_mon_percent_AllSolid=34.3434343434;OUT_Auto_full_history=4;OUT_Auto_allprev_season=1;OUT_Auto_prev_season=0;OUT_Auto_last_45days=0;OUT_Auto_last_7days=0;OUT_CIOA_full_history=0;OUT_CIOA_allprev_season=0;OUT_CIOA_prev_season=0;OUT_CIOA_last_45days=0;OUT_CIOA_last_7days=0;OUT_FOT_full_history=0;OUT_FOT_allprev_season=0;OUT_FOT_prev_season=0;OUT_FOT_last_45days=0;OUT_FOT_last_7days=0;OUT_OOE_full_history=6;OUT_OOE_allprev_season=1;OUT_OOE_prev_season=1;OUT_OOE_last_45days=1;OUT_OOE_last_7days=1;OUT_Other_full_history=14;OUT_Other_allprev_season=8;OUT_Other_prev_season=0;OUT_Other_last_45days=0;OUT_Other_last_7days=0;LPW_SUM_Min_last_45_days=1.0;LPW_SUM_Min_last_07_days=6.0;LPW_SUM_Min_last_24_hrs=10.0;LPW_SUM_Min_last_04_hrs=10.0;LPW_SUM_Max_last_45_days=37.0;LPW_SUM_Max_last_07_days=37.0;LPW_SUM_Max_last_24_hrs=10.0;LPW_SUM_Max_last_04_hrs=10.0;LPW_SUM_Avg_last_45_days=9.0;LPW_SUM_Avg_last_07_days=11.0;LPW_SUM_Avg_last_24_hrs=10.0;LPW_SUM_Avg_last_04_hrs=10.0;LPW_RMS_Min_last_45_days=?;LPW_RMS_Min_last_07_days=1.0;LPW_RMS_Min_last_24_hrs=1.0;LPW_RMS_Min_last_04_hrs=1.0;LPW_RMS_Max_last_45_days=33.0;LPW_RMS_Max_last_07_days=33.0;LPW_RMS_Max_last_24_hrs=1.0;LPW_RMS_Max_last_04_hrs=1.0;LPW_RMS_Avg_last_45_days=2.0;LPW_RMS_Avg_last_07_days=2.0"

```
base_rank = WebserviceClient.WebserviceClient.proxy_rerank(data_base_rank.ToString)

'lpwoutcome.Text = Chr(34) + data_base_rank.ToString + Chr(34)

lpwoutcome.Text = base_rank.ToString

Dim i As Integer

i = base_rank.IndexOf(",")
```

```
RankBaseLPW.Text = base_rank.Substring(i + 1)

Dim score As Double

score = Double.Parse(base_rank.Substring(0, i - 1))

'MTBFBaseLPW.Text = -(112.33 * score) + 278.71

MTBFBaseLPW.Text = (1.1733 * score) + 1.6807

' MTBFBaseLPW.Text = 1 / Math.Round(1.1638 * score + 1.6793, 3)

'1/round(1.1638*BR.SCORE+1.6793,3)

Dim lpwmtbf_base As Integer

lpwmtbf_base = Math.Round(1.1733 * score) + 1.6807

If lpwmtbf_base = 0 Then

   MTBFBaseLPW.Text = 4999
```

```
Else

    MTBFBaseLPW.Text = 1 / Math.Round(1.1733 * score + 1.6807, 3) * 365


End If
```

```
target_rms_weight = Convert.ToString(Int32.Parse(Target_Sum_ClosedOld.Text) + Int32.Parse(Target_Sum_ClosedUNR.Text))

data_new_rank = "feeder=" + feeder_Id.ToString + ";"

data_new_rank += "LPW_SUM_Avg_last_04_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_04_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_04_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_45_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_45_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_45_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_24_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_24_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_24_hrs=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Avg_last_07_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Min_last_07_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_SUM_Max_last_07_days=" + Target_TotalLPWCurrent.Text + ";" + "LPW_RMS_Avg_last_04_hrs=" + target_rms_weight + ";" + "LPW_RMS_Min_last_04_hrs=" + target_rms_weight + ";" + "LPW_RMS_Max_last_04_hrs=" +
```

```
target_rms_weight + ";" + "LPW_RMS_Avg_last_45_days=" + target_rms_weight + ";" +
"LPW_RMS_Min_last_45_days=" + target_rms_weight + ";" + "LPW_RMS_Max_last_45_days=" +
target_rms_weight + ";" + "LPW_RMS_Avg_last_24_hrs=" + target_rms_weight + ";" +
"LPW_RMS_Min_last_24_hrs=" + target_rms_weight + ";" + "LPW_RMS_Max_last_24_hrs=" +
target_rms_weight + ";" + "LPW_RMS_Avg_last_07_days=" + target_rms_weight + ";" +
"LPW_RMS_Min_last_07_days=" + target_rms_weight + ";" + "LPW_RMS_Max_last_07_days=" +
target_rms_weight
```

```
Dim new_rank As String

'new_rank = myService.WSRerank(data_new_rank)

new_rank = WebserviceClient.WebserviceClient.proxy_rerank(data_new_rank.ToString)

Dim j As Integer

j = new_rank.IndexOf(",")

Dim score_new As Double

score_new = Double.Parse(new_rank.Substring(0, j - 1))



RankNewLPW.Text = new_rank.Substring(j + 1)
```

```
RankChangeLPW.Text = Convert.ToString(Int32.Parse(RankNewLPW.Text) - Int32.Parse(RankBaseLPW.Text))


'MTBFNewLPW.Text = (1.1733 * score_new) + 1.6807

'MTBFNewLPW.Text = -(112.33 * score_new) + 278.71

'MTBFNewLPW.Text = 1 / Math.Round(1.1638 * score_new + 1.6793, 3)


Dim lpwmtbf_new As Integer


lpwmtbf_new = Math.Round(1.1733 * score_new) + 1.6807


If lpwmtbf_new = 0 Then

  MTBFNewLPW.Text = 4999

Else

  MTBFNewLPW.Text = 1 / Math.Round(1.1733 * score_new + 1.6807, 3) * 365


End If
```

```
MTBFNewLPW.Text = Math.Round(Convert.ToDouble(MTBFNewLPW.Text))

MTBFChangeLPW.Text = Convert.ToString(Convert.ToDouble(MTBFNewLPW.Text) -
Convert.ToDouble(MTBFBaseLPW.Text))

MTBFBaseLPW.Text = Math.Round(Convert.ToDouble(MTBFBaseLPW.Text))

MTBFChangeLPW.Text = Math.Round(Convert.ToDouble(MTBFChangeLPW.Text))

FOTBaseLPW.Text =
Convert.ToString(Math.Round(Math.Max((Int32.Parse(RankBaseLPW.Text) * -0.0095 + 6), 0.0), 0))

FOTNewLPW.Text = Convert.ToString(Math.Max((Int32.Parse(RankNewLPW.Text) * -
0.0095 + 6), 0.0))

FOTChangeLPW.Text = Convert.ToString(Convert.ToDouble(FOTBaseLPW.Text) -
Convert.ToDouble(FOTNewLPW.Text))



FOTBaseLPW.Text = Math.Round(Convert.ToDouble(FOTBaseLPW.Text))

FOTChangeLPW.Text = Math.Round(Convert.ToDouble(FOTChangeLPW.Text))

FOTNewLPW.Text = Math.Round(Convert.ToDouble(FOTNewLPW.Text))

End Sub

Protected Sub GVovlnonsec_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GVovlnonsec.SelectedIndexChanged

'executed when a cell in gridview is selected
```

```
Dim struct As String

struct = GVovlnonsec.SelectedRow.Cells(1).Text


Session("struct") = struct.ToString

' TextBox1.Text = Session("struct")

Dim URL As String


URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"



Dim MyScript As New System.Text.StringBuilder

With MyScript

  .Append("<script type='text/javascript'>")

  '.Append("Response.Redirect(URL.ToString)")

  .Append("window.open('" & URL & "');")


  .Append("</script>")

End With
```

```
'Response.Redirect(URL)

'Server.Transfer(URL.ToString)

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text

End Sub

Protected Sub GV_OpenSwitches_update()

' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_openswitches As ArrayList

If Session("visited_openSwitches") Is Nothing Then
```

```
    visited_openswitches = New ArrayList
Else
    visited_openswitches = CType(Session("visited_openswitches"), ArrayList)
End If
If visited_openswitches.Count < 1 Then visited_openswitches.Add(0)

If GV_OpenSwitches.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_OpenSwitches.Rows
        Dim cb As CheckBox = row.FindControl("selectswitches")
        If cb IsNot Nothing And cb.Checked Then
            ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

            If visited_openswitches.Contains(id) Then
                Continue For
            Else
                visited_openswitches.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_openswitches") = visited_openswitches



      End If


  End Sub

  Protected Sub GV_fuse_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_fuse As ArrayList

    If Session("visited_fuse") Is Nothing Then

      visited_fuse = New ArrayList

    Else
```

```
    visited_fuse = CType(Session("visited_fuse"), ArrayList)
End If
If visited_fuse.Count < 1 Then visited_fuse.Add(0)

If GV_Fuse.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_Fuse.Rows
        Dim cb As CheckBox = row.FindControl("selectfuse")
        If cb IsNot Nothing And cb.Checked Then
            ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
            If visited_fuse.Contains(id) Then
                Continue For
            Else
                visited_fuse.Add(id)
            End If

            TextBox2.Text += id
```

```
            End If

        Next


        Session("visited_fuse") = visited_fuse




    End If


End Sub

Protected Sub GV_BanksOff_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_BanksOff As ArrayList

    If Session("visited_BanksOff") Is Nothing Then

        visited_BanksOff = New ArrayList

    Else

        visited_BanksOff = CType(Session("visited_BanksOff"), ArrayList)

    End If

    If visited_BanksOff.Count < 1 Then visited_BanksOff.Add(0)
```

```
If GV_BanksOff.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_BanksOff.Rows
    Dim cb As CheckBox = row.FindControl("selectBanksOff")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_BanksOff.Contains(id) Then
        Continue For
      Else
        visited_BanksOff.Add(id)
      End If


      TextBox2.Text += id
    End If
  Next
```

```
    Session("visited_BanksOff") = visited_BanksOff




  End If



End Sub



Protected Sub GV_ClosedOld_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_ClosedOld As ArrayList

  If Session("visited_ClosedOld") Is Nothing Then

    visited_ClosedOld = New ArrayList

  Else

    visited_ClosedOld = CType(Session("visited_ClosedOld"), ArrayList)

  End If

  If visited_ClosedOld.Count < 1 Then visited_ClosedOld.Add(0)



  If GV_ClosedOld.Rows.Count > 0 Then
```

```
For Each row As GridViewRow In GV_ClosedOld.Rows
  Dim cb As CheckBox = row.FindControl("selectClosedOld")
  If cb IsNot Nothing And cb.Checked Then
    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
    If visited_ClosedOld.Contains(id) Then
      Continue For
    Else
      visited_ClosedOld.Add(id)
    End If

    TextBox2.Text += id
  End If
Next

Session("visited_ClosedOld") = visited_ClosedOld
```

```
    End If


End Sub


Protected Sub GV_ClosedUNR_update()
    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
    Dim visited_ClosedUNR As ArrayList
    If Session("visited_ClosedUNR") Is Nothing Then
        visited_ClosedUNR = New ArrayList
    Else
        visited_ClosedUNR = CType(Session("visited_ClosedUNR"), ArrayList)
    End If
    If visited_ClosedUNR.Count < 1 Then visited_ClosedUNR.Add(0)


    If GV_ClosedUNR.Rows.Count > 0 Then


        For Each row As GridViewRow In GV_ClosedUNR.Rows
```

```
            Dim cb As CheckBox = row.FindControl("selectClosedUNR")
            If cb IsNot Nothing And cb.Checked Then
                ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                If visited_ClosedUNR.Contains(id) Then
                    Continue For
                Else
                    visited_ClosedUNR.Add(id)
                End If

                TextBox2.Text += id
            End If
        Next

        Session("visited_ClosedUNR") = visited_ClosedUNR


    End If
```

```
End Sub

Protected Sub GV_ClosedDNU_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_ClosedDNU As ArrayList

  If Session("visited_ClosedDNU") Is Nothing Then

    visited_ClosedDNU = New ArrayList

  Else

    visited_ClosedDNU = CType(Session("visited_ClosedDNU"), ArrayList)

  End If

  If visited_ClosedDNU.Count < 1 Then visited_ClosedDNU.Add(0)



  If GV_ClosedDNU.Rows.Count > 0 Then



    For Each row As GridViewRow In GV_ClosedDNU.Rows

      Dim cb As CheckBox = row.FindControl("selectClosedDNU")

      If cb IsNot Nothing And cb.Checked Then

        ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
```

```
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

            If visited_ClosedDNU.Contains(id) Then

               Continue For

            Else

               visited_ClosedDNU.Add(id)

            End If



            TextBox2.Text += id

         End If

      Next



      Session("visited_ClosedDNU") = visited_ClosedDNU



   End If

End Sub
```

```
Protected Sub GV_OpenMainsLTE2_update()
  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
  Dim visited_OpenMainsLTE2 As ArrayList
  If Session("visited_OpenMainsLTE2") Is Nothing Then
    visited_OpenMainsLTE2 = New ArrayList
  Else
    visited_OpenMainsLTE2 = CType(Session("visited_OpenMainsLTE2"), ArrayList)
  End If
  If visited_OpenMainsLTE2.Count < 1 Then visited_OpenMainsLTE2.Add(0)

  If GV_OpenMainsLTE2.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_OpenMainsLTE2.Rows
      Dim cb As CheckBox = row.FindControl("selectOpenMainsLTE2")
      If cb IsNot Nothing And cb.Checked Then
        ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
        Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
        If visited_OpenMainsLTE2.Contains(id) Then
```

```
                Continue For

            Else

                visited_OpenMainsLTE2.Add(id)

            End If



            TextBox2.Text += id

        End If

    Next



    Session("visited_OpenMainsLTE2") = visited_OpenMainsLTE2




End If



End Sub

Protected Sub GV_OpenMainsGT2_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_OpenMainsGT2 As ArrayList
```

```
If Session("visited_OpenMainsGT2") Is Nothing Then
   visited_OpenMainsGT2 = New ArrayList
Else
   visited_OpenMainsGT2 = CType(Session("visited_OpenMainsGT2"), ArrayList)
End If
If visited_OpenMainsGT2.Count < 1 Then visited_OpenMainsGT2.Add(0)

If GV_OpenMainsGT2.Rows.Count > 0 Then

   For Each row As GridViewRow In GV_OpenMainsGT2.Rows
      Dim cb As CheckBox = row.FindControl("selectOpenMainsGT2")
      If cb IsNot Nothing And cb.Checked Then
         ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
         Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
         If visited_OpenMainsGT2.Contains(id) Then
            Continue For
         Else
            visited_OpenMainsGT2.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_OpenMainsGT2") = visited_OpenMainsGT2


    End If


End Sub


Protected Sub GV_Missing_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_Missing As ArrayList

    If Session("visited_Missing") Is Nothing Then
```

```
    visited_Missing = New ArrayList
Else
    visited_Missing = CType(Session("visited_Missing"), ArrayList)
End If
If visited_Missing.Count < 1 Then visited_Missing.Add(0)

If GV_Missing.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_Missing.Rows
        Dim cb As CheckBox = row.FindControl("selectMissing")
        If cb IsNot Nothing And cb.Checked Then
            ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
            Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

            If visited_Missing.Contains(id) Then
                Continue For
            Else
                visited_Missing.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_Missing") = visited_Missing


      End If

End Sub


Protected Sub GV_OpenSwitches_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_openswitches_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_openSwitches_NbyFactorGTE10") Is Nothing Then
  visited_openswitches_NbyFactorGTE10 = New ArrayList
Else
  visited_openswitches_NbyFactorGTE10 = CType(Session("visited_openswitches_NbyFactorGTE10"), ArrayList)
End If
If visited_openswitches_NbyFactorGTE10.Count < 1 Then visited_openswitches_NbyFactorGTE10.Add(0)

If GV_OpenSwitches_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_OpenSwitches_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectswitches_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_openswitches_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
                Else

                    visited_openswitches_NbyFactorGTE10.Add(id)

                End If



                TextBox2.Text += id

            End If

        Next



        Session("visited_openswitches_NbyFactorGTE10") =
visited_openswitches_NbyFactorGTE10



    End If



End Sub

Protected Sub GV_fuse_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_fuse_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_fuse_NbyFactorGTE10") Is Nothing Then
  visited_fuse_NbyFactorGTE10 = New ArrayList
Else
  visited_fuse_NbyFactorGTE10 = CType(Session("visited_fuse_NbyFactorGTE10"),
ArrayList)
End If
If visited_fuse_NbyFactorGTE10.Count < 1 Then visited_fuse_NbyFactorGTE10.Add(0)

If GV_Fuse_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_Fuse_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectfuse_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_fuse_NbyFactorGTE10.Contains(id) Then
        Continue For
      Else
```

```
                visited_fuse_NbyFactorGTE10.Add(id)

            End If


            TextBox2.Text += id

        End If

    Next


    Session("visited_fuse_NbyFactorGTE10") = visited_fuse_NbyFactorGTE10




End If


End Sub

Protected Sub GV_BanksOff_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_BanksOff_NbyFactorGTE10 As ArrayList

    If Session("visited_BanksOff_NbyFactorGTE10") Is Nothing Then

        visited_BanksOff_NbyFactorGTE10 = New ArrayList
```

```
Else

  visited_BanksOff_NbyFactorGTE10 =
CType(Session("visited_BanksOff_NbyFactorGTE10"), ArrayList)

End If

If visited_BanksOff_NbyFactorGTE10.Count < 1 Then
visited_BanksOff_NbyFactorGTE10.Add(0)


If GV_BanksOff_NbyFactorGTE10.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_BanksOff_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectBanksOff_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_BanksOff_NbyFactorGTE10.Contains(id) Then
        Continue For
      Else
        visited_BanksOff_NbyFactorGTE10.Add(id)
```

```
            End If


            TextBox2.Text += id

          End If

        Next


        Session("visited_BanksOff_NbyFactorGTE10") = visited_BanksOff_NbyFactorGTE10



      End If


End Sub


Protected Sub GV_ClosedOld_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_ClosedOld_NbyFactorGTE10 As ArrayList

  If Session("visited_ClosedOld_NbyFactorGTE10") Is Nothing Then

    visited_ClosedOld_NbyFactorGTE10 = New ArrayList
```

```
Else

  visited_ClosedOld_NbyFactorGTE10 =
CType(Session("visited_ClosedOld_NbyFactorGTE10"), ArrayList)

End If

If visited_ClosedOld_NbyFactorGTE10.Count < 1 Then
visited_ClosedOld_NbyFactorGTE10.Add(0)


If GV_ClosedOld_NbyFactorGTE10.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_ClosedOld_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectClosedOld_GTE10)")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_ClosedOld_NbyFactorGTE10.Contains(id) Then
        Continue For
      Else
        visited_ClosedOld_NbyFactorGTE10.Add(id)
```

```
                End If


                TextBox2.Text += id

            End If

        Next


                Session("visited_ClosedOld_NbyFactorGTE10") =
visited_ClosedOld_NbyFactorGTE10


    End If


End Sub


Protected Sub GV_ClosedUNR_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_ClosedUNR_NbyFactorGTE10 As ArrayList

    If Session("visited_ClosedUNR_NbyFactorGTE10") Is Nothing Then
```

```
visited_ClosedUNR_NbyFactorGTE10 = New ArrayList
Else
visited_ClosedUNR_NbyFactorGTE10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE10"), ArrayList)
End If
If visited_ClosedUNR_NbyFactorGTE10.Count < 1 Then
visited_ClosedUNR_NbyFactorGTE10.Add(0)

If GV_ClosedUNR_NbyFactorGTE10.Rows.Count > 0 Then

For Each row As GridViewRow In GV_ClosedUNR_NbyFactorGTE10.Rows
Dim cb As CheckBox = row.FindControl("selectClosedUNR_GTE10")
If cb IsNot Nothing And cb.Checked Then
' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
If visited_ClosedUNR_NbyFactorGTE10.Contains(id) Then
Continue For
Else
```

```
                visited_ClosedUNR_NbyFactorGTE10.Add(id)

            End If


            TextBox2.Text += id

        End If

    Next


    Session("visited_ClosedUNR_NbyFactorGTE10") =
visited_ClosedUNR_NbyFactorGTE10



End If


End Sub

Protected Sub GV_ClosedDNU_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_ClosedDNU_NbyFactorGTE10 As ArrayList

  If Session("visited_ClosedDNU_NbyFactorGTE10") Is Nothing Then
```

```
visited_ClosedDNU_NbyFactorGTE10 = New ArrayList

Else

visited_ClosedDNU_NbyFactorGTE10 = CType(Session("visited_ClosedDNU_NbyFactorGTE10"), ArrayList)

End If

If visited_ClosedDNU_NbyFactorGTE10.Count < 1 Then visited_ClosedDNU_NbyFactorGTE10.Add(0)

If GV_ClosedDNU_NbyFactorGTE10.Rows.Count > 0 Then

For Each row As GridViewRow In GV_ClosedDNU_NbyFactorGTE10.Rows
Dim cb As CheckBox = row.FindControl("selectClosedDNU_GTE10")
If cb IsNot Nothing And cb.Checked Then
' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
If visited_ClosedDNU_NbyFactorGTE10.Contains(id) Then
Continue For
Else
```

```
                visited_ClosedDNU_NbyFactorGTE10.Add(id)

            End If


            TextBox2.Text += id

        End If

    Next


    Session("visited_ClosedDNU_NbyFactorGTE10") =
visited_ClosedDNU_NbyFactorGTE10


End If


End Sub


Protected Sub GV_missing_NbyFactorGTE10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_missing_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_missing_NbyFactorGTE10") Is Nothing Then
  visited_missing_NbyFactorGTE10 = New ArrayList
Else
  visited_missing_NbyFactorGTE10 = CType(Session("visited_missing_NbyFactorGTE10"), ArrayList)
End If
If visited_missing_NbyFactorGTE10.Count < 1 Then visited_missing_NbyFactorGTE10.Add(0)

If GV_Missing_NbyFActorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_Missing_NbyFActorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectMissing_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_missing_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
            Else

               visited_missing_NbyFactorGTE10.Add(id)

            End If



            TextBox2.Text += id

         End If

      Next



      Session("visited_Missing_NbyFactorGTE10") = visited_missing_NbyFactorGTE10



   End If



End Sub



Protected Sub GV_OpenMainsLTE2_NbyFactorGTE10_update()

   ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

   Dim visited_OpenMainsLTE2_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_OpenMainsLTE2_NbyFactorGTE10") Is Nothing Then
  visited_OpenMainsLTE2_NbyFactorGTE10 = New ArrayList
Else
  visited_OpenMainsLTE2_NbyFactorGTE10 = CType(Session("visited_OpenMainsLTE2_NbyFactorGTE10"), ArrayList)
End If
If visited_OpenMainsLTE2_NbyFactorGTE10.Count < 1 Then visited_OpenMainsLTE2_NbyFactorGTE10.Add(0)

If GV_OpenMainsLTE2_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_OpenMainsLTE2_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectOpenMainsLTE2_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_OpenMainsLTE2_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
                Else

                    visited_OpenMainsLTE2_NbyFactorGTE10.Add(id)

                End If



                TextBox2.Text += id

            End If

        Next



        Session("visited_OpenMainsLTE2_NbyFactorGTE10") =
visited_OpenMainsLTE2_NbyFactorGTE10



    End If



End Sub

Protected Sub GV_OpenMainsGT2_NbyFactorGTE10_update()

    ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_OpenMainsGT2_NbyFactorGTE10 As ArrayList
```

```
If Session("visited_OpenMainsGT2_NbyFactorGTE10") Is Nothing Then
  visited_OpenMainsGT2_NbyFactorGTE10 = New ArrayList
Else
  visited_OpenMainsGT2_NbyFactorGTE10 = CType(Session("visited_OpenMainsGT2_NbyFactorGTE10"), ArrayList)
End If
If visited_OpenMainsGT2_NbyFactorGTE10.Count < 1 Then visited_OpenMainsGT2_NbyFactorGTE10.Add(0)

If GV_OpenMainsGT2_NbyFactorGTE10.Rows.Count > 0 Then

  For Each row As GridViewRow In GV_OpenMainsGT2_NbyFactorGTE10.Rows
    Dim cb As CheckBox = row.FindControl("selectOpenMainsGT2_GTE10")
    If cb IsNot Nothing And cb.Checked Then
      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
      If visited_OpenMainsGT2_NbyFactorGTE10.Contains(id) Then
        Continue For
```

```
                Else

                    visited_OpenMainsGT2_NbyFactorGTE10.Add(id)

                End If



                TextBox2.Text += id

            End If

        Next



        Session("visited_OpenMainsGT2_NbyFactorGTE10") =
visited_OpenMainsGT2_NbyFactorGTE10



    End If



End Sub
```

```
Protected Sub GV_OpenSwitches_NbyFactorGTE5LT10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_openswitches_NbyFactorGTE5LT10 As ArrayList

  If Session("visited_openSwitches_NbyFactorGTE5LT10") Is Nothing Then

    visited_openswitches_NbyFactorGTE5LT10 = New ArrayList

  Else

    visited_openswitches_NbyFactorGTE5LT10 =
CType(Session("visited_openswitches_NbyFactorGTE5LT10"), ArrayList)

  End If

  If visited_openswitches_NbyFactorGTE5LT10.Count < 1 Then
visited_openswitches_NbyFactorGTE5LT10.Add(0)


  If GV_OpenSwitches_NbyFactorGTE5LT10.Rows.Count > 0 Then


    For Each row As GridViewRow In GV_OpenSwitches_NbyFactorGTE5LT10.Rows

      Dim cb As CheckBox = row.FindControl("selectswitches_GTE5LT10")
```

```
                If cb IsNot Nothing And cb.Checked Then

                    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

                    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

                    If visited_openswitches_NbyFactorGTE5LT10.Contains(id) Then

                        Continue For

                    Else

                        visited_openswitches_NbyFactorGTE5LT10.Add(id)

                    End If



                    TextBox2.Text += id

                End If

            Next



            Session("visited_openswitches_NbyFactorGTE5LT10") =
visited_openswitches_NbyFactorGTE5LT10



        End If
```

```
End Sub

Protected Sub GV_fuse_NbyFactorGTE5LT10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_fuse_NbyFactorGTE5LT10 As ArrayList

  If Session("visited_fuse_NbyFactorGTE5LT10") Is Nothing Then

    visited_fuse_NbyFactorGTE5LT10 = New ArrayList

  Else

    visited_fuse_NbyFactorGTE5LT10 =
CType(Session("visited_fuse_NbyFactorGTE5LT10"), ArrayList)

  End If

  If visited_fuse_NbyFactorGTE5LT10.Count < 1 Then
visited_fuse_NbyFactorGTE5LT10.Add(0)


  If GV_Fuse_NbyFactorGTE5LT10.Rows.Count > 0 Then


    For Each row As GridViewRow In GV_Fuse_NbyFactorGTE5LT10.Rows

      Dim cb As CheckBox = row.FindControl("selectfuse_GTE5LT10")
```

```
            If cb IsNot Nothing And cb.Checked Then

                ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

                Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

                If visited_fuse_NbyFactorGTE5LT10.Contains(id) Then

                    Continue For

                Else

                    visited_fuse_NbyFactorGTE5LT10.Add(id)

                End If



                TextBox2.Text += id

            End If

        Next



        Session("visited_fuse_NbyFactorGTE5LT10") = visited_fuse_NbyFactorGTE5LT10



    End If
```

```
End Sub

Protected Sub GV_BanksOff_NbyFactorGTE5LT10_update()

  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

  Dim visited_BanksOff_NbyFactorGTE5LT10 As ArrayList

  If Session("visited_BanksOff_NbyFactorGTE5LT10") Is Nothing Then

    visited_BanksOff_NbyFactorGTE5LT10 = New ArrayList

  Else

    visited_BanksOff_NbyFactorGTE5LT10 =
CType(Session("visited_BanksOff_NbyFactorGTE5LT10"), ArrayList)

  End If

  If visited_BanksOff_NbyFactorGTE5LT10.Count < 1 Then
visited_BanksOff_NbyFactorGTE5LT10.Add(0)


  If GV_BanksOff_NbyFactorGTE5LT10.Rows.Count > 0 Then


    For Each row As GridViewRow In GV_BanksOff_NbyFactorGTE5LT10.Rows

      Dim cb As CheckBox = row.FindControl("selectBanksOff_GTE5LT10")

      If cb IsNot Nothing And cb.Checked Then
```

```
                ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

                Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

                If visited_BanksOff_NbyFactorGTE5LT10.Contains(id) Then

                  Continue For

                Else

                  visited_BanksOff_NbyFactorGTE5LT10.Add(id)

                End If



                TextBox2.Text += id

              End If

            Next



            Session("visited_BanksOff_NbyFactorGTE5LT10")                    =
visited_BanksOff_NbyFactorGTE5LT10



          End If
```

```
End Sub

Protected Sub GV_ClosedOld_NbyFactorGTE5LT10_update()
  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
  Dim visited_ClosedOld_NbyFactorGTE5LT10 As ArrayList
  If Session("visited_ClosedOld_NbyFactorGTE5LT10") Is Nothing Then
    visited_ClosedOld_NbyFactorGTE5LT10 = New ArrayList
  Else
    visited_ClosedOld_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedOld_NbyFactorGTE5LT10"), ArrayList)
  End If
  If visited_ClosedOld_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedOld_NbyFactorGTE5LT10.Add(0)

  If GV_ClosedOld_NbyFactorGTE5LT10.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_ClosedOld_NbyFactorGTE5LT10.Rows
      Dim cb As CheckBox = row.FindControl("selectClosedOld_GTE5LT10")
```

```
                If cb IsNot Nothing And cb.Checked Then

                    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

                    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

                    If visited_ClosedOld_NbyFactorGTE5LT10.Contains(id) Then

                        Continue For

                    Else

                        visited_ClosedOld_NbyFactorGTE5LT10.Add(id)

                    End If



                    TextBox2.Text += id

                End If

            Next



            Session("visited_ClosedOld_NbyFactorGTE5LT10") =
visited_ClosedOld_NbyFactorGTE5LT10



        End If
```

```
End Sub

Protected Sub GV_ClosedUNR_NbyFactorGTE5LT10_update()
  ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
  Dim visited_ClosedUNR_NbyFactorGTE5LT10 As ArrayList
  If Session("visited_ClosedUNR_NbyFactorGTE5LT10") Is Nothing Then
    visited_ClosedUNR_NbyFactorGTE5LT10 = New ArrayList
  Else
    visited_ClosedUNR_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE5LT10"), ArrayList)
  End If
  If visited_ClosedUNR_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedUNR_NbyFactorGTE5LT10.Add(0)

  If GV_ClosedUNR_NbyFactorGTE5LT10.Rows.Count > 0 Then

    For Each row As GridViewRow In GV_ClosedUNR_NbyFactorGTE5LT10.Rows
```

```
                Dim cb As CheckBox = row.FindControl("selectClosedUNR_GTE5LT10")
                If cb IsNot Nothing And cb.Checked Then
                  ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                  Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                  If visited_ClosedUNR_NbyFactorGTE5LT10.Contains(id) Then
                    Continue For
                  Else
                    visited_ClosedUNR_NbyFactorGTE5LT10.Add(id)
                  End If

                  TextBox2.Text += id
                End If
              Next

              Session("visited_ClosedUNR_NbyFactorGTE5LT10") =
visited_ClosedUNR_NbyFactorGTE5LT10
```

```
End If


End Sub

Protected Sub GV_ClosedDNU_NbyFactorGTE5LT10_update()

' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_ClosedDNU_NbyFactorGTE5LT10 As ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE5LT10") Is Nothing Then

visited_ClosedDNU_NbyFactorGTE5LT10 = New ArrayList

Else

visited_ClosedDNU_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE5LT10"), ArrayList)

End If

If visited_ClosedDNU_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedDNU_NbyFactorGTE5LT10.Add(0)


If GV_ClosedDNU_NbyFactorGTE5LT10.Rows.Count > 0 Then


For Each row As GridViewRow In GV_ClosedDNU_NbyFactorGTE5LT10.Rows
```

```
                Dim cb As CheckBox = row.FindControl("selectClosedDNU_GTE5LT10")
                If cb IsNot Nothing And cb.Checked Then
                  ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
                  Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
                  If visited_ClosedDNU_NbyFactorGTE5LT10.Contains(id) Then
                    Continue For
                  Else
                    visited_ClosedDNU_NbyFactorGTE5LT10.Add(id)
                  End If

                  TextBox2.Text += id
                End If
              Next

              Session("visited_ClosedDNU_NbyFactorGTE5LT10") =
visited_ClosedDNU_NbyFactorGTE5LT10
```

```
        End If


    End Sub



    Protected Sub GV_missing_NbyFactorGTE5LT10_update()
        ' GV_missing.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
        Dim visited_missing_NbyFactorGTE5LT10 As ArrayList
        If Session("visited_missing_NbyFactorGTE5LT10") Is Nothing Then
            visited_missing_NbyFactorGTE5LT10 = New ArrayList
        Else
            visited_missing_NbyFactorGTE5LT10 =
CType(Session("visited_missing_NbyFactorGTE5LT10"), ArrayList)
        End If
        If visited_missing_NbyFactorGTE5LT10.Count < 1 Then
visited_missing_NbyFactorGTE5LT10.Add(0)


        If GV_Missing_NbyFactorGTE5LT10.Rows.Count > 0 Then
```

```
For Each row As GridViewRow In GV_Missing_NbyFactorGTE5LT10.Rows
  Dim cb As CheckBox = row.FindControl("selectMissing_GTE5LT10")
  If cb IsNot Nothing And cb.Checked Then
    ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text
    Dim id As String = row.Cells(0).Text + row.RowIndex.ToString
    If visited_missing_NbyFactorGTE5LT10.Contains(id) Then
      Continue For
    Else
      visited_missing_NbyFactorGTE5LT10.Add(id)
    End If

    TextBox2.Text += id
  End If
Next

Session("visited_missing_NbyFactorGTE5LT10") =
visited_missing_NbyFactorGTE5LT10
```

```
        End If


    End Sub


    Protected Sub GV_OpenMainsLTE2_NbyFactorGTE5LT10_update()
        ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
        Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As ArrayList
        If Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") Is Nothing Then
            visited_OpenMainsLTE2_NbyFactorGTE5LT10 = New ArrayList
        Else
            visited_OpenMainsLTE2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10"), ArrayList)
        End If
        If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Count < 1 Then
visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(0)
```

```
If GV_OpenMainsLTE2_NbyFactorGTE5LT10.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_OpenMainsLTE2_NbyFactorGTE5LT10.Rows

    Dim cb As CheckBox = row.FindControl("selectOpenMainsLTE2_GTE5LT10")

    If cb IsNot Nothing And cb.Checked Then

      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

      If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Contains(id) Then

        Continue For

      Else

        visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(id)

      End If



      TextBox2.Text += id

    End If

  Next
```

```
                Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") =
visited_OpenMainsLTE2_NbyFactorGTE5LT10

            End If

        End Sub

        Protected Sub GV_OpenMainsGT2_NbyFactorGTE5LT10_update()

            ' GV_OpenSwitches.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

            Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As ArrayList

            If Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") Is Nothing Then

                visited_OpenMainsGT2_NbyFactorGTE5LT10 = New ArrayList

            Else

                visited_OpenMainsGT2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE5LT10"), ArrayList)

            End If

            If visited_OpenMainsGT2_NbyFactorGTE5LT10.Count < 1 Then
visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(0)
```

```
If GV_OpenMainsGT2_NbyFactorGTE5LT10.Rows.Count > 0 Then


  For Each row As GridViewRow In GV_OpenMainsGT2_NbyFactorGTE5LT10.Rows

    Dim cb As CheckBox = row.FindControl("selectOpenMainsGT2_GTE5LT10")

    If cb IsNot Nothing And cb.Checked Then

      ' Dim id As String = GV_OpenSwitches.SelectedRow.Cells(0).Text

      Dim id As String = row.Cells(0).Text + row.RowIndex.ToString

      If visited_OpenMainsGT2_NbyFactorGTE5LT10.Contains(id) Then

        Continue For

      Else

        visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(id)

      End If



      TextBox2.Text += id

    End If

  Next
```

Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") = visited_OpenMainsGT2_NbyFactorGTE5LT10

End If

End Sub

Protected Sub GV_BanksOff_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_BanksOff.SelectedIndexChanged

GV_BanksOff.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_BanksOff As ArrayList

If Session("visited_BanksOff") Is Nothing Then

visited_BanksOff = New ArrayList

Else

visited_BanksOff = CType(Session("visited_BanksOff"), ArrayList)

End If

Dim id As String = GV_BanksOff.SelectedRow.Cells(0).Text

```
If visited_BanksOff.Count < 1 Then visited_BanksOff.Add(0)

If visited_BanksOff.Contains(id) Then

  Exit Sub

Else

  visited_BanksOff.Add(id)

End If



Session("visited_BanksOff") = visited_BanksOff









End Sub



Protected Sub GV_Fuse_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GV_Fuse.SelectedIndexChanged

  GV_Fuse.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
```

```
Dim visited_Fuse As ArrayList
If Session("visited_Fuse") Is Nothing Then
   visited_Fuse = New ArrayList
Else
   visited_Fuse = CType(Session("visited_Fuse"), ArrayList)
End If
Dim id As String = GV_Fuse.SelectedRow.Cells(0).Text
If visited_Fuse.Count < 1 Then visited_Fuse.Add(0)
If visited_Fuse.Contains(id) Then
   Exit Sub
Else
   visited_Fuse.Add(id)
End If

Session("visited_Fuse") = visited_Fuse
```

```
End Sub

Protected Sub GV_ClosedOld_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles GV_ClosedOld.SelectedIndexChanged
    GV_ClosedOld.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
    Dim visited_ClosedOld As ArrayList
    If Session("visited_ClosedOld") Is Nothing Then
      visited_ClosedOld = New ArrayList
    Else
      visited_ClosedOld = CType(Session("visited_ClosedOld"), ArrayList)
    End If
    Dim id As String = GV_ClosedOld.SelectedRow.Cells(0).Text
    If visited_ClosedOld.Count < 1 Then visited_ClosedOld.Add(0)
    If visited_ClosedOld.Contains(id) Then
      Exit Sub
    Else
```

```
            visited_ClosedOld.Add(id)
        End If

        Session("visited_fuse") = visited_ClosedOld

    End Sub

    Protected Sub GV_ClosedUNR_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles GV_ClosedUNR.SelectedIndexChanged
        GV_ClosedUNR.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
        Dim visited_ClosedUNR As ArrayList
        If Session("visited_ClosedUNR") Is Nothing Then
            visited_ClosedUNR = New ArrayList
        Else
            visited_ClosedUNR = CType(Session("visited_ClosedUNR"), ArrayList)
```

```
        End If

        Dim id As String = GV_ClosedUNR.SelectedRow.Cells(0).Text

        If visited_ClosedUNR.Count < 1 Then visited_ClosedUNR.Add(0)

        If visited_ClosedUNR.Contains(id) Then

          Exit Sub

        Else

          visited_ClosedUNR.Add(id)

        End If



        Session("visited_ClosedUNR") = visited_ClosedUNR



    End Sub



        Protected Sub GV_ClosedDNU_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles GV_ClosedDNU.SelectedIndexChanged

        GV_ClosedDNU.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

        Dim visited_ClosedDNU As ArrayList
```

```
If Session("visited_ClosedDNU") Is Nothing Then
    visited_ClosedDNU = New ArrayList
Else
    visited_ClosedDNU = CType(Session("visited_ClosedDNU"), ArrayList)
End If
Dim id As String = GV_ClosedDNU.SelectedRow.Cells(0).Text
If visited_ClosedDNU.Count < 1 Then visited_ClosedDNU.Add(0)
If visited_ClosedDNU.Contains(id) Then
    Exit Sub
Else
    visited_ClosedDNU.Add(id)
End If

Session("visited_ClosedDNU") = visited_ClosedDNU

End Sub
```

```
Protected Sub GV_OpenMainsLTE2_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles GV_OpenMainsLTE2.SelectedIndexChanged

    GV_OpenMainsLTE2.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_OpenMainsLTE2 As ArrayList

    If Session("visited_OpenMainsLTE2") Is Nothing Then

      visited_OpenMainsLTE2 = New ArrayList

    Else

      visited_OpenMainsLTE2 = CType(Session("visited_OpenMainsLTE2"), ArrayList)

    End If

    Dim id As String = GV_OpenMainsLTE2.SelectedRow.Cells(0).Text

    If visited_OpenMainsLTE2.Count < 1 Then visited_OpenMainsLTE2.Add(0)

    If visited_OpenMainsLTE2.Contains(id) Then

      Exit Sub

    Else

      visited_OpenMainsLTE2.Add(id)

    End If



    Session("visited_OpenMainsLTE2") = visited_OpenMainsLTE2
```

```
End Sub

Protected Sub GV_OpenMainsGT2_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles GV_OpenMainsGT2.SelectedIndexChanged

    GV_OpenMainsGT2.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_OpenMainsGT2 As ArrayList

    If Session("visited_OpenMainsGT2") Is Nothing Then

      visited_OpenMainsGT2 = New ArrayList

    Else

      visited_OpenMainsGT2 = CType(Session("visited_OpenMainsGT2"), ArrayList)

    End If

    Dim id As String = GV_OpenMainsGT2.SelectedRow.Cells(0).Text

    If visited_OpenMainsGT2.Count < 1 Then visited_OpenMainsGT2.Add(0)

    If visited_OpenMainsGT2.Contains(id) Then

      Exit Sub

    Else

      visited_OpenMainsGT2.Add(id)

    End If
```

```
    Session("visited_OpenMainsGT2") = visited_OpenMainsGT2



End Sub



Private Sub reset_visitedLPWSessionVariables()

    'reset session variables for the case when transformer is on feeder

    Dim visited_OpenSwitches As New ArrayList

    visited_OpenSwitches.Add(0)

    Session("visited_OpenSwitches") = visited_OpenSwitches



    Dim visited_BanksOff As New ArrayList

    visited_BanksOff.Add(0)

    Session("visited_BanksOff") = visited_BanksOff



    Dim visited_Fuse As New ArrayList

    visited_Fuse.Add(0)
```

```
Session("visited_Fuse") = visited_Fuse



Dim visited_ClosedOld As New ArrayList

visited_ClosedOld.Add(0)

Session("visited_ClosedOld") = visited_ClosedOld



Dim visited_ClosedUNR As New ArrayList

visited_ClosedUNR.Add(0)

Session("visited_ClosedUNR") = visited_ClosedUNR



Dim visited_ClosedDNU As New ArrayList

visited_ClosedDNU.Add(0)

Session("visited_ClosedDNU") = visited_ClosedDNU



Dim visited_OpenMainsLTE2 As New ArrayList

visited_OpenMainsLTE2.Add(0)

Session("visited_OpenMainsLTE2") = visited_OpenMainsLTE2
```

```
Dim visited_OpenMainsGT2 As New ArrayList

visited_OpenMainsGT2.Add(0)

Session("visited_OpenMainsGT2") = visited_OpenMainsGT2



Dim visited_Missing As New ArrayList

visited_Missing.Add(0)

Session("visited_Missing") = visited_Missing



'reset session variables for the case - transformer with nearby factor GTE10



Dim visited_OpenSwitches_NbyFactorGTE10 As New ArrayList

visited_OpenSwitches_NbyFactorGTE10.Add(0)

Session("visited_OpenSwitches_NbyFactorGTE10") =
visited_OpenSwitches_NbyFactorGTE10



Dim visited_BanksOff_NbyFactorGTE10 As New ArrayList

visited_BanksOff_NbyFactorGTE10.Add(0)
```

Session("visited_BanksOff_NbyFactorGTE10") = visited_BanksOff_NbyFactorGTE10

Dim visited_Fuse_NbyFactorGTE10 As New ArrayList

visited_Fuse_NbyFactorGTE10.Add(0)

Session("visited_Fuse_NbyFactorGTE10") = visited_Fuse_NbyFactorGTE10

Dim visited_ClosedOld_NbyFactorGTE10 As New ArrayList

visited_ClosedOld_NbyFactorGTE10.Add(0)

Session("visited_ClosedOld_NbyFactorGTE10") = visited_ClosedOld_NbyFactorGTE10

Dim visited_ClosedUNR_NbyFactorGTE10 As New ArrayList

visited_ClosedUNR_NbyFactorGTE10.Add(0)

Session("visited_ClosedUNR_NbyFactorGTE10") = visited_ClosedUNR_NbyFactorGTE10

Dim visited_ClosedDNU_NbyFactorGTE10 As New ArrayList

visited_ClosedDNU_NbyFactorGTE10.Add(0)

Session("visited_ClosedDNU_NbyFactorGTE10") = visited_ClosedDNU_NbyFactorGTE10

Dim visited_Missing_NbyFactorGTE10 As New ArrayList

visited_Missing_NbyFactorGTE10.Add(0)

Session("visited_Missing_NbyFactorGTE10") = visited_Missing_NbyFactorGTE10

Dim visited_OpenMainsLTE2_NbyFactorGTE10 As New ArrayList

visited_OpenMainsLTE2_NbyFactorGTE10.Add(0)

Session("visited_OpenMainsLTE2_NbyFactorGTE10") = visited_OpenMainsLTE2_NbyFactorGTE10

Dim visited_OpenMainsGT2_NbyFactorGTE10 As New ArrayList

visited_OpenMainsGT2_NbyFactorGTE10.Add(0)

Session("visited_OpenMainsGT2_NbyFactorGTE10") = visited_OpenMainsGT2_NbyFactorGTE10

'reset session variables for the case - transformer with nearby factor GTE5lt10

```
Dim visited_OpenSwitches_NbyFactorGTE5LT10 As New ArrayList

visited_OpenSwitches_NbyFactorGTE5LT10.Add(0)

Session("visited_OpenSwitches_NbyFactorGTE5LT10") =
visited_OpenSwitches_NbyFactorGTE5LT10


Dim visited_BanksOff_NbyFactorGTE5LT10 As New ArrayList

visited_BanksOff_NbyFactorGTE5LT10.Add(0)

Session("visited_BanksOff_NbyFactorGTE5LT10") =
visited_BanksOff_NbyFactorGTE5LT10


Dim visited_Fuse_NbyFactorGTE5LT10 As New ArrayList

visited_Fuse_NbyFactorGTE5LT10.Add(0)

Session("visited_Fuse_NbyFactorGTE5LT10") = visited_Fuse_NbyFactorGTE5LT10


Dim visited_ClosedOld_NbyFactorGTE5LT10 As New ArrayList

visited_ClosedOld_NbyFactorGTE5LT10.Add(0)

Session("visited_ClosedOld_NbyFactorGTE5LT10") =
visited_ClosedOld_NbyFactorGTE5LT10
```

Dim visited_ClosedUNR_NbyFactorGTE5LT10 As New ArrayList

visited_ClosedUNR_NbyFactorGTE5LT10.Add(0)

Session("visited_ClosedUNR_NbyFactorGTE5LT10") = visited_ClosedUNR_NbyFactorGTE5LT10

Dim visited_ClosedDNU_NbyFactorGTE5LT10 As New ArrayList

visited_ClosedDNU_NbyFactorGTE5LT10.Add(0)

Session("visited_ClosedDNU_NbyFactorGTE5LT10") = visited_ClosedDNU_NbyFactorGTE5LT10

Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As New ArrayList

visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(0)

Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") = visited_OpenMainsLTE2_NbyFactorGTE5LT10

Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As New ArrayList

visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(0)

Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") = visited_OpenMainsGT2_NbyFactorGTE5LT10

```
Dim visited_Missing_NbyFactorGTE5LT10 As New ArrayList

visited_Missing_NbyFactorGTE5LT10.Add(0)

Session("visited_Missing_NbyFactorGTE5LT10") =
visited_Missing_NbyFactorGTE5LT10

End Sub

Protected Sub UpdateTargetVaultCounts_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UpdateTargetVaultCounts.Click

update_Target_OpenSwitches()

update_Target_Fuse()

update_Target_BanksOff()

update_Target_ClosedOld()
```

update_Target_ClosedUNR()

update_Target_ClosedDNU()

update_Target_OpenMainsLTE2()

update_Target_OpenMainsGT2()

' Target_Total_LPW_Calc()

update_Target_Missing()

update_Target_OpenSwitches_NbyFactorGTE10()

update_Target_Fuse_NbyFactorGTE10()

update_Target_BanksOff_NbyFactorGTE10()

update_Target_ClosedOld_NbyFactorGTE10()

update_Target_ClosedUNR_NbyFactorGTE10()

update_Target_ClosedDNU_NbyFactorGTE10()

update_Target_OpenMainsLTE2_NbyFactorGTE10()

update_Target_OpenMainsGT2_NbyFactorGTE10()

update_Target_MIssing_NbyFactorGTE10()

update_Target_OpenSwitches_NbyFactorGTE5LT10()

update_Target_Fuse_NbyFactorGTE5LT10()

update_Target_BanksOff_NbyFactorGTE5LT10()

update_Target_ClosedOld_NbyFactorGTE5LT10()

update_Target_ClosedUNR_NbyFactorGTE5LT10()

update_Target_ClosedDNU_NbyFactorGTE5LT10()

update_Target_OpenMainsLTE2_NbyFactorGTE5LT10()

update_Target_OpenMainsGT2_NbyFactorGTE5LT10()

update_Target_Missing_NbyFactorGTE5LT10()

Target_Total_LPW_Calc()

LPW_Cost_Calculation()

Dim MyScript As New System.Text.StringBuilder

With MyScript

```
.Append("<script type='text/javascript'>")

.Append("alert ('update completed');")

.Append("</script>")

End With



'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)

ScriptManager.RegisterStartupScript(Me, Me.GetType(), "updatesdone",
MyScript.ToString, False)



End Sub

Private Sub reset_lpwtotals()

TotalLPWCurrent.Text = ""
```

```
End Sub

''' <summary>

''' below are the methods to update target vault counts

''' based on checked boxes

''' </summary>

''' <remarks></remarks>

Private Sub update_Target_OpenSwitches()

  GV_OpenSwitches_update()

  Dim visited_OpenSwitches As New ArrayList

  If Session("visited_OpenSwitches") Is Nothing Then Exit Sub



  visited_OpenSwitches = CType(Session("visited_OpenSwitches"), ArrayList)



  Target_OpenSwitchesFeeder.Text = Int32.Parse(OpenSwitchesFeederCurrent.Text) -
(visited_OpenSwitches.Count - 1)

  'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

  TextBox2.Text = visited_OpenSwitches.Count

  'TextBox2.Text = vault_count
```

```
End Sub

Private Sub update_Target_Fuse()
  GV_fuse_update()
  Dim visited_fuse As New ArrayList
  If Session("visited_fuse") Is Nothing Then Exit Sub

  visited_fuse = CType(Session("visited_fuse"), ArrayList)

  Target_FuseFeederCurrent.Text = Int32.Parse(FuseFeederCurrent.Text) -
(visited_fuse.Count - 1)

  Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 *
Int32.Parse(Target_Fuse_GTE10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)
End Sub
```

```
Private Sub update_Target_BanksOff()

  GV_BanksOff_update()

  Dim visited_BanksOff As New ArrayList

  If Session("visited_BanksOff") Is Nothing Then Exit Sub



  visited_BanksOff = CType(Session("visited_BanksOff"), ArrayList)



  Target_BanksOffFeederCurrent.Text = Int32.Parse(BanksOffFeederCurrent.Text) -
(visited_BanksOff.Count - 1)

End Sub

Private Sub update_Target_ClosedOld()

  GV_ClosedOld_update()

  Dim visited_ClosedOld As New ArrayList

  If Session("visited_ClosedOld") Is Nothing Then Exit Sub



  visited_ClosedOld = CType(Session("visited_ClosedOld"), ArrayList)



  Target_ClosedOld.Text = Int32.Parse(ClosedOld.Text) - (visited_ClosedOld.Count - 1)
```

```
End Sub

Private Sub update_Target_ClosedUNR()
  GV_ClosedUNR_update()
  Dim visited_ClosedUNR As New ArrayList
  If Session("visited_ClosedUNR") Is Nothing Then Exit Sub

  visited_ClosedUNR = CType(Session("visited_ClosedUNR"), ArrayList)

  Target_ClosedUNR.Text = Int32.Parse(ClosedUNR.Text) - (visited_ClosedUNR.Count -
1)
End Sub
Private Sub update_Target_ClosedDNU()
  GV_ClosedDNU_update()
  Dim visited_ClosedDNU As New ArrayList
  If Session("visited_ClosedDNU") Is Nothing Then Exit Sub
```

```
visited_ClosedDNU = CType(Session("visited_ClosedDNU"), ArrayList)

Target_ClosedDNUCurrent.Text = Int32.Parse(ClosedDNUCurrent.Text) -
(visited_ClosedDNU.Count - 1)

End Sub

Private Sub update_Target_OpenMainsLTE2()

GV_OpenMainsLTE2_update()

Dim visited_OpenMainsLTE2 As New ArrayList

If Session("visited_OpenMainsLTE2") Is Nothing Then Exit Sub

visited_OpenMainsLTE2 = CType(Session("visited_OpenMainsLTE2"), ArrayList)

Target_OpenMainsLTE2Current.Text = Int32.Parse(OpenMainsLTE2Current.Text) -
(visited_OpenMainsLTE2.Count - 1)

End Sub

Private Sub update_Target_OpenMainsGT2()
```

```
GV_OpenMainsGT2_update()

Dim visited_OpenMainsGT2 As New ArrayList

If Session("visited_OpenMainsGT2") Is Nothing Then Exit Sub


visited_OpenMainsGT2 = CType(Session("visited_OpenMainsGT2"), ArrayList)


Target_OpenMainsGT2Current.Text = Int32.Parse(OpenMainsGT2Current.Text) -
(visited_OpenMainsGT2.Count - 1)

End Sub
```

```
Private Sub update_Target_Missing()

  GV_Missing_update()



  Dim visited_Missing As New ArrayList

  If Session("visited_Missing") Is Nothing Then Exit Sub



  visited_Missing = CType(Session("visited_Missing"), ArrayList)



  Target_MissingFeederCurrent.Text = Int32.Parse(MissingFeederCurrent.Text) -
(visited_Missing.Count - 1)

  'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

  TextBox2.Text = visited_Missing.Count

  'TextBox2.Text = vault_count



End Sub
```

```
Protected Sub GV_OpenSwitches_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_OpenSwitches_NbyFactorGTE10.SelectedIndexChanged

    GV_OpenSwitches_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

    Dim visited_openswitches_NbyFactorGTE10 As ArrayList

    If Session("visited_openSwitches_NbyFactorGTE10") Is Nothing Then

      visited_openswitches_NbyFactorGTE10 = New ArrayList

    Else

      visited_openswitches_NbyFactorGTE10 = CType(Session("visited_openswitches_NbyFactorGTE10"), ArrayList)

    End If

    Dim id As String = GV_OpenSwitches_NbyFactorGTE10.SelectedRow.Cells(0).Text

    If visited_openswitches_NbyFactorGTE10.Count < 1 Then visited_openswitches_NbyFactorGTE10.Add(0)

    If visited_openswitches_NbyFactorGTE10.Contains(id) Then

      Exit Sub

    Else
```

visited_openswitches_NbyFactorGTE10.Add(id)

End If

Session("visited_openswitches_NbyFactorGTE10") = visited_openswitches_NbyFactorGTE10

End Sub

Protected Sub GV_BanksOff_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_BanksOff_NbyFactorGTE10.SelectedIndexChanged

GV_BanksOff_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_BanksOff_NbyFactorGTE10 As ArrayList

If Session("visited_BanksOff_NbyFactorGTE10") Is Nothing Then

visited_BanksOff_NbyFactorGTE10 = New ArrayList

Else

```
visited_BanksOff_NbyFactorGTE10 =
CType(Session("visited_BanksOff_NbyFactorGTE10"), ArrayList)

End If

Dim id As String = GV_BanksOff_NbyFactorGTE10.SelectedRow.Cells(0).Text

If visited_BanksOff_NbyFactorGTE10.Count < 1 Then
visited_BanksOff_NbyFactorGTE10.Add(0)

If visited_BanksOff_NbyFactorGTE10.Contains(id) Then

  Exit Sub

Else

  visited_BanksOff_NbyFactorGTE10.Add(id)

End If



Session("visited_BanksOff_NbyFactorGTE10") = visited_BanksOff_NbyFactorGTE10



End Sub
```

```
Protected Sub GV_Fuse_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles GV_Fuse_NbyFactorGTE10.SelectedIndexChanged

    GV_Fuse_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue



    Dim visited_Fuse_NbyFactorGTE10 As ArrayList

    If Session("visited_Fuse_NbyFactorGTE10") Is Nothing Then

      visited_Fuse_NbyFactorGTE10 = New ArrayList

    Else

      visited_Fuse_NbyFactorGTE10  =  CType(Session("visited_Fuse_NbyFactorGTE10"),
ArrayList)

    End If

    Dim id As String = GV_Fuse_NbyFactorGTE10.SelectedRow.Cells(0).Text

    If visited_Fuse_NbyFactorGTE10.Count < 1 Then visited_Fuse_NbyFactorGTE10.Add(0)

    If visited_Fuse_NbyFactorGTE10.Contains(id) Then

      Exit Sub

    Else
```

```
            visited_Fuse_NbyFactorGTE10.Add(id)

        End If


        Session("visited_Fuse_NbyFactorGTE10") = visited_Fuse_NbyFactorGTE10




    End Sub


    Protected Sub GV_ClosedOld_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles GV_ClosedOld_NbyFactorGTE10.SelectedIndexChanged

        GV_ClosedOld_NbyFactorGTE10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

        Dim visited_ClosedOld_NbyFactorGTE10 As ArrayList

        If Session("visited_ClosedOld_NbyFactorGTE10") Is Nothing Then

            visited_ClosedOld_NbyFactorGTE10 = New ArrayList

        Else
```

```
                visited_ClosedOld_NbyFactorGTE10 =
CType(Session("visited_ClosedOld_NbyFactorGTE10"), ArrayList)

                End If

                Dim id As String = GV_ClosedOld_NbyFactorGTE10.SelectedRow.Cells(0).Text

                If visited_ClosedOld_NbyFactorGTE10.Count < 1 Then
visited_ClosedOld_NbyFactorGTE10.Add(0)

                If visited_ClosedOld_NbyFactorGTE10.Contains(id) Then

                  Exit Sub

                Else

                  visited_ClosedOld_NbyFactorGTE10.Add(id)

                End If



                Session("visited_ClosedOld_NbyFactorGTE10") = visited_ClosedOld_NbyFactorGTE10



            End Sub
```

```
Protected Sub GV_ClosedUNR_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles GV_ClosedUNR_NbyFactorGTE10.SelectedIndexChanged

        GV_ClosedUNR_NbyFactorGTE10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

        Dim visited_ClosedUNR_NbyFactorGTE10 As ArrayList

        If Session("visited_ClosedUNR_NbyFactorGTE10") Is Nothing Then

          visited_ClosedUNR_NbyFactorGTE10 = New ArrayList

        Else

          visited_ClosedUNR_NbyFactorGTE10 = CType(Session("visited_ClosedUNR_NbyFactorGTE10"), ArrayList)

        End If

        Dim id As String = GV_ClosedUNR_NbyFactorGTE10.SelectedRow.Cells(0).Text

        If visited_ClosedUNR_NbyFactorGTE10.Count < 1 Then visited_ClosedUNR_NbyFactorGTE10.Add(0)

        If visited_ClosedUNR_NbyFactorGTE10.Contains(id) Then

          Exit Sub

        Else
```

```
            visited_ClosedUNR_NbyFactorGTE10.Add(id)
        End If

        Session("visited_ClosedUNR_NbyFactorGTE10") =
visited_ClosedUNR_NbyFactorGTE10

    End Sub

    Protected Sub GV_ClosedDNU_NbyFactorGTE10_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles GV_ClosedDNU_NbyFactorGTE10.SelectedIndexChanged

        GV_ClosedDNU_NbyFactorGTE10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

        Dim visited_ClosedDNU_NbyFactorGTE10 As ArrayList

        If Session("visited_ClosedDNU_NbyFactorGTE10") Is Nothing Then

            visited_ClosedDNU_NbyFactorGTE10 = New ArrayList

        Else

            visited_ClosedDNU_NbyFactorGTE10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE10"), ArrayList)
```

```
End If

Dim id As String = GV_ClosedDNU_NbyFactorGTE10.SelectedRow.Cells(0).Text

If visited_ClosedDNU_NbyFactorGTE10.Count < 1 Then
visited_ClosedDNU_NbyFactorGTE10.Add(0)

If visited_ClosedDNU_NbyFactorGTE10.Contains(id) Then

  Exit Sub

Else

  visited_ClosedDNU_NbyFactorGTE10.Add(id)

End If


Session("visited_ClosedDNU_NbyFactorGTE10") =
visited_ClosedDNU_NbyFactorGTE10


End Sub


Protected Sub GV_OpenMainsLTE2_NbyFactorGTE10_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs) Handles
GV_OpenMainsLTE2_NbyFactorGTE10.SelectedIndexChanged
```

```
GV_OpenMainsLTE2_NbyFactorGTE10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_OpenMainsLTE2_NbyFactorGTE10 As ArrayList

If Session("visited_OpenMainsLTE2_NbyFactorGTE10") Is Nothing Then

  visited_OpenMainsLTE2_NbyFactorGTE10 = New ArrayList

Else

  visited_OpenMainsLTE2_NbyFactorGTE10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE10"), ArrayList)

End If

Dim id As String = GV_OpenMainsLTE2_NbyFactorGTE10.SelectedRow.Cells(0).Text

If visited_OpenMainsLTE2_NbyFactorGTE10.Count < 1 Then
visited_OpenMainsLTE2_NbyFactorGTE10.Add(0)

If visited_OpenMainsLTE2_NbyFactorGTE10.Contains(id) Then

  Exit Sub

Else

  visited_OpenMainsLTE2_NbyFactorGTE10.Add(id)

End If
```

```
                Session("visited_OpenMainsLTE2_NbyFactorGTE10") =
visited_OpenMainsLTE2_NbyFactorGTE10


                End Sub

                Protected Sub GV_OpenMainsGT2_NbyFactorGTE10_SelectedIndexChanged(ByVal sender
As Object, ByVal e As System.EventArgs) Handles GV_OpenMainsGT2_NbyFactorGTE10.SelectedIndexChanged

                GV_OpenMainsGT2_NbyFactorGTE10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

                Dim visited_OpenMainsGT2_NbyFactorGTE10 As ArrayList

                If Session("visited_OpenMainsGT2_NbyFactorGTE10") Is Nothing Then

                  visited_OpenMainsGT2_NbyFactorGTE10 = New ArrayList

                Else

                  visited_OpenMainsGT2_NbyFactorGTE10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE10"), ArrayList)

                End If

                Dim id As String = GV_OpenMainsGT2_NbyFactorGTE10.SelectedRow.Cells(0).Text

                If visited_OpenMainsGT2_NbyFactorGTE10.Count < 1 Then
visited_OpenMainsGT2_NbyFactorGTE10.Add(0)

                If visited_OpenMainsGT2_NbyFactorGTE10.Contains(id) Then
```

```
        Exit Sub
      Else
        visited_OpenMainsGT2_NbyFactorGTE10.Add(id)
      End If

      Session("visited_OpenMainsGT2_NbyFactorGTE10") =
visited_OpenMainsGT2_NbyFactorGTE10

    End Sub

    'code to update values in GTE10 columns
    Private Sub update_Target_OpenSwitches_NbyFactorGTE10()
      GV_OpenSwitches_NbyFactorGTE10_update()
      Dim visited_OpenSwitches_NbyFactorGTE10 As New ArrayList
```

```
If Session("visited_OpenSwitches_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_OpenSwitches_NbyFactorGTE10 = CType(Session("visited_OpenSwitches_NbyFactorGTE10"), ArrayList)

Target_OpenSwitches_GTE10.Text = Int32.Parse(OpenSwitches_GTE10.Text) - (visited_OpenSwitches_NbyFactorGTE10.Count - 1)

'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

'TextBox2.Text = visited_OpenSwitches_NbyFactorGTE10.Count

'TextBox2.Text = vault_count

End Sub

Private Sub update_Target_Fuse_NbyFactorGTE10()

GV_fuse_NbyFactorGTE10_update()

Dim visited_fuse_NbyFactorGTE10 As New ArrayList

If Session("visited_fuse_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_fuse_NbyFactorGTE10 = CType(Session("visited_fuse_NbyFactorGTE10"), ArrayList)
```

Target_Fuse_GTE10.Text = Int32.Parse(Fuse_GTE10.Text) - (visited_fuse_NbyFactorGTE10.Count - 1)

Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 * Int32.Parse(Target_Fuse_GTE10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)

End Sub

Private Sub update_Target_BanksOff_NbyFactorGTE10()

GV_BanksOff_NbyFactorGTE10_update()

Dim visited_BanksOff_NbyFactorGTE10 As New ArrayList

If Session("visited_BanksOff_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_BanksOff_NbyFactorGTE10 = CType(Session("visited_BanksOff_NbyFactorGTE10"), ArrayList)

Target_BanksOff_GTE10.Text = Int32.Parse(BanksOff_GTE10.Text) - (visited_BanksOff_NbyFactorGTE10.Count - 1)

End Sub

Private Sub update_Target_ClosedOld_NbyFactorGTE10()

```
GV_ClosedOld_NbyFactorGTE10_update()

Dim visited_ClosedOld_NbyFactorGTE10 As New ArrayList

If Session("visited_ClosedOld_NbyFactorGTE10") Is Nothing Then Exit Sub


visited_ClosedOld_NbyFactorGTE10 =
CType(Session("visited_ClosedOld_NbyFactorGTE10"), ArrayList)


Target_ClosedOld_GTE10.Text = Int32.Parse(ClosedOld_GTE10.Text) -
(visited_ClosedOld_NbyFactorGTE10.Count - 1)


End Sub


Private Sub update_Target_ClosedUNR_NbyFactorGTE10()

GV_ClosedUNR_NbyFactorGTE10_update()

Dim visited_ClosedUNR_NbyFactorGTE10 As New ArrayList

If Session("visited_ClosedUNR_NbyFactorGTE10") Is Nothing Then Exit Sub
```

```
visited_ClosedUNR_NbyFactorGTE10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE10"), ArrayList)

Target_ClosedUNR_GTE10.Text = Int32.Parse(ClosedUNR_GTE10.Text) -
(visited_ClosedUNR_NbyFactorGTE10.Count - 1)

End Sub

Private Sub update_Target_ClosedDNU_NbyFactorGTE10()

GV_ClosedDNU_NbyFactorGTE10_update()

Dim visited_ClosedDNU_NbyFactorGTE10 As New ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_ClosedDNU_NbyFactorGTE10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE10"), ArrayList)

Target_ClosedDNU_GTE10.Text = Int32.Parse(ClosedDNU_GTE10.Text) -
(visited_ClosedDNU_NbyFactorGTE10.Count - 1)

End Sub

Private Sub update_Target_Missing_NbyFactorGTE10()
```

```
GV_Missing_NbyFactorGTE10_update()

Dim visited_Missing_NbyFactorGTE10 As New ArrayList

If Session("visited_Missing_NbyFactorGTE10") Is Nothing Then Exit Sub


visited_Missing_NbyFactorGTE10 =
CType(Session("visited_Missing_NbyFactorGTE10"), ArrayList)


Target_Missing_GTE10.Text = Int32.Parse(ClosedDNU_GTE10.Text) -
(visited_Missing_NbyFactorGTE10.Count - 1)

End Sub


Private Sub update_Target_OpenMainsLTE2_NbyFactorGTE10()

GV_OpenMainsLTE2_NbyFactorGTE10_update()


Dim visited_OpenMainsLTE2_NbyFactorGTE10 As New ArrayList

If Session("visited_OpenMainsLTE2_NbyFactorGTE10") Is Nothing Then Exit Sub
```

visited_OpenMainsLTE2_NbyFactorGTE10 = CType(Session("visited_OpenMainsLTE2_NbyFactorGTE10"), ArrayList)

Target_OMLTE2_GTE10.Text = Int32.Parse(OMLTE2_GTE10.Text) - (visited_OpenMainsLTE2_NbyFactorGTE10.Count - 1)

End Sub

Private Sub update_Target_OpenMainsGT2_NbyFactorGTE10()

GV_OpenMainsGT2_NbyFactorGTE10_update()

Dim visited_OpenMainsGT2_NbyFactorGTE10 As New ArrayList

If Session("visited_OpenMainsGT2_NbyFactorGTE10") Is Nothing Then Exit Sub

visited_OpenMainsGT2_NbyFactorGTE10 = CType(Session("visited_OpenMainsGT2_NbyFactorGTE10"), ArrayList)

Target_OMGT2_GTE10.Text = Int32.Parse(OMGT2_GTE10.Text) - (visited_OpenMainsGT2_NbyFactorGTE10.Count - 1)

```
End Sub


'code to update values in GTE5LT10 columns
Private Sub update_Target_OpenSwitches_NbyFactorGTE5LT10()
    GV_OpenSwitches_NbyFactorGTE5LT10_update()
    Dim visited_OpenSwitches_NbyFactorGTE5LT10 As New ArrayList
    If Session("visited_OpenSwitches_NbyFactorGTE5LT10") Is Nothing Then Exit Sub
```

```
                visited_OpenSwitches_NbyFactorGTE5LT10 =
CType(Session("visited_OpenSwitches_NbyFactorGTE5LT10"), ArrayList)

                Target_OpenSwitches_GTE5LT10.Text = Int32.Parse(OpenSwitches_GTE5LT10.Text) -
(visited_OpenSwitches_NbyFactorGTE5LT10.Count - 1)

                'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

                TextBox2.Text = visited_OpenSwitches_NbyFactorGTE5LT10.Count

                'TextBox2.Text = vault_count


            End Sub


            Private Sub update_Target_Fuse_NbyFactorGTE5LT10()
              GV_fuse_NbyFactorGTE5LT10_update()
              Dim visited_fuse_NbyFactorGTE5LT10 As New ArrayList
              If Session("visited_fuse_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

                visited_fuse_NbyFactorGTE5LT10 =
CType(Session("visited_fuse_NbyFactorGTE5LT10"), ArrayList)
```

```
Target_Fuse_GTE5LT10.Text = Int32.Parse(Fuse_GTE5LT10.Text) - (visited_fuse_NbyFactorGTE5LT10.Count - 1)

Target_Sum_Fuse.Text = 5 * Int32.Parse(Target_FuseFeederCurrent.Text) + 5 * Int32.Parse(Target_Fuse_GTE5LT10.Text) + 3 * Int32.Parse(Target_Fuse_GTE5LT10.Text)

End Sub

Private Sub update_Target_BanksOff_NbyFactorGTE5LT10()

GV_BanksOff_NbyFactorGTE5LT10_update()

Dim visited_BanksOff_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_BanksOff_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_BanksOff_NbyFactorGTE5LT10 = CType(Session("visited_BanksOff_NbyFactorGTE5LT10"), ArrayList)

Target_BanksOff_GTE5LT10.Text = Int32.Parse(BanksOff_GTE5LT10.Text) - (visited_BanksOff_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_ClosedOld_NbyFactorGTE5LT10()
```

```
GV_ClosedOld_NbyFactorGTE5LT10_update()

Dim visited_ClosedOld_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_ClosedOld_NbyFactorGTE5LT10") Is Nothing Then Exit Sub


visited_ClosedOld_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedOld_NbyFactorGTE5LT10"), ArrayList)


Target_ClosedOld_GTE5LT10.Text = Int32.Parse(ClosedOld_GTE5LT10.Text) -
(visited_ClosedOld_NbyFactorGTE5LT10.Count - 1)


End Sub


Private Sub update_Target_ClosedUNR_NbyFactorGTE5LT10()

GV_ClosedUNR_NbyFactorGTE5LT10_update()

Dim visited_ClosedUNR_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_ClosedUNR_NbyFactorGTE5LT10") Is Nothing Then Exit Sub
```

```
visited_ClosedUNR_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedUNR_NbyFactorGTE5LT10"), ArrayList)

Target_ClosedUNR_GTE5LT10.Text = Int32.Parse(ClosedUNR_GTE5LT10.Text) -
(visited_ClosedUNR_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_ClosedDNU_NbyFactorGTE5LT10()

GV_ClosedDNU_NbyFactorGTE5LT10_update()

Dim visited_ClosedDNU_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_ClosedDNU_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE5LT10"), ArrayList)

Target_ClosedDNU_GTE5LT10.Text = Int32.Parse(ClosedDNU_GTE5LT10.Text) -
(visited_ClosedDNU_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_OpenMainsLTE2_NbyFactorGTE5LT10()

GV_OpenMainsLTE2_NbyFactorGTE5LT10_update()
```

```
Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_OpenMainsLTE2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10"), ArrayList)

Target_OMLTE2_GTE5LT10.Text = Int32.Parse(OMLTE2_GTE5LT10.Text) -
(visited_OpenMainsLTE2_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_OpenMainsGT2_NbyFactorGTE5LT10()

GV_OpenMainsGT2_NbyFactorGTE5LT10_update()

Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As New ArrayList

If Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

visited_OpenMainsGT2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE5LT10"), ArrayList)
```

```
Target_OMGT2_GTE5LT10.Text = Int32.Parse(OMGT2_GTE5LT10.Text) -
(visited_OpenMainsGT2_NbyFactorGTE5LT10.Count - 1)

End Sub

Private Sub update_Target_missing_NbyFactorGTE5LT10()

  GV_missing_NbyFactorGTE5LT10_update()

  Dim visited_missing_NbyFactorGTE5LT10 As New ArrayList

  If Session("visited_missing_NbyFactorGTE5LT10") Is Nothing Then Exit Sub

  visited_missing_NbyFactorGTE5LT10 =
CType(Session("visited_missing_NbyFactorGTE5LT10"), ArrayList)

  Target_Missing_GTE5LT10.Text = Int32.Parse(Missing_GTE5LT10.Text) -
(visited_missing_NbyFactorGTE5LT10.Count - 1)

  'TextBox2.Text = GV_OpenSwitches.SelectedRow.Cells(0).Text

  TextBox2.Text = visited_missing_NbyFactorGTE5LT10.Count

  'TextBox2.Text = vault_count
```

```
End Sub

Protected Sub GV_OpenSwitches_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs)

GV_OpenSwitches_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_openswitches_NbyFactorGTE5LT10 As ArrayList

If Session("visited_openSwitches_NbyFactorGTE5LT10") Is Nothing Then

  visited_openswitches_NbyFactorGTE5LT10 = New ArrayList

Else

  visited_openswitches_NbyFactorGTE5LT10 =
CType(Session("visited_openswitches_NbyFactorGTE5LT10"), ArrayList)

End If

Dim id As String = GV_OpenSwitches_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_openswitches_NbyFactorGTE5LT10.Count < 1 Then
visited_openswitches_NbyFactorGTE5LT10.Add(0)

If visited_openswitches_NbyFactorGTE5LT10.Contains(id) Then
```

```
            Exit Sub
        Else
            visited_openswitches_NbyFactorGTE5LT10.Add(id)
        End If

        Session("visited_openswitches_NbyFactorGTE5LT10") =
visited_openswitches_NbyFactorGTE5LT10

    End Sub

    Protected Sub GV_BanksOff_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender
As Object, ByVal e As System.EventArgs)
        GV_BanksOff_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue
        Dim visited_BanksOff_NbyFactorGTE5LT10 As ArrayList
        If Session("visited_BanksOff_NbyFactorGTE5LT10") Is Nothing Then
            visited_BanksOff_NbyFactorGTE5LT10 = New ArrayList
        Else
```

```
                visited_BanksOff_NbyFactorGTE5LT10 =
CType(Session("visited_BanksOff_NbyFactorGTE5LT10"), ArrayList)

                End If

                Dim id As String = GV_BanksOff_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

                If visited_BanksOff_NbyFactorGTE5LT10.Count < 1 Then
visited_BanksOff_NbyFactorGTE5LT10.Add(0)

                If visited_BanksOff_NbyFactorGTE5LT10.Contains(id) Then

                  Exit Sub

                Else

                  visited_BanksOff_NbyFactorGTE5LT10.Add(id)

                End If



                Session("visited_BanksOff_NbyFactorGTE5LT10") =
visited_BanksOff_NbyFactorGTE5LT10




              End Sub
```

```
Protected Sub GV_Fuse_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)

    GV_Fuse_NbyFactorGTE5LT10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue



    Dim visited_Fuse_NbyFactorGTE5LT10 As ArrayList

    If Session("visited_Fuse_NbyFactorGTE5LT10") Is Nothing Then

      visited_Fuse_NbyFactorGTE5LT10 = New ArrayList

    Else

      visited_Fuse_NbyFactorGTE5LT10 = CType(Session("visited_Fuse_NbyFactorGTE5LT10"), ArrayList)

    End If

    Dim id As String = GV_Fuse_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

    If visited_Fuse_NbyFactorGTE5LT10.Count < 1 Then visited_Fuse_NbyFactorGTE5LT10.Add(0)

    If visited_Fuse_NbyFactorGTE5LT10.Contains(id) Then

      Exit Sub

    Else

      visited_Fuse_NbyFactorGTE5LT10.Add(id)
```

```
End If

Session("visited_Fuse_NbyFactorGTE5LT10") = visited_Fuse_NbyFactorGTE5LT10

End Sub

Protected Sub GV_ClosedOld_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender
As Object, ByVal e As System.EventArgs)

GV_ClosedOld_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_ClosedOld_NbyFactorGTE5LT10 As ArrayList

If Session("visited_ClosedOld_NbyFactorGTE5LT10") Is Nothing Then

  visited_ClosedOld_NbyFactorGTE5LT10 = New ArrayList

Else

  visited_ClosedOld_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedOld_NbyFactorGTE5LT10"), ArrayList)

End If
```

```
Dim id As String = GV_ClosedOld_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_ClosedOld_NbyFactorGTE5LT10.Count < 1 Then visited_ClosedOld_NbyFactorGTE5LT10.Add(0)

If visited_ClosedOld_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else

  visited_ClosedOld_NbyFactorGTE5LT10.Add(id)

End If

Session("visited_ClosedOld_NbyFactorGTE5LT10") = visited_ClosedOld_NbyFactorGTE5LT10

End Sub

Protected Sub GV_ClosedUNR_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)

GV_ClosedUNR_NbyFactorGTE5LT10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue
```

```
Dim visited_ClosedUNR_NbyFactorGTE5LT10 As ArrayList
If Session("visited_ClosedUNR_NbyFactorGTE5LT10") Is Nothing Then
  visited_ClosedUNR_NbyFactorGTE5LT10 = New ArrayList
Else
  visited_ClosedUNR_NbyFactorGTE5LT10 = CType(Session("visited_ClosedUNR_NbyFactorGTE5LT10"), ArrayList)
End If
Dim id As String = GV_ClosedUNR_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text
If visited_ClosedUNR_NbyFactorGTE5LT10.Count < 1 Then visited_ClosedUNR_NbyFactorGTE5LT10.Add(0)
If visited_ClosedUNR_NbyFactorGTE5LT10.Contains(id) Then
  Exit Sub
Else
  visited_ClosedUNR_NbyFactorGTE5LT10.Add(id)
End If

Session("visited_ClosedUNR_NbyFactorGTE5LT10") = visited_ClosedUNR_NbyFactorGTE5LT10
```

```
End Sub

Protected Sub GV_ClosedDNU_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs)

GV_ClosedDNU_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_ClosedDNU_NbyFactorGTE5LT10 As ArrayList

If Session("visited_ClosedDNU_NbyFactorGTE5LT10") Is Nothing Then

  visited_ClosedDNU_NbyFactorGTE5LT10 = New ArrayList

Else

  visited_ClosedDNU_NbyFactorGTE5LT10 =
CType(Session("visited_ClosedDNU_NbyFactorGTE5LT10"), ArrayList)

End If

Dim id As String = GV_ClosedDNU_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_ClosedDNU_NbyFactorGTE5LT10.Count < 1 Then
visited_ClosedDNU_NbyFactorGTE5LT10.Add(0)

If visited_ClosedDNU_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else
```

visited_ClosedDNU_NbyFactorGTE5LT10.Add(id)

End If

Session("visited_ClosedDNU_NbyFactorGTE5LT10") = visited_ClosedDNU_NbyFactorGTE5LT10

End Sub

Protected Sub GV_OpenMainsLTE2_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)

GV_OpenMainsLTE2_NbyFactorGTE5LT10.SelectedRow.BackColor = Drawing.Color.LightSkyBlue

Dim visited_OpenMainsLTE2_NbyFactorGTE5LT10 As ArrayList

If Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") Is Nothing Then

visited_OpenMainsLTE2_NbyFactorGTE5LT10 = New ArrayList

Else

visited_OpenMainsLTE2_NbyFactorGTE5LT10 = CType(Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10"), ArrayList)

End If

```
Dim id As String = GV_OpenMainsLTE2_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text

If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Count < 1 Then visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(0)

If visited_OpenMainsLTE2_NbyFactorGTE5LT10.Contains(id) Then

  Exit Sub

Else

  visited_OpenMainsLTE2_NbyFactorGTE5LT10.Add(id)

End If



Session("visited_OpenMainsLTE2_NbyFactorGTE5LT10") = visited_OpenMainsLTE2_NbyFactorGTE5LT10



End Sub



Protected Sub GV_OpenMainsGT2_NbyFactorGTE5LT10_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs)
```

```
GV_OpenMainsGT2_NbyFactorGTE5LT10.SelectedRow.BackColor =
Drawing.Color.LightSkyBlue

Dim visited_OpenMainsGT2_NbyFactorGTE5LT10 As ArrayList
If Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") Is Nothing Then
  visited_OpenMainsGT2_NbyFactorGTE5LT10 = New ArrayList
Else
  visited_OpenMainsGT2_NbyFactorGTE5LT10 =
CType(Session("visited_OpenMainsGT2_NbyFactorGTE5LT10"), ArrayList)
End If
Dim id As String =
GV_OpenMainsGT2_NbyFactorGTE5LT10.SelectedRow.Cells(0).Text
If visited_OpenMainsGT2_NbyFactorGTE5LT10.Count < 1 Then
visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(0)
If visited_OpenMainsGT2_NbyFactorGTE5LT10.Contains(id) Then
  Exit Sub
Else
  visited_OpenMainsGT2_NbyFactorGTE5LT10.Add(id)
End If
```

```
Session("visited_OpenMainsGT2_NbyFactorGTE5LT10") =
visited_OpenMainsGT2_NbyFactorGTE5LT10


End Sub



Private Sub IR_Scandatetime()
  Dim feeder As String
  feeder = cmbSearchFeeder.SelectedValue.ToString
  'If feeder.Length < 5 Then
  '   feeder = "0" + feeder.ToString
  'End If


  'Dim sql As String = "SELECT * FROM LoadPocket.dbo.latest_lpw WHERE Feeder = '"
& cmbSearchFeeder.SelectedValue & "'"
  'Dim sql As String = "SELECT * FROM
[Transformer_OM_BO].[dbo].[loadpocketweight_test] WHERE Feeder ='" + feeder + "' And
```

IR_nwk_scan_datetime = (select max(IR_nwk_scan_datetime) from [Transformer_OM_BO].[dbo].[loadpocketweight_test] where Feeder ='" + feeder + "')"

Dim sql As String = "SELECT * FROM LoadPocket.dbo.latest_lpw WHERE Feeder ='" + feeder + "' And nwk_scan_datetime = (select max(nwk_scan_datetime) from LoadPocket.dbo.latest_lpw where Feeder ='" + feeder + "')"

'Dim sql As String = "SELECT * FROM rerank_dynamic_Attrib WHERE Feeder = '" & cmbSearchFeeder.SelectedValue & "'"

'Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("DB2FeederSuscept").ConnectionString)

'wDim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Vault").ConnectionString)

Dim myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("Attributes").ConnectionString)

Dim myCommand As New SqlDataAdapter(sql, myConnection)

myCommand.SelectCommand.CommandTimeout = 300

Dim dt As DataTable = New DataTable

myCommand.Fill(dt)

```
If dt.Rows.Count = 0 Then

   Exit Sub

End If




Dim dr As DataRow = dt.Rows(0)

'getting the rank from the precomputed table now

'RankBaseLoad.Text = drcheck(dr, "Rank")

IR_ScanTime.Text = dr("nwk_scan_datetime")



myCommand = Nothing

myConnection.Close()

myConnection = Nothing



End Sub



Protected Sub GVPILCRuns_Select_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles GVPILCRuns_Select.SelectedIndexChanged
```

```
Dim dt As DataTable

Dim Run As Integer

Dim feeder As String

feeder = cmbSearchFeeder.SelectedValue

Run = Int32.Parse(GVPILCRuns_Select.SelectedItem.Cells(2).Text)



dt = GetSectionInRunTable(Run, feeder)



Dim struct As String

struct = dt.Rows(0).Item("from_struct")

Session("struct") = struct



Dim URL As String



URL = "feedermaps/" & cmbSearchFeeder.SelectedValue & "_map.aspx"
```

```
Dim MyScript1 As New System.Text.StringBuilder
With MyScript1
  .Append("<script type='text/javascript'>")
  .Append("window.open('" & URL & "');")
  .Append("</script>")
End With

'Page.ClientScript.RegisterStartupScript(GetType(String), "FeederMap",
MyScript.ToString)
ScriptManager.RegisterStartupScript(Me, Me.GetType(), "FeederMap",
MyScript1.ToString, False)

'RegisterStartupScript(Me, Me.GetType(), "FeederMap1", MyScript1.ToString, False)

'Label2.Text = GVovlnonsec.SelectedRow.Cells(3).Text
```

```
        Dim m_deltaMTBF As String

        m_deltaMTBF = Convert.ToString(ConfigurationManager.AppSettings("deltaMTBF"))

        'm_currYear  = Convert.ToInt32(ConfigurationManager.AppSettings["year"]);



        If Equals(m_deltaMTBF, "no") Then

           GVPILCRuns_Select.Columns(7).Visible = False

        End If



    End Sub



        Protected Sub GVPILCRuns_Select_SortCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridSortCommandEventArgs) Handles GVPILCRuns_Select.SortCommand



        Dim dv As DataView = New DataView(GetPILCLoadTable(Feeder.Text))

        Dim dv1 As DataView = New DataView(GetPILCRunLoadTable(Feeder.Text))



        ' dv1.Sort = e.SortExpression + " " + "desc"
```

```
dv1.Sort = e.SortExpression

If ViewState("sortdirection").ToString.Equals("ASC") Then

   dv1.Sort = e.SortExpression + " " + "desc"

   ViewState("sortdirection") = "DESC"

Else

   dv1.Sort = e.SortExpression + " " + "ASC"

   ViewState("sortdirection") = "ASC"

End If



Dim dss As DataSet = New DataSet()

Dim dt1 As DataTable = New DataTable("PILCRuns")

Dim dt2 As DataTable = New DataTable("PICLSections")

dt1 = dv1.ToTable()

dt2 = dv.ToTable()

dss.Tables.Add(dt1)

dss.Tables.Add(dt2)

Dim dcc1 As DataColumn

Dim dcc2 As DataColumn
```

```
'Relation dv => dv1 on Run_ID column

dcc1 = dss.Tables(0).Columns("Run_ID")

dcc2 = dss.Tables(1).Columns("Run_ID")

Dim drr As DataRelation = New DataRelation("Run_Section", dcc1, dcc2, False)

dss.Relations.Add(drr)

'bind the DataSet to the HierarGrid

GVPILCRuns_Select.DataSource = dss

GVPILCRuns_Select.DataMember = dss.Tables(0).TableName()

GVPILCRuns_Select.DataBind()

Dim visited_run_grid As Integer

For visited_run_grid = 0 To (GVPILCRuns_Select.Items.Count) - 1

  If                                                                    Not
(Equals(VisitedGlbRuns.Items.FindByText(GVPILCRuns_Select.Items(visited_run_grid).Cells(2).Text),  Nothing))
Then

    GVPILCRuns_Select.Items(visited_run_grid).ForeColor = Drawing.Color.Red

  End If
```

```
Next

End Sub

Protected Sub GVPILCRuns_Select_TemplateSelection(ByVal sender As Object, ByVal e
As DBauer.Web.UI.WebControls.HierarGridTemplateSelectionEventArgs) Handles
GVPILCRuns_Select.TemplateSelection

e.TemplateFilename = "Templates\\PILCSections.ascx"

End Sub
```

End Class

End Namespace

APPENDIX C

```
using System;

using System.Collections;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Web;

using System.Web.SessionState;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.HtmlControls;

using System.Text;

using System.Data.SqlClient;

using System.Configuration;

using System.Web.Mail;

using System.IO;

using System.Threading;

using FeederModelMgr;

using System.Reflection;
```

```
public partial class Precomp : System.Web.UI.Page
{
    #region protected and private fields
    protected SqlConnection m_conn;
    protected int m_runTot, m_loadRunTot, m_pilcRunTot;
    protected int m_seqID, m_loadSeqID, m_pilcSeqID;
    protected int m_loadPer, m_pilcPer, m_relPer;
    protected int m_maxSeqID;
    protected decimal m_maxFreqSum, m_maxFreqWtdNor;
    protected float m_actualPer;
    protected string m_pilcStrat, m_relStrat;
    protected string m_colName, m_colVal;
    protected string m_Error;
    protected string m_feeder;
    private DataView LRdv, PILCdv, SJdv, SJdv_1, XLPdv;
    private StringBuilder m_sqlFdrAttrs, m_sqlRerank;
    private ArrayList m_replacedLoadRuns, m_replacedLoadSections, m_replacedPilcRuns;
```

```
private FeederModelMgr.FdrModMgr m_fdrMgr;
private Hashtable FdrAttrsHt;
#endregion

protected void Page_Load(object sender, EventArgs e)
{
  m_conn = new SqlConnection(ConfigurationManager.AppSettings["dbConn"]);
  m_conn.Open();

  try
  {
    if (!IsPostBack)
    {
      /* Get max version # */
      GetMaxVersionNum();
    }
  }
  catch (SqlException sEx)
```

```
    {
        m_Error += sEx.Message.ToString();
        ErrorMessage();
    }
    catch (Exception ex)
    {
        m_Error += ex.Message.ToString();
        ErrorMessage();
    }
    finally
    {
        m_conn.Close();
    }
}
protected void btnCompute_Click(object sender, EventArgs e)
{
    //test section
    //m_fdrMgr = new FeederModelMgr.FdrModMgr("", "", "01b56");
```

```
//m_fdrMgr.ChangeFeeder("01b56");

//Hashtable ht = m_fdrMgr.CountMe();

//Hashtable ht = m_fdrMgr.ReplaceMe(104439, true);

//PrepRerankSql(ht);



//int p = Convert.ToInt32(ht["Static_CSR_le_30percent"]);

//int j = m_fdrMgr.JointsLs.Count;

//int c = m_fdrMgr.SectionsLs.Count;

//int s = m_fdrMgr.StrucLs.Count;

//return;



if (!ValidateFields())

   return;



try

{

   FdrAttrsHt = new Hashtable();

   m_maxSeqID = GetMaxSeqID();
```

```
m_sqlRerank = new StringBuilder();

m_sqlFdrAttrs = new StringBuilder();



/* REPLACE A La Cart */

if (chkALaCart.Checked)

{

   DropIndexes_ALaCart();

   ReplaceALaCart();

   RebuilIndexes_ALaCart();

   return;

}



/* REBUILD freq hipot base table */

if (chkHipotBase.Checked)

{

   RebuildHipotBase();

   return;

}
```

```
/* Fetch data for requested feeders */

string boro = "";

string network = "";

string feeder = "";

DataView dv = new DataView(GetFdrNwkBoro(ref boro, ref network, ref feeder));

m_fdrMgr = new FeederModelMgr.FdrModMgr(boro, network, feeder);

PopulateData();



/* Drop index for faster insertion */

DropIndexes();



for (int i = 0; i < dv.Count; i++)

{

   network = dv[i].Row["network"].ToString();

   boro = dv[i].Row["borough"].ToString();

   m_feeder = dv[i].Row["feeder"].ToString();
```

```
if (m_fdrMgr.ChangeFeeder(m_feeder))    //skip feeder with no cables or joints
   continue;

FeederModelMgr.Feeder oFeeder =
(FeederModelMgr.Feeder)m_fdrMgr.oFeeder.Clone(); //set check point

for (int x = 1; x < ddlLoadPer.Items.Count; x++)
{
   /* Replace Load Relief */
   if (chkLoadRelief.Checked)
   {
      if (ddlLoadPer.SelectedIndex > 0)

ReplaceLoadRelief(Convert.ToInt32(ddlLoadPer.Items[ddlLoadPer.SelectedIndex].Value));
      else
         ReplaceLoadRelief(Convert.ToInt32(ddlLoadPer.Items[x].Value));
   }
```

```
/* Replace PILC */
if (chkPilc.Checked)
   PILCSecRun();

/* Replace standalone Reliability */
if (chkRel.Checked && !chkLoadRelief.Checked && !chkPilc.Checked)
   RelSecRun();

///* Save what we just did into db */
//SaveBulkData();

//set break for selected load relief percentage
if ((chkLoadRelief.Checked && ddlLoadPer.SelectedIndex != 0) ||
(!chkLoadRelief.Checked))
   break;
else
{
   //reset to check point for next strategy
```

```
                m_fdrMgr.oFeeder = (FeederModelMgr.Feeder)oFeeder.Clone();

            }


        } //end for x


        /* Save what we just did into db */

        SaveBulkData();


    } //end for i


    /* Rebuild the indexes */

    RebuilIndexes();


    /* CONFIRMATION message */

    m_Error = "Requested Transaction is completed.\nPlease check table PRECOM_Rerank
for results.";


}
```

```
    catch (SqlException sEx)
    {
        m_Error += sEx.Message.ToString();
    }
    catch (Exception ex)
    {
        m_Error += ex.Message.ToString();
    }
    finally
    {
        m_conn.Close();
        ErrorMessage(m_Error);
    }
}

/// <summary>
/// Drop indexes for faster insertion
/// </summary>
```

```
private void DropIndexes()
{
   if (m_conn.State == ConnectionState.Closed)
      m_conn.Open();

   string sql = "exec spDropPrecomIndexes";
   SqlCommand cmd = new SqlCommand(sql, m_conn);
   cmd.ExecuteNonQuery();

   cmd = null;
}
private void RebuilIndexes()
{
   if (m_conn.State == ConnectionState.Closed)
      m_conn.Open();

   string sql = "exec spBuildPrecomIndexes";
   SqlCommand cmd = new SqlCommand(sql, m_conn);
```

```
    cmd.CommandTimeout = 180;
    cmd.ExecuteNonQuery();

    cmd = null;
}
private void DropIndexes_ALaCart()
{
    if (m_conn.State == ConnectionState.Closed)
        m_conn.Open();

    string sql = "exec spDropPrecomIndexes_ALaCart";
    SqlCommand cmd = new SqlCommand(sql, m_conn);
    cmd.ExecuteNonQuery();

    cmd = null;
}
private void RebuilIndexes_ALaCart()
{
```

```
    if (m_conn.State == ConnectionState.Closed)
        m_conn.Open();

    string sql = "exec spBuildPrecomIndexes_ALaCart";
    SqlCommand cmd = new SqlCommand(sql, m_conn);
    cmd.CommandTimeout = 180;
    cmd.ExecuteNonQuery();

    cmd = null;
}

private void PopulateData()
{
    /* Connection */
    if (m_conn.State == ConnectionState.Closed)
        m_conn.Open();

    /* LOAD RELIEF */
```

```
if (chkLoadRelief.Checked)
    LRdv = new DataView(PopulateLoadRelief());

/* PILC */
if ((chkPilc.Checked) && (ddlPilcStrat.SelectedIndex != 1))
    PILCdv = new DataView(PopulatePILCSecRun());

/* XLP */
if ((chkRel.Checked) && (ddlRelStrat.SelectedIndex != 1))
    XLPdv = new DataView(PopulateXLP());

/* STOPJOINTS */
if ((chkPilc.Checked && ddlPilcStrat.SelectedIndex < 2) || (chkRel.Checked &&
ddlRelStrat.SelectedIndex < 2))
    PopulateStopJoints();
}
private DataTable PopulateLoadRelief()
{
```

```
m_Error = "ERROR at PopulateLoadRelief.\n";



/* Prepare to select by network, borough or feeder */

string str1 = "";

string str2 = "";

if (ddlFeeder.SelectedIndex > 0)

{

   str1 = "AND feeder='" + ddlFeeder.Items[ddlFeeder.SelectedIndex].Value.Trim() + "' ";

   str2 = "AND feeder='" + ddlFeeder.Items[ddlFeeder.SelectedIndex].Value.Trim() + "' ";

}

else if (ddlNetwork.SelectedIndex > 0)

{

   string nwk = ddlNetwork.Items[ddlNetwork.SelectedIndex].Value.Trim();

   if (nwk.Length == 2)

      nwk = "0" + nwk;



   str1 = "AND feeder like '" + nwk + "%' ";
```

```
                str2 = "AND nwk='" + ddlNetwork.Items[ddlNetwork.SelectedIndex].Value.Trim() + "'
";

            }

            else if (ddlBorough.SelectedIndex > 0)

            {

                str1 = "AND feeder like '%" +
ddlBorough.Items[ddlBorough.SelectedIndex].Value.Trim() + "%' ";

                str2 = "AND csa='" + ddlBorough.Items[ddlBorough.SelectedIndex].Value.Trim() + "'
";

            }



            /* Replace a whole run for overloaded pilc but per section for other types */

            string sql = "SELECT feeder, run_id, length, 0 as vision_id, max(percentage_load) as
percentage, insulation " +

                "FROM Rerank_Load_Runs WHERE percentage_load >= 100 AND insulation='paper'
" + str1 +

                "GROUP BY feeder, run_id, length, insulation " +

                "UNION " +

                "SELECT feeder, run_id, 1 as length, vision_id, max(load_percentage) as percentage,
insulation " +
```

```
                "FROM Rerank_Load WHERE load_percentage >= 100 AND isnull(insulation,'UNK')
!= 'paper' " + str2 +
                "GROUP BY feeder, run_id, vision_id, insulation " +
                "ORDER BY feeder, run_id, percentage";

            /* Fetch load relief data */
            SqlDataAdapter da = new SqlDataAdapter(sql, m_conn);
            DataTable dt = new DataTable();
            da.Fill(dt);
            da = null;
            return dt;
        }
        private DataTable PopulatePILCSecRun()
        {
            m_Error = "ERROR at PopulatePILCSecRun.\n";

            /* Prepare to select by network, borough or feeder */
            string str = "";
```

```
if (ddlFeeder.SelectedIndex > 0)

  str = " WHERE feeder='" + ddlFeeder.Items[ddlFeeder.SelectedIndex].Value.Trim() +
"'";



else if (ddlNetwork.SelectedIndex > 0)

  str = " WHERE nwk='" + ddlNetwork.Items[ddlNetwork.SelectedIndex].Value.Trim() +
"'";



else if (ddlBorough.SelectedIndex > 0)

  str = " WHERE csa='" + ddlBorough.Items[ddlBorough.SelectedIndex].Value.Trim() +
"'";



string sql = "SELECT feeder, run_id, length, max(load_percentage) as load_percentage,
min([Feeder PILC Rank]) as rank, " +

  "max(loadXrank) as loadXrank FROM Rerank_PILC_Load" + str +

  " GROUP BY run_id, feeder, length ORDER BY feeder";



SqlDataAdapter da = new SqlDataAdapter(sql, m_conn);

DataTable dt = new DataTable();
```

```
            da.Fill(dt);

            da = null;

            return dt;

        }

        private DataTable PopulateXLP()

        {

            m_Error = "ERROR at PopulateXLP.\n";



            string str = "";

            if (ddlFeeder.SelectedIndex > 0)

                str = "AND feeder='" + ddlFeeder.Items[ddlFeeder.SelectedIndex].Value.Trim() + "' ";

            else if (ddlNetwork.SelectedIndex > 0)

                str = "AND network='" + ddlNetwork.Items[ddlNetwork.SelectedIndex].Value.Trim() +
"' ";

            else if (ddlBorough.SelectedIndex > 0)

                str = "AND borough='" + ddlBorough.Items[ddlBorough.SelectedIndex].Value.Trim() +
"' ";
```

```
string sql = "SELECT feeder, run_id, min([Feeder PILC RANK]) as rank,
max(load_percentage) as load_percentage, " +
    "max(loadxrank) as loadxrank, 1 as length, borough, network, vision_id " +
    "FROM Precom_Cables_MgrModel WHERE insulation = 'XLP' AND (install_year
between 1970 and 1975) " + str +
    "GROUP BY feeder, run_id, length, borough, network, vision_id " +
    "ORDER BY feeder, run_id, vision_id";

SqlDataAdapter da = new SqlDataAdapter(sql, m_conn);
DataTable dt = new DataTable();
da.Fill(dt);
da = null;
return dt;
}
private DataTable PopulateALaCart()
{
/* Prepare to select by borough, network or feeder */
m_Error = "ERROR at PopulateALaCart.\n";
```

```
string str = "";

if (ddlRunID.SelectedIndex > 0)
  str = " WHERE run_id=" + ddlRunID.Items[ddlRunID.SelectedIndex].Value;
else if (ddlFeeder.SelectedIndex > 0)
  str = " WHERE feeder='" + ddlFeeder.Items[ddlFeeder.SelectedIndex].Value.Trim() +
"'";
else if (ddlNetwork.SelectedIndex > 0)
  str = " WHERE network='" + ddlNetwork.Items[ddlNetwork.SelectedIndex].Value.Trim() + "'";
else if (ddlBorough.SelectedIndex > 0)
  str = " WHERE borough='" + ddlBorough.Items[ddlBorough.SelectedIndex].Value.Trim() + "'";

string sql = "SELECT distinct feeder, run_id, length as length, network, borough FROM
Precom_Cables_MgrModel" + str + " ORDER BY feeder, run_id";

if (m_conn.State == ConnectionState.Closed)
  m_conn.Open();
```

```
    /* Fetch data */
    SqlDataAdapter da = new SqlDataAdapter(sql, m_conn);
    DataTable dt = new DataTable();
    da.Fill(dt);
    da = null;
    return dt;
}
private void PopulateStopJoints()
{
    m_Error = "ERROR at PopulateStopJoints.\n";

    string str = "";
    if (ddlFeeder.SelectedIndex > 0)
        str = "AND feeder='" + ddlFeeder.Items[ddlFeeder.SelectedIndex].Value.Trim() + "' ";
    else if (ddlNetwork.SelectedIndex > 0)
        str = "AND network='" + ddlNetwork.Items[ddlNetwork.SelectedIndex].Value.Trim() +
"' ";
```

```
else if (ddlBorough.SelectedIndex > 0)

  str = "AND borough='" + ddlBorough.Items[ddlBorough.SelectedIndex].Value.Trim() +
"' ";


string sql = "SELECT distinct run_id, max(length) as length, min([Feeder Joint Rank]) as
worst_joint_rank, count(distinct id) as stops, feeder " +

  "FROM Rerank_Runs_For_Joints " +

  "WHERE stop='s' AND insulation='paper' " + str +

  "GROUP BY run_id, feeder ORDER BY min([Feeder Joint Rank]) ASC";


SqlDataAdapter da = new SqlDataAdapter(sql, m_conn);

DataTable dt = new DataTable();

da.Fill(dt);

da = null;

SJdv = new DataView(dt);

dt = null;


//get joints
```

```
            sql = "SELECT distinct wei_id as joint_id, run_id, feeder FROM Rerank_Runs_For_Joints
WHERE insulation='paper' AND stop='S' " + str +
              " ORDER BY run_id";

            da = new SqlDataAdapter(sql, m_conn);
            dt = new DataTable();
            da.Fill(dt);
            da = null;
            SJdv_1 = new DataView(dt);
        }

        /// <summary>
        /// Replace a whole run for pilc and stop joints
        /// </summary>
        private void PILCSecRun()
        {
            /* Prepare string to not redo run_id done previously in load relief */
            string str = "";
```

```
if (m_replacedLoadRuns != null && m_replacedLoadRuns.Count > 0)
{
   for (int i = 0; i < m_replacedLoadRuns.Count; i++)
   {
      if (str == "")
         str = "AND run_id NOT IN (" + m_replacedLoadRuns[i].ToString();
      else
         str += "," + m_replacedLoadRuns[i].ToString();
   }

   str += ")";
}

/* ALL STRATEGIES */
if (ddlPilcStrat.SelectedIndex == 0)
{
   FeederModelMgr.Feeder oFeeder =
(FeederModelMgr.Feeder)m_fdrMgr.oFeeder.Clone(); //set check point
```

```
for (int s = 1; s < ddlPilcStrat.Items.Count; s++)
{
  m_actualPer = 0;
  m_runTot = m_loadRunTot;
  m_seqID = m_loadSeqID;
  m_pilcStrat = ddlPilcStrat.Items[s].Value;
  m_replacedPilcRuns = new ArrayList();
  m_pilcSeqID = m_pilcRunTot = m_pilcPer = 0;

  //reset check point for next tier 2 strategy
  if (s > 1)
    m_fdrMgr.oFeeder = (FeederModelMgr.Feeder)oFeeder.Clone();

  /* STOP JOINTS */
  if (m_pilcStrat == "stopJoints")
  {
    /* All percentages */
    if (chk5.Checked && ddlPilcPer5.SelectedIndex == 0)
```

```
        ReplaceStopJoints(str);


    else if (!chk5.Checked && ddlPilcPer.SelectedIndex == 0)
        ReplaceStopJoints(str);


    /* Selected percentage */
    else
        ReplaceStopJoints_selPer(str);
}


/* PILC */
else
{
    /* All percentages */
    if (chk5.Checked && ddlPilcPer5.SelectedIndex == 0)
        ReplacePILCSecRun(str);


    else if (!chk5.Checked && ddlPilcPer.SelectedIndex == 0)
```

```
            ReplacePILCSecRun(str);


          /* Selected percentage */

          else

            ReplacePILCSecRun_selPer(str);

        }


        ///* Save what we just did into db */

        //SaveBulkData();


      } //end for


      m_replacedPilcRuns = null;

      m_pilcSeqID = m_pilcRunTot = m_pilcPer = 0;

      m_pilcStrat = "";

      return;

    }
```

```
m_actualPer = 0;

m_runTot = m_loadRunTot;

m_seqID = m_loadSeqID;

m_pilcStrat = ddlPilcStrat.Items[ddlPilcStrat.SelectedIndex].Value;

m_replacedPilcRuns = new ArrayList();

m_pilcSeqID = m_pilcRunTot = m_pilcPer = 0;



/* STOP JOINTS */

if (m_pilcStrat == "stopJoints")

{

   /* All percentages */

   if (chk5.Checked && ddlPilcPer5.SelectedIndex == 0)

      ReplaceStopJoints(str);

   else if (!chk5.Checked && ddlPilcPer.SelectedIndex == 0)

      ReplaceStopJoints(str);



   /* Selected percentage */

   else
```

```
        ReplaceStopJoints_selPer(str);
    }

    /* PILC */
    else
    {
      /* All percentages */
      if (chk5.Checked && ddlPilcPer5.SelectedIndex == 0)
        ReplacePILCSecRun(str);

      else if (!chk5.Checked && ddlPilcPer.SelectedIndex == 0)
        ReplacePILCSecRun(str);

      /* Selected percentage */
      else
        ReplacePILCSecRun_selPer(str);
    }
```

```
        /* Clean up */
        m_replacedPilcRuns = null;
        m_pilcSeqID = m_pilcRunTot = m_pilcPer = 0;
        m_pilcStrat = "";
    }
    private void RebuildHipotBase()
    {
        string boro = "";
        string network = "";
        string feeder = "";
        DataView dv = new DataView(GetFdrNwkBoro(ref boro, ref network, ref feeder));    //get
feeders for looping
        m_fdrMgr = new FeederModelMgr.FdrModMgr(boro, network, feeder);

        for (int i = 0; i < dv.Count; i++)
        {
            feeder = dv[i].Row["feeder"].ToString().Trim();
            if (m_fdrMgr.ChangeFeeder(feeder))  //skip feeder with no cables or joints
```

```
                continue;


                Hashtable ht = m_fdrMgr.CountMe();

                SaveHipotBase(ht);

            }


            m_Error = "Requested Transaction is completed.\nPlease check table
PRECOM_HipotBase for results.";

            ErrorMessage(m_Error);

        }

        private void RelSecRun()

        {

            //prepare string to store run_id already done in tier 1 and tier 2

            string str = "";

            string str1 = "";

            if (m_replacedLoadRuns != null && m_replacedLoadRuns.Count > 0)

            {

                for (int i = 0; i < m_replacedLoadRuns.Count; i++)
```

```
    {
        if (str == "")
            str = "AND run_id NOT IN (" + Convert.ToInt32(m_replacedLoadRuns[i]);
        else
            str += "," + Convert.ToInt32(m_replacedLoadRuns[i]);
    }
}

if (m_replacedPilcRuns != null && m_replacedPilcRuns.Count > 0)
{
    for (int i = 0; i < m_replacedPilcRuns.Count; i++)
    {
        if (str == "")
            str = "AND run_id NOT IN (" + Convert.ToInt32(m_replacedPilcRuns[i]);
        else
            str += "," + Convert.ToInt32(m_replacedPilcRuns[i]);
    }
}
```

```
if (str != "")
   str += ")";

str1 = str;
if (m_replacedLoadSections != null && m_replacedLoadSections.Count > 0)
{
   for (int i = 0; i < m_replacedLoadSections.Count; i++)
   {
      if (i == 0)
         str += " AND vision_id NOT IN (" +
Convert.ToInt32(m_replacedLoadSections[i]);
      else
         str += "," + Convert.ToInt32(m_replacedLoadSections[i]);
   }
}

if (str != "" && str[str.Length - 1] != ')')
```

```
            str += ")";



            /* ALL STRATEGIES */

            if (ddlRelStrat.SelectedIndex == 0)

            {

              FeederModelMgr.Feeder oFeeder =
(FeederModelMgr.Feeder)m_fdrMgr.oFeeder.Clone();

              for (int s = 1; s < ddlRelStrat.Items.Count; s++)

              {

                m_actualPer = 0;

                m_seqID = m_pilcSeqID;

                m_runTot = m_pilcRunTot;

                m_relStrat = ddlRelStrat.Items[s].Value;



                //reset check point for next tier 3 strategy

                if (s > 1)

                  m_fdrMgr.oFeeder = (FeederModelMgr.Feeder)oFeeder.Clone();
```

```
/* STOP JOINTS */
if (m_relStrat == "rel_stopJoints")
{
   /* All percentages */
   if (ddlRelPer.SelectedIndex == 0)
      ReplaceRelStopJoints(str1);

   /* Selected percentage */
   else
      ReplaceRelStopJoints_selPer(str1);
}

/* XLP */
else
{
   /* All percentages */
   if (ddlRelPer.SelectedIndex == 0)
      ReplaceXLP(str);
```

```
            /* Selected percentage */
            else
               ReplaceXLP_selPer(str);
         }

         /* Save what we just did into db */
         SaveBulkData();

      } //end for

      m_relPer = 0;
      m_relStrat = "";
      return;

   } //end if

   m_actualPer = 0;
```

```
m_seqID = m_pilcSeqID;

m_runTot = m_pilcRunTot;

m_relStrat = ddlRelStrat.Items[ddlRelStrat.SelectedIndex].Value;



/* STOP JOINTS */

if (m_relStrat == "rel_stopJoints")

{

   /* All percentages */

   if (ddlRelPer.SelectedIndex == 0)

      ReplaceRelStopJoints(str1);



   /* Selected percentage */

   else

      ReplaceRelStopJoints_selPer(str1);

}



/* XLP */

else
```

```
    {
      /* All percentages */
      if (ddlRelPer.SelectedIndex == 0)
        ReplaceXLP(str);

      /* Selected percentage */
      else
        ReplaceXLP_selPer(str);
    }

    //clear data for next pilc strategy
    m_relPer = 0;
    m_relStrat = "";
  }

  private void ReplacePILCSecRun(string str)
  {
```

```
PILCdv.RowFilter = "feeder='" + m_feeder + "' " + str;  //exclude run_id done previously
in tier 1

/* NO PILC */
Hashtable ht = null;
if ((PILCdv.Count == 0) || (PILCdv.Count == 1 &&
PILCdv[0].Row["run_id"].Equals(DBNull.Value)))
{
  int count = ddlPilcPer.Items.Count;
  if (chk5.Checked)
    count = ddlPilcPer5.Items.Count;

  for (int i = 1; i < count; i++)
  {
    m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[i].Value);
    if (chk5.Checked)
      m_pilcPer = Convert.ToInt32(ddlPilcPer5.Items[i].Value);
```

```
                m_Error = "ERROR at ReplacePILCRSecRun.\nFeeder: " + m_feeder + "\nLoad
Relief " + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\n";

                if (m_seqID == 0)

                   ht = m_fdrMgr.CountMe();



                PrepRerankSql(ht);

                ht = null;



                /* Compute 3rd tier */

                if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)

                   ReplaceRel();

             }



             return;

          }



          /* REPLACE PILC */

          m_pilcPer = 0;
```

```
int cnt_p = 0;

int running_total = 0;

bool firstTime = true;

float running_percent;

float sumLen = (float)Convert.ToDouble(PILCdv.Table.Compute("Sum(length)",
"feeder='" + m_feeder + "' " + str));

ArrayList runIDLs = new ArrayList();



/* Sort data */

if (m_pilcStrat == "rank")

   PILCdv.Sort = "rank asc";

else if (m_pilcStrat == "load")

   PILCdv.Sort = "load_percentage desc";

else if (m_pilcStrat == "loadxrank")

   PILCdv.Sort = "loadxrank desc";



/* Retrieve data where percentage <= stop percent */

for (int i = 0; i < PILCdv.Count; i++)
```

```
{
  //skip null run_id
  if (PILCdv[i].Row["run_id"].Equals(DBNull.Value))
    continue;

  running_total += Convert.ToInt32(PILCdv[i].Row["length"]);
  running_percent = running_total / sumLen * 100;

  if (firstTime)
  {
    /* Get next stop percent */
    while (running_percent > m_pilcPer)
    {
      cnt_p++;
      m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[cnt_p].Value);
      if (chk5.Checked)
        m_pilcPer = Convert.ToInt32(ddlPilcPer5.Items[cnt_p].Value);
```

```
        if (running_percent > m_pilcPer)
        {
            if (m_seqID == 0)
                ht = m_fdrMgr.CountMe();

            PrepRerankSql(ht);
            ht = null;

            /* Compute 3rd tier */
            if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
                ReplaceRel();
        }
    }

    firstTime = false;
}
```

```
m_Error = "ERROR at ReplacePILCSecRun.\nFeeder: " + m_feeder + "%\nLoad Relief
" + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\n";


int run_id = Convert.ToInt32(PILCdv[i].Row["run_id"]);
if (running_percent <= m_pilcPer)
{
  runIDLs.Add(run_id);
  m_replacedPilcRuns.Add(run_id);
  m_actualPer = (float)Math.Round(running_percent, 5);
}


//save data for rerank at this stop percent
if (running_percent >= m_pilcPer)
{
  //add load relief running total if load relief was done
  if (running_percent > m_pilcPer)
    m_runTot = running_total + m_loadRunTot -
Convert.ToInt32(PILCdv[i].Row["length"]);
```

```
else
    m_runTot = running_total + m_loadRunTot;

//save data for rerank
ht = m_fdrMgr.ReplaceMe(runIDLs, false);
PrepRerankSql(ht);

//clear for next percentage
runIDLs.Clear();
ht = null;

/* Compute 3rd tier */
if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
    ReplaceRel();

/* Get next stop percent */
if (running_percent == m_pilcPer)
    firstTime = true;
```

```
else
{
  while (running_percent > m_pilcPer)
  {
    cnt_p++;
    m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[cnt_p].Value);
    if (chk5.Checked)
      m_pilcPer =  Convert.ToInt32(ddlPilcPer5.Items[cnt_p].Value);

    if (running_percent > m_pilcPer)
    {
      PrepRerankSql(ht);

      /* Compute 3rd tier */
      if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
        ReplaceRel();
    }
    else
```

```
            {
               runIDLs.Add(run_id);
               m_replacedPilcRuns.Add(run_id);
               m_actualPer = (float)Math.Round(running_percent, 5);
            }
         }
      }

   } /* end if */

} /* end for */

/* Run the last batch */
if (runIDLs.Count > 0)
{
   m_runTot = running_total + m_loadRunTot;
   ht = m_fdrMgr.ReplaceMe(runIDLs,false);
   PrepRerankSql(ht);
```

```
        /* Compute 3rd tier */
        if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
        {
          m_pilcRunTot = m_runTot;
          m_pilcSeqID = m_seqID;
          RelSecRun();
        }
      }

      /* MONOtonic check */
      //MonoCheck();
    }
    private void ReplaceRelStopJoints(string str)
    {
      /* Exclude run_id for tier 1 and 2 done previously */
      SJdv.RowFilter = "feeder='" + m_feeder + "' " + str;
```

```
/* NO STOP JOINTS */

Hashtable ht = null;

if (SJdv.Count == 0)

{

   for (int i = 1; i < ddlRelPer.Items.Count; i++)

   {

      m_relPer = Convert.ToInt32(ddlRelPer.Items[i].Value);

      m_Error = "ERROR at ReplaceRelStopJoints().\nFeeder: " + m_feeder + "\nLoad
Relief " + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer +

         "%\nReliablity " + m_relStrat + " " + m_relPer + "%\n";



      if (m_seqID == 0)

         ht = m_fdrMgr.CountMe();



      PrepRerankSql(ht);

      ht = null;

   }
```

```
    return;
}

/* REPLACE STOP JOINTS */
int cnt_p = 0;
int runningTot = 0;
int total_section = 0;
float sumStops = (float)Convert.ToDouble(SJdv.Table.Compute("Sum(stops)", "feeder='"
+ m_feeder + "' " + str));
bool firstTime = true;
float running_percent = 0;
ArrayList dupLs = new ArrayList();
ArrayList runIDLs = new ArrayList();

/* Sort */
SJdv.Sort = "worst_joint_rank asc";

/* Retrieve data where percentage <= stop percent */
```

```
for (int i = 0; i < SJdv.Count; i++)
{
  /* Check for duplicate joints */
  int run_id = Convert.ToInt32(SJdv[i].Row["run_id"]);
  int dupCnt = 0;
  SJdv_1.RowFilter = "feeder='" + m_feeder + "' AND run_id=" + run_id;

  for (int j = 0; j < SJdv_1.Count; j++)
  {
    int joint = Convert.ToInt32(SJdv_1[j]["joint_id"]);
    if (dupLs.Contains(joint))
      dupCnt += 1;
    else
      dupLs.Add(joint);
  }

  /* Calculate percentage */
  sumStops -= dupCnt;
```

```
runningTot += Convert.ToInt32(SJdv[i].Row["stops"]) - dupCnt;

running_percent = runningTot / sumStops * 100;



if (firstTime)

{

  /* Get next stop percent */

  while (running_percent > m_relPer)

  {

    cnt_p++;

    m_relPer = Convert.ToInt32(ddlRelPer.Items[cnt_p].Value);



    if (running_percent > m_relPer)

    {

      if (m_seqID == 0)

        ht = m_fdrMgr.CountMe();



      PrepRerankSql(ht);

      ht = null;
```

```
            }
        }

        firstTime = false;
    }

    m_Error = "ERROR at ReplaceStopJoints().\nFeeder: " + m_feeder + "\nLoad Relief " +
m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer +
        "%\nReliability " + m_relStrat + " " + m_relPer + "%\n";

    if (running_percent <= m_relPer)
    {
        runIDLs.Add(run_id);
        m_actualPer = (float)Math.Round(running_percent, 5);
        total_section += Convert.ToInt32(SJdv[i].Row["length"]);
    }

    /* Send data for rerank at this stop percent */
```

```
if (running_percent >= m_relPer)
{
  m_runTot = total_section + m_pilcRunTot;     //add load relief running total if load
relief was done
  ht = m_fdrMgr.ReplaceMe(runIDLs, false);
  PrepRerankSql(ht);

  runIDLs.Clear();
  ht = null;

  /* Get next stop percent */
  if (running_percent == m_relPer)
    firstTime = true;
  else
  {
    while (running_percent > m_relPer)
    {
      cnt_p++;
```

```
            m_relPer = Convert.ToInt32(ddlRelPer.Items[cnt_p].Value);



            if (running_percent > m_relPer)

               PrepRerankSql(ht);

            else

            {

               runIDLs.Add(run_id);

               m_actualPer = (float)Math.Round(running_percent, 5);

               total_section += Convert.ToInt32(SJdv[i].Row["length"]);

            }

         }

      }



   } /* end if */



} /* end for */



/* Run the last batch */
```

```
            if (runIDLs.Count > 0)
            {
                m_runTot = total_section + m_pilcRunTot;
                ht = m_fdrMgr.ReplaceMe(runIDLs, false);
                PrepRerankSql(ht);
            }

            /* MONOtonic check */
            //MonoCheck();
        }
        private void ReplaceRelStopJoints_selPer(string str)
        {
            m_relPer = Convert.ToInt32(ddlRelPer.Items[ddlRelPer.SelectedIndex].Value);
            m_Error = "ERROR at ReplaceRelStopJoints_selPer.\nFeeder: " + m_feeder + "\nLoad
Relief " + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer +
                "%\nReliability " + m_relStrat + " " + m_relPer + "%\n";

            //exclude run_id for tier 1 and 2 done previously
```

```
SJdv.RowFilter = "feeder='" + m_feeder + "' " + str;



/* NO STOP JOINTS */

Hashtable ht = null;

if (SJdv.Count == 0)

{

   if (m_seqID == 0)

      ht = m_fdrMgr.CountMe();



   PrepRerankSql(ht);

   return;

}



/* REPLACE STOP JOINTS */

int runningTot = 0;

int total_section = 0;

float sumStops = (float)Convert.ToDouble(SJdv.Table.Compute("Sum(stops)", "feeder='"
+ m_feeder + "' " + str));
```

```
float running_percent = 0;

ArrayList dupLs = new ArrayList();

ArrayList runIDLs = new ArrayList();



/* Sort */

SJdv.Sort = "worst_joint_rank asc";



/* Retrieve data where percentage <= stop percent */

for (int i = 0; i < SJdv.Count; i++)

{

  /* Check for duplicate joints */

  int run_id = Convert.ToInt32(SJdv[i]["run_id"]);

  int dupCnt = 0;

  SJdv_1.RowFilter = "feeder='" + m_feeder + "' AND run_id=" + run_id;



  for (int j = 0; j < SJdv_1.Count; j++)

  {

    int joint = Convert.ToInt32(SJdv_1[j]["joint_id"]);
```

```
    if (dupLs.Contains(joint))
        dupCnt += 1;
    else
        dupLs.Add(joint);
}

/* Calculate percentage */
sumStops -= dupCnt;
runningTot += Convert.ToInt32(SJdv[i].Row["stops"]) - dupCnt;
running_percent = runningTot / sumStops * 100;

if (running_percent <= m_relPer)
{
    runIDLs.Add(run_id);
    m_actualPer = (float)Math.Round(running_percent, 5);
    total_section += Convert.ToInt32(SJdv[i].Row["length"]);
}
else
```

```
            break;


        } //end for i


        //no run_id for this percentage
        if ((runIDLs.Count == 0) && (m_seqID == 0))
            ht = m_fdrMgr.CountMe();
        else
        {
            m_runTot = total_section + m_pilcRunTot;    //add previous tiers running total
            ht = m_fdrMgr.ReplaceMe(runIDLs, false);
        }


        PrepRerankSql(ht);
    }
    /// <summary>
    /// Replace by section for XLP
    /// </summary>
```

```
private void ReplaceXLP(string str)

{

  XLPdv.RowFilter = "feeder='" + m_feeder + "' " + str;   //exclude run_id done previously
in tier 1 and 2



  /* NO XLP */

  Hashtable ht = null;

  if (XLPdv.Count == 0)

  {

    for (int i = 1; i < ddlRelPer.Items.Count; i++)

    {

      m_relPer = Convert.ToInt32(ddlRelPer.Items[i].Value);

      m_Error = "ERROR at ReplacePILCRSecRun.\nFeeder: " + m_feeder + "\nLoad
Relief " + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer +

        "%\nReliability " + m_relStrat + " " + m_relPer  + "%\n";



      if (m_seqID == 0)

        ht = m_fdrMgr.CountMe();
```

```
            PrepRerankSql(ht);

            ht = null;


        }


        return;

    }


    /* REPLACE XLP */

    m_Error = "ERROR at ReplacePILCSecRun.\nFeeder: " + m_feeder + "%\nLoad Relief "
+ m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer +

        "%Reliability " + m_relStrat + " " + m_relPer + "%\n";


    m_relPer = 0;

    int cnt_p = 0;

    int running_total = 0;

    bool firstTime = true;
```

```
float running_percent;

float sumLen = (float)Convert.ToDouble(XLPdv.Table.Compute("Sum(length)",
"feeder='" + m_feeder + "' " + str));

ArrayList visionIDLs = new ArrayList();



/* Sort data */

if (m_relStrat == "rel_rank")

   XLPdv.Sort = "rank asc";

else if (m_relStrat == "rel_load")

   XLPdv.Sort = "load_percentage desc";

else if (m_relStrat == "rel_loadxrank")

   XLPdv.Sort = "loadxrank desc";



/* Retrieve data where percentage <= stop percent */

for (int i = 0; i < XLPdv.Count; i++)

{

   running_total += Convert.ToInt32(XLPdv[i].Row["length"]);

   running_percent = running_total / sumLen * 100;
```

```
if (firstTime)

{

  /* Get next stop percent */

  while (running_percent > m_relPer)

  {

    cnt_p++;

    m_relPer = Convert.ToInt32(ddlRelPer.Items[cnt_p].Value);



    if (running_percent > m_relPer)

    {

      if (m_seqID == 0)

        ht = m_fdrMgr.CountMe();



      PrepRerankSql(ht);

      ht = null;

    }

  }
```

```
                firstTime = false;

            }


            m_Error = "ERROR at ReplacePILCSecRun.\nFeeder: " + m_feeder + "%\nLoad Relief
" + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer +

                "%Reliability " + m_relStrat + " " + m_relPer + "%\n";



            int vision_id = Convert.ToInt32(XLPdv[i].Row["vision_id"]);

            if (running_percent <= m_relPer)

            {

                visionIDLs.Add(vision_id);

                m_actualPer = (float)Math.Round(running_percent, 5);

            }



            /* Send data for rerank at this stop percent */

            if (running_percent >= m_relPer)

            {
```

```
/* Add load relief and pilc running total if were done before.  m_pilcRunTot already
included the load relief total */
if (running_percent > m_relPer)
   m_runTot = running_total - Convert.ToInt32(XLPdv[i].Row["length"]) +
m_pilcRunTot;
else
   m_runTot = running_total + m_pilcRunTot;

//save data for rerank
ht = m_fdrMgr.ReplaceMe(visionIDLs, true);
PrepRerankSql(ht);

//clear for next percentage
visionIDLs.Clear();
ht = null;

/* Get next stop percent */
if (running_percent == m_relPer)
```

```
    firstTime = true;
  else
  {
    while (running_percent > m_relPer)
    {
      cnt_p++;
      m_relPer = Convert.ToInt32(ddlRelPer.Items[cnt_p].Value);

      if (running_percent > m_relPer)
        PrepRerankSql(ht);
      else
      {
        visionIDLs.Add(vision_id);
        m_actualPer = (float)Math.Round(running_percent, 5);
        break;
      }
    }
  }
```

```
    } /* end if */


  } //end for


  /* Run the last batch */
  if (visionIDLs.Count > 0)
  {
    m_runTot = running_total + m_pilcRunTot;
    ht = m_fdrMgr.ReplaceMe(visionIDLs, true);
    PrepRerankSql(ht);
  }


  /* MONOtonic check */
  //MonoCheck();
}
private void ReplaceXLP_selPer(string str)
{
```

```
m_relPer = Convert.ToInt32(ddlRelPer.Items[ddlRelPer.SelectedIndex].Value);

m_Error = "ERROR at ReplaceXLP_selPer().\nFeeder: " + m_feeder + "\nLoad Relief " +
m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\nReliability " + m_relStrat + " " + m_relPer +
"%\n";

XLPdv.RowFilter = "feeder='" + m_feeder + "' " + str;   //exclude run_id done previously
in tier 1 and 2



/* NO XLP */

Hashtable ht = null;

if (XLPdv.Count == 0)

{

  if (m_seqID == 0)

    ht = m_fdrMgr.CountMe();



  PrepRerankSql(ht);

  return;

}



/* REPLACE XLP */
```

```
int runningTot = 0;

float running_percent;

float sumLen = (float)Convert.ToDouble(XLPdv.Table.Compute("Sum(length)",
"feeder='" + m_feeder + "' " + str));

ArrayList visionIDLs = new ArrayList();



/* Sort data */

if (m_relStrat == "rel_rank")

  XLPdv.Sort = "rank asc";

else if (m_relStrat == "rel_load")

  XLPdv.Sort = "load_percentage desc";

else if (m_relStrat == "rel_loadxrank")

  XLPdv.Sort = "loadxrank desc";



/* Compute data */

for (int i = 0; i < XLPdv.Count; i++)

{

  runningTot += Convert.ToInt32(XLPdv[i].Row["length"]);
```

```
running_percent = runningTot / sumLen * 100;

if (running_percent <= m_relPer)
{
    m_actualPer = (float)Math.Round(running_percent, 5);
    visionIDLs.Add(Convert.ToInt32(XLPdv[i].Row["vision_id"]));
}
else
{
    if (m_relPer != 100)
        runningTot -= Convert.ToInt32(XLPdv[i].Row["length"]);

    break;
}
}

//no run_id for this percentage
if ((visionIDLs.Count == 0) && (m_seqID == 0))
```

```
            ht = m_fdrMgr.CountMe();
          else
          {
            m_runTot = runningTot + m_pilcRunTot;
            ht = m_fdrMgr.ReplaceMe(visionIDLs, true);
          }

          PrepRerankSql(ht);
        }
        private void ReplacePILCSecRun_selPer(string str)
        {
          m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[ddlPilcPer.SelectedIndex].Value);
          if (chk5.Checked)
            m_pilcPer = Convert.ToInt32(ddlPilcPer5.Items[ddlPilcPer5.SelectedIndex].Value);

          m_Error = "ERROR at ReplacePILCRSecRun_selPer.\nFeeder: " + m_feeder + "\nLoad
Relief " + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\n";
```

```
PILCdv.RowFilter = "feeder='" + m_feeder + "' " + str;  //exclude run_id done previously
in tier 1

/* NO PILC */
Hashtable ht = null;
if ((PILCdv.Count == 0) || (PILCdv.Count == 1 &&
PILCdv[0].Row["run_id"].Equals(DBNull.Value)))
{
  if (m_seqID == 0)
    ht = m_fdrMgr.CountMe();

  PrepRerankSql(ht);

  /* Compute 3rd tier */
  if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
  {
    m_pilcRunTot = m_runTot;
    m_pilcSeqID  = m_seqID;
```

```
            RelSecRun();
        }

        return;
    }

    /* REPLACE PILC */
    int runningTot = 0;
    float running_percent, sumLen;
    sumLen = (float)Convert.ToDouble(PILCdv.Table.Compute("Sum(length)", "feeder='" +
m_feeder + "' " + str));

    /* Sort data */
    if (m_pilcStrat == "rank")
        PILCdv.Sort = "rank asc";
    else if (m_pilcStrat == "load")
        PILCdv.Sort = "load_percentage desc";
```

```
else if (m_pilcStrat == "loadxrank")
   PILCdv.Sort = "loadxrank desc";

/* Compute data */
for (int i = 0; i < PILCdv.Count; i++)
{
   //skip null run_id
   if (PILCdv[i].Row["run_id"].Equals(DBNull.Value))
      continue;

   runningTot += Convert.ToInt32(PILCdv[i].Row["length"]);
   running_percent = runningTot / sumLen * 100;

   if (running_percent <= m_pilcPer)
   {
      m_actualPer = (float)Math.Round(running_percent, 5);
      m_replacedPilcRuns.Add(Convert.ToInt32(PILCdv[i].Row["run_id"]));
   }
```

```
    else
    {
      if (m_pilcPer != 100)
        runningTot -= Convert.ToInt32(PILCdv[i].Row["length"]);

      break;
    }
  }

  //no run_id for this percentage
  if ((m_replacedPilcRuns.Count == 0) && (m_seqID == 0))
    ht = m_fdrMgr.CountMe();
  else
  {
    m_runTot = runningTot + m_loadRunTot;    // Add load relief running total if load relief
was done
    ht = m_fdrMgr.ReplaceMe(m_replacedPilcRuns, false);
  }
```

```
    PrepRerankSql(ht);


    /* Compute 3rd tier */
    if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
    {
      m_pilcRunTot = m_runTot;
      m_pilcSeqID  = m_seqID;
      RelSecRun();
    }
}
private void ReplaceStopJoints(string str)
{
    //if Load Relief done previously, exclude the load relief run_id
    SJdv.RowFilter = "feeder='" + m_feeder + "' " + str;
    SJdv_1.RowFilter = "feeder='" + m_feeder + "'";


    //copy this filter in case 3rd tier stop joints being done after this
```

```
DataView dv = new DataView(SJdv.ToTable());

DataView dv_1 = new DataView(SJdv_1.ToTable());



/* NO STOP JOINTS */

float actual_per = 0;

Hashtable ht = null;

if (dv.Count == 0)

{

  for (int i = 1; i < ddlPilcPer.Items.Count; i++)

  {

    m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[i].Value);

    m_Error = "ERROR at ReplaceStopJoints().\nFeeder: " + m_feeder + "\nLoad Relief
" + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\n";



    if (m_seqID == 0)

      ht = m_fdrMgr.CountMe();



    PrepRerankSql(ht);
```

```
        ht = null;



        /* Compute 3rd tier */

        if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)

            ReplaceRel();

    }



    return;

}




/* REPLACE STOP JOINTS */

int cnt_p = 0;

int runningTot = 0;

int total_section = 0;

int sumStops = Convert.ToInt32(dv.Table.Compute("Sum(stops)", ""));

bool firstTime = true;

float running_percent = 0;
```

```
ArrayList runIDLs = new ArrayList();

ArrayList dupLs = new ArrayList();



/* Sort */

dv.Sort = "worst_joint_rank asc";



/* Retrieve data where percentage <= stop percent */

for (int i = 0; i < dv.Count; i++)

{

   /* Check for duplicate joints */

   int run_id = Convert.ToInt32(dv[i].Row["run_id"]);

   int dupCnt = 0;

   dv_1.RowFilter = "run_id=" + run_id;



   for (int j = 0; j < dv_1.Count; j++)

   {

      int joint = Convert.ToInt32(dv_1[j]["joint_id"]);

      if (dupLs.Contains(joint))
```

```
        dupCnt += 1;
    else
        dupLs.Add(joint);
}

/* Calculate percentage */
sumStops -= dupCnt;
runningTot += Convert.ToInt32(dv[i].Row["stops"]) - dupCnt;
running_percent = runningTot / (float)sumStops * 100;

if (firstTime)
{
    /* Get next stop percent */
    while (running_percent > m_pilcPer)
    {
        cnt_p++;
        m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[cnt_p].Value);
```

```
        if (running_percent > m_pilcPer)
        {
          if (m_seqID == 0)
            ht = m_fdrMgr.CountMe();

          PrepRerankSql(ht);
          ht = null;

          /* Compute 3rd tier */
          if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
            ReplaceRel();
        }
      }

      firstTime = false;
    }
```

```
m_Error = "ERROR at ReplaceStopJoints().\nFeeder: " + m_feeder + "\nLoad Relief " +
m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\n";

if (running_percent <= m_pilcPer)
{
  runIDLs.Add(run_id);
  m_replacedPilcRuns.Add(run_id);
  m_actualPer = actual_per = (float)Math.Round(running_percent, 5);
  total_section += Convert.ToInt32(dv[i].Row["length"]);
}

/* Send data for rerank at this stop percent */
if (running_percent >= m_pilcPer)
{

  m_runTot = total_section + m_loadRunTot;    //add load relief running total if load
relief was done
  ht = m_fdrMgr.ReplaceMe(runIDLs, false);
```

```
PrepRerankSql(ht);



runIDLs.Clear();

ht = null;



/* Compute 3rd tier */

if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)

   ReplaceRel();



/* Get next stop percent */

if (running_percent == m_pilcPer)

   firstTime = true;

else

{

   while (running_percent > m_pilcPer)

   {

      cnt_p++;

      m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[cnt_p].Value);
```

```
        if (running_percent > m_pilcPer)
        {
            PrepRerankSql(ht);

            /* Compute 3rd tier */
            if ((chkRel.Checked) && (chkLoadRelief.Checked) && (chkPilc.Checked))
                ReplaceRel();
        }
        else
        {
            runIDLs.Add(run_id);
            m_replacedPilcRuns.Add(run_id);
            m_actualPer = actual_per = (float)Math.Round(running_percent, 5);
            total_section += Convert.ToInt32(dv[i].Row["length"]);
        }
    }
}
```

```
    } /* end if */


} /* end for */


/* Run the last batch */
if (runIDLs.Count > 0)
{
    m_runTot = total_section + m_loadRunTot;
    ht = m_fdrMgr.ReplaceMe(runIDLs, false);
    PrepRerankSql(ht);


    /* Compute 3rd tier */
    if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
    {
        m_pilcRunTot = m_runTot;
        m_pilcSeqID = m_seqID;
        RelSecRun();
```

```
            }
        }
    }
    private void ReplaceStopJoints_selPer(string str)
    {
        m_pilcPer = Convert.ToInt32(ddlPilcPer.Items[ddlPilcPer.SelectedIndex].Value);
        m_Error = "ERROR at ReplaceStopJoints_selPer.\nFeeder: " + m_feeder + "\nLoad Relief
" + m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\n";

        //if Load Relief done previously, exclude the load relief run_id
        SJdv.RowFilter = "feeder='" + m_feeder + "' " + str;
        SJdv_1.RowFilter = "feeder='" + m_feeder + "'";

        //copy this filter in case 3rd tier stop joints being done after this
        DataView dv = new DataView(SJdv.ToTable());
        DataView dv_1 = new DataView(SJdv_1.ToTable());

        /* NO STOP JOINTS */
```

```
Hashtable ht = null;
if ((dv.Count == 0) || (dv.Count == 1 && dv[0].Row["run_id"].Equals(DBNull.Value)))
{
  if (m_seqID == 0)
    ht = m_fdrMgr.CountMe();

  PrepRerankSql(ht);

  /* Replace 3rd tier */
  if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
  {
    m_pilcRunTot = m_runTot;
    m_pilcSeqID = m_seqID;
    RelSecRun();
  }

  return;
}
```

```
/* REPLACE STOP JOINTS */

int runningTot = 0;

int total_section = 0;

float sumStops = (float)Convert.ToDouble(dv.Table.Compute("Sum(stops)", ""));

float running_percent = 0;

ArrayList dupLs = new ArrayList();



/* Sort */

dv.Sort = "worst_joint_rank asc";



/* Retrieve data where percentage <= stop percent */

for (int i = 0; i < dv.Count; i++)

{

   /* Check for duplicate joints */

   int run_id = Convert.ToInt32(dv[i]["run_id"]);

   int dupCnt = 0;

   dv_1.RowFilter = "run_id=" + run_id;
```

```
for (int j = 0; j < dv_1.Count; j++)
{
   int joint = Convert.ToInt32(dv_1[j]["joint_id"]);
   if (dupLs.Contains(joint))
      dupCnt += 1;
   else
      dupLs.Add(joint);
}

/* Calculate percentage */
sumStops -= dupCnt;
runningTot += Convert.ToInt32(dv[i].Row["stops"]) - dupCnt;
running_percent = runningTot / (float)sumStops * 100;

if (running_percent <= m_pilcPer)
{
   m_replacedPilcRuns.Add(run_id);
```

```
        m_actualPer = (float)Math.Round(running_percent, 5);

        total_section += Convert.ToInt32(dv[i].Row["length"]);

      }

      else

        break;



} //end for i



//no run_id for this percentage

if ((m_replacedPilcRuns.Count == 0) && (m_seqID == 0))

    ht = m_fdrMgr.CountMe();



//replace runs for this percentage

else if (m_replacedPilcRuns.Count > 0)

{

   m_runTot = total_section + m_loadRunTot;    //add previous tiers running total

   ht = m_fdrMgr.ReplaceMe(m_replacedPilcRuns, false);

}
```

```
    PrepRerankSql(ht);


    /* Compute 3rd tier */
    if (chkLoadRelief.Checked && chkPilc.Checked && chkRel.Checked)
    {
      m_pilcRunTot = m_runTot;
      m_pilcSeqID = m_seqID;
      RelSecRun();
    }
}
/// <summary>
/// Replace by run if insulation is paper else replace by section if other types
/// </summary>
/// <param name="loadPer"></param>
private void ReplaceLoadRelief(int loadPer)
{
    m_Error = "ERROR at ReplaceLoadRelief for Feeder: " + m_feeder + " " + loadPer + "%";
```

```
m_actualPer = m_runTot = m_seqID = 0;    //init these just in case previous strategy has been done

m_loadSeqID = m_loadRunTot = 0;

m_loadPer = loadPer;



m_replacedLoadRuns = new ArrayList();

m_replacedLoadSections = new ArrayList();



/* Filter and sort */

if (loadPer == 100)

   LRdv.RowFilter = "feeder='" + m_feeder + "' AND percentage >= 100";

else if (loadPer == 105)

   LRdv.RowFilter = "feeder='" + m_feeder + "' AND percentage >= 105";

else

   LRdv.RowFilter = "feeder='" + m_feeder + "' AND percentage >= 110";



/* NO load relief */
```

```
Hashtable ht = null;
if (LRdv.Count == 0)
{
   ht = m_fdrMgr.CountMe();
   PrepRerankSql(ht);
   return;
}

/* HAS load relief */
else
{
   m_actualPer = loadPer;
   for (int i = 0; i < LRdv.Count; i++)
   {
      //replace by run if insulation is paper else replace by section if other types
      int run_id = Convert.ToInt32(LRdv[i].Row["run_id"]);
      if (LRdv[i].Row["insulation"].ToString().ToLower() == "paper")
      {
```

```
        if (!m_replacedLoadRuns.Contains(run_id))
            m_replacedLoadRuns.Add(run_id);
    }
    else
    {
        int vision_id = Convert.ToInt32(LRdv[i].Row["vision_id"]);
        if (!m_replacedLoadSections.Contains(vision_id))
            m_replacedLoadSections.Add(vision_id);
    }

    m_loadRunTot += Convert.ToInt32(LRdv[i].Row["length"]);
}

m_runTot = m_loadRunTot;

if (m_replacedLoadRuns.Count > 0)
    ht = m_fdrMgr.ReplaceMe(m_replacedLoadRuns, false);   //ignore this count
if (m_replacedLoadSections.Count > 0)
```

```
                ht = m_fdrMgr.ReplaceMe(m_replacedLoadSections, true);        //this count includes
both sections and runs


                PrepRerankSql(ht);

            }


            m_loadSeqID = m_seqID;

        }

        private void ReplaceRel()

        {

            float actual_per = m_actualPer;

            m_pilcSeqID = m_seqID;

            m_pilcRunTot = m_runTot;

            FeederModelMgr.Feeder oFeeder = (FeederModelMgr.Feeder)m_fdrMgr.oFeeder.Clone();


            RelSecRun();


            /* Reset variables */
```

```
            m_actualPer = actual_per;

            m_seqID = m_pilcSeqID;

            m_runTot = m_pilcRunTot;

            m_fdrMgr.oFeeder = (FeederModelMgr.Feeder)oFeeder.Clone();

        }

        private void ReplaceALaCart()

        {

            string boro =  "";

            string network = "";

            string feeder = "";

            DataView fdrsDv = new DataView(GetFdrNwkBoro(ref boro, ref network, ref feeder));
//get feeders for looping

            DataView aLaCartDv = new DataView(PopulateALaCart());

            FeederModelMgr.FdrModMgr m_fdrMgr = new FeederModelMgr.FdrModMgr(boro,
network, feeder);

            FeederModelMgr.Feeder oFeeder;



            for (int i = 0; i < fdrsDv.Count; i++)
```

```
{
    m_feeder = fdrsDv[i].Row["feeder"].ToString().Trim();
    m_Error = "ERROR at ReplaceALaCart(). Feeder: " + m_feeder + "\n";

    if (m_fdrMgr.ChangeFeeder(m_feeder))    //skip feeder with no cables or joints
      continue;

    oFeeder = (FeederModelMgr.Feeder)m_fdrMgr.oFeeder.Clone(); //clone the object to
make sure the data is fresh for new run

    //replace a la cart
    aLaCartDv.RowFilter = "feeder='" + m_feeder + "'";
    for (int x = 0; x < aLaCartDv.Count; x++)
    {
      if (aLaCartDv[x].Row["run_id"] == DBNull.Value)
        continue;

      if (x > 0)
```

```
                m_fdrMgr.oFeeder = (FeederModelMgr.Feeder)oFeeder.Clone();


                int run_id = Convert.ToInt32(aLaCartDv[x].Row["run_id"]);

                int length = Convert.ToInt32(aLaCartDv[x].Row["length"]);


                Hashtable ht = m_fdrMgr.ReplaceMe(run_id, false);

                SaveALaCart(ht, length, run_id);

            }


            //save data for each feeder

            SaveBulkData();

        }


        /* CONFIRMATION message */

        m_Error = "Requested Transaction is completed.\nPlease check table PRECOM_ALaCart
for results.";

    }
```

```
private void SaveALaCart(Hashtable ht, int length, int run_id)

{

  m_Error = "ERROR at SaveALaCart.\nFeeder: " + ht["feeder"].ToString() + ", Run_ID: "
+ run_id;



  //save data to Precom_FeedersAttributes_ALaCart

  int seq_id = CheckFdrAttrs(ht);



  //save data to Precom_ALaCart

  m_sqlRerank.Append("INSERT INTO PRECOM_ALaCart (id, feeder, run_id, seq_id,
version, length, rank, score, date) " +

    "VALUES(newid(), '" + ht["feeder"].ToString() + "', " + run_id + ", " + seq_id + ", " +

    tbVersion.Text.Trim() + ", " + length + ", 0, 0, getdate()); ");

}

private void SaveBulkData()

{

  m_Error = "ERROR at SaveBulkData.\nFeeder: " + m_feeder + "\nLoad Relief " +
m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\nReliabilitiy " + m_relStrat + " " + m_relPer +
"%\n";
```

```
if (m_conn.State == ConnectionState.Closed)
    m_conn.Open();

if (m_sqlRerank.ToString() == "" && m_sqlFdrAttrs.ToString() == "")
    return;

if (m_sqlFdrAttrs.ToString() != "")
    m_sqlRerank.Append(m_sqlFdrAttrs);

SqlCommand cmd = new SqlCommand(m_sqlRerank.ToString(), m_conn);
cmd.CommandTimeout = 120;
cmd.ExecuteNonQuery();
cmd = null;

m_sqlFdrAttrs = new StringBuilder();
m_sqlRerank = new StringBuilder();
}
private void SaveHipotBase(Hashtable ht)
```

```
{
    m_Error = "ERROR at SaveHipotBase() for Feeder: " + ht["feeder"].ToString() + "\n";

    StringBuilder col = new StringBuilder();
    StringBuilder val = new StringBuilder();

    col.AppendFormat("DELETE FROM PRECOM_HipotBase WHERE version={0} AND
feeder='{1}'; ", tbVersion.Text.Trim(), ht["feeder"].ToString());
    col.Append("INSERT INTO PRECOM_HipotBase (feeder, version, date, borough,
network");
    val.AppendFormat("VALUES ('{0}',{1},getdate(),'{2}','{3}'", ht["feeder"].ToString(),
tbVersion.Text.Trim(), ht["borough"].ToString(), ht["network"].ToString());

    col.Append(", static_papercable_gt_40, static_papercable_le_40");
    val.AppendFormat(",{0},{1}", ht["static_papercable_gt_40"].ToString(),
ht["static_papercable_le_40"].ToString());

    col.Append(", static_solidcable_70_75, static_solidcable_lt_16, static_solidcable_ge_16");
```

```
val.AppendFormat(",{0},{1},{2}", ht["static_solidcable_70_75"].ToString(),
ht["static_solidcable_lt_16"].ToString(), ht["static_solidcable_ge_16"].ToString());

col.Append(", static_e2_1_le_1993, static_e2_1_gt_1993");

val.AppendFormat(",{0},{1}", ht["static_e2_1_le_1993"].ToString(),
ht["static_e2_1_gt_1993"].ToString());

col.Append(", static_r3_1_le_1993, static_r3_1_gt_1993");

val.AppendFormat(",{0},{1}", ht["static_r3_1_le_1993"].ToString(),
ht["static_r3_1_gt_1993"].ToString());

col.Append(", static_paperjoints_total");

val.AppendFormat(",{0}", ht["static_paperjoints_total"].ToString());

col.Append(", static_solidjoints_total");

val.AppendFormat(",{0}", ht["static_solidjoints_total"].ToString());

col.Append(", static_other_stops_total");

val.AppendFormat(",{0}", ht["static_other_stops_total"].ToString());
```

col.Append(", static_hipot_freq_paper_gt_40, static_hipot_freq_paper_other");

val.AppendFormat(",{0},{1}", ht["static_hipot_freq_paper_gt_40"].ToString(), ht["static_hipot_freq_paper_other"].ToString());

col.Append(", static_hipot_freq_solid_70_75, static_hipot_freq_solid_lt_16, static_hipot_freq_solid_ge_16");

val.AppendFormat(",{0},{1},{2}", ht["static_hipot_freq_solid_70_75"].ToString(), ht["static_hipot_freq_solid_lt_16"].ToString(), ht["static_hipot_freq_solid_ge_16"].ToString());

col.Append(", static_hipot_freq_e21_pre93, static_hipot_freq_e21_post93");

val.AppendFormat(",{0},{1}", ht["static_hipot_freq_e21_pre93"].ToString(), ht["static_hipot_freq_e21_post93"].ToString());

col.Append(", static_hipot_freq_r31_pre93, static_hipot_freq_r31_post93");

val.AppendFormat(",{0},{1}", ht["static_hipot_freq_r31_pre93"].ToString(), ht["static_hipot_freq_r31_post93"].ToString());

col.Append(", static_hipot_freq_other_stops, static_hipot_freq_paper_ns, static_hipot_freq_solid_ns");

```
val.AppendFormat(",{0},{1},{2}", ht["static_hipot_freq_other_stops"].ToString(), ht["static_hipot_freq_paper_ns"].ToString(), ht["static_hipot_freq_solid_ns"].ToString());

col.Append(", static_hipot_freq_Sum, static_hipot_freq_norm_sum");
val.AppendFormat(",{0},{1}", ht["static_hipot_freq_sum"].ToString(), ht["static_hipot_freq_norm_sum"].ToString());

if (Convert.ToDouble(ht["static_hipot_freq_weighted_norm"]) != -.9999)
{
  col.Append(", static_hipot_freq_weighted_norm");
  val.AppendFormat(",{0}", ht["static_hipot_freq_weighted_norm"]);
}

if (Convert.ToDouble(ht["static_hipot_norm_weighted_norm"]) != -.9999)
{
  col.Append(", static_hipot_norm_weighted_norm");
  val.AppendFormat(",{0}", ht["static_hipot_norm_weighted_norm"]);
}
```

```
    val.Append(") ");
    col.AppendFormat(") {0}", val);



    /* Connection */
    if (m_conn.State == ConnectionState.Closed)
        m_conn.Open();


    /* Execute sql command */
    SqlCommand cmd = new SqlCommand(col.ToString(), m_conn);
    cmd.ExecuteNonQuery();
    cmd = null;
}


private DataTable GetFdrNwkBoro(ref string boro, ref string network, ref string feeder)
{
    m_Error = "ERROR at GetFdrNwkBoro\n";
```

```
string str = "";

if (ddlFeeder.SelectedIndex > 0)
{
   feeder = ddlFeeder.Items[ddlFeeder.SelectedIndex].Value;
   str = "WHERE feeder='" + feeder + "' ";
}
else if (ddlNetwork.SelectedIndex > 0)
{
   network = ddlNetwork.Items[ddlNetwork.SelectedIndex].Value;
   str = "WHERE network='" + network + "' ";
}
else if (ddlBorough.SelectedIndex > 0)
{
   boro = ddlBorough.Items[ddlBorough.SelectedIndex].Value;
   str = "WHERE borough='" + boro + "' ";
}
```

```
            string sql = "SELECT distinct feeder , LTRIM(network) as network, LTRIM(borough) as
borough FROM Precom_Cables_MgrModel " + str + "ORDER BY borough, feeder";

            SqlDataAdapter da = new SqlDataAdapter(sql, m_conn);

            DataTable dt = new DataTable();

            da.Fill(dt);

            da = null;



            return (dt);

        }

        private void GetMaxVersionNum()

        {

            m_Error = "ERROR at GetMaxVersionNum.\n";



            if (m_conn.State == ConnectionState.Closed)

                m_conn.Open();



            string sql = "";
```

```
if (chkALaCart.Checked)
   sql = "SELECT max(version) as version FROM PRECOM_ALaCart";
else if (chkHipotBase.Checked)
   sql = "SELECT max(version) as version FROM PRECOM_HipotBase";
else
   sql = "SELECT max(version) as version FROM PRECOM_Rerank";

SqlCommand cmd = new SqlCommand(sql, m_conn);
SqlDataReader dr = cmd.ExecuteReader();
dr.Read();

if (dr["version"] != System.DBNull.Value)
   tbVersion.Text = Convert.ToString(Convert.ToInt32(dr["version"]) + 1);
else
   tbVersion.Text = "1";

dr.Close();
cmd = null;
```

```
}

private int GetMaxSeqID()

{

   if (m_conn.State == ConnectionState.Closed)

      m_conn.Open();



   string sql = "";

   if (chkALaCart.Checked)

      sql = "SELECT max(seq_id) as maxID FROM PRECOM_FeedersAttributes_ALaCart";

   else

      sql = "SELECT max(seq_id) as maxID FROM PRECOM_FeedersAttributes";



   SqlCommand cmd = new SqlCommand(sql, m_conn);

   SqlDataReader dr = cmd.ExecuteReader();



   dr.Read();

   int seq_id = 0;

   if (dr["maxID"] != System.DBNull.Value)
```

```
            seq_id = Convert.ToInt32(dr["maxID"]) + 1;


            dr.Close();
            cmd = null;
            return seq_id;
        }
        private void GetNetworks(string boro)
        {
            m_Error = "ERROR seleting networks for borough " + boro + ".\n";

            string sql = "SELECT distinct LTRIM(network) as network FROM CanonicalFeeder
WHERE borough='" + boro + "'";

            if (m_conn.State == ConnectionState.Closed)
                m_conn.Open();

            SqlCommand cmd = new SqlCommand(sql, m_conn);
            SqlDataReader dr = cmd.ExecuteReader();
```

```
            ddlNetwork.Items.Clear();

            ddlNetwork.DataSource = dr;

            ddlNetwork.DataTextField = "network";

            ddlNetwork.DataValueField = "network";

            ddlNetwork.DataBind();

            ddlNetwork.Items.Insert(0, new ListItem("All"));



            dr.Close();

            cmd = null;

        }

        private void GetFeeders(string boro, string network)

        {

            m_Error = "ERROR selecting feeders for borough " +
ddlBorough.Items[ddlBorough.SelectedIndex].Text + ".\n";



            string sql = "SELECT distinct zFeeder as feeder FROM CanonicalFeeder WHERE
borough='" + boro + "' AND LTRIM(network)='" + network + "' ORDER BY zFeeder"; //2008 data
```

```
        if (m_conn.State == ConnectionState.Closed)
            m_conn.Open();

        SqlCommand cmd = new SqlCommand(sql, m_conn);
        SqlDataReader dr = cmd.ExecuteReader();

        ddlFeeder.Items.Clear();
        ddlFeeder.DataSource = dr;
        ddlFeeder.DataTextField = "feeder";
        ddlFeeder.DataValueField = "feeder";
        ddlFeeder.DataBind();
        ddlFeeder.Items.Insert(0, new ListItem("All"));

        dr.Close();
        cmd = null;
    }
    private void GetRunID()
```

```
{
    m_Error = "ERROR at GetRunID().\n";

    if (m_conn.State == ConnectionState.Closed)
        m_conn.Open();

    string sql = "SELECT distinct run_id_nulls as run_id FROM Rerank_Cable_Runs
WHERE feeder='" + m_feeder + "' ORDER BY run_id";

    SqlCommand cmd = new SqlCommand(sql, m_conn);
    cmd.CommandTimeout = 180;
    SqlDataReader dr = cmd.ExecuteReader();
    ddlRunID.Items.Clear();
    ddlRunID.DataSource = dr;
    ddlRunID.DataTextField = "run_id";
    ddlRunID.DataValueField = "run_id";
    ddlRunID.DataBind();
    ddlRunID.Items.Insert(0, new ListItem("All"));
```

```
            dr.Close();
            cmd = null;
        }

        private void PrepRerankSql(Hashtable ht)
        {
            //prepare sql string to be inserted and save data to FeedersAttributes hashtable
            if (ht != null)
                m_seqID = CheckFdrAttrs(ht);

            //prepare sql string for Precom_Rerank
            m_Error = "ERROR at PrepRerankSql.\nFeeder: " + m_feeder + "\nLoad Relief " +
m_loadPer + "%\nPILC " + m_pilcStrat + " " + m_pilcPer + "%\nReliabilitiy " + m_relStrat + " " + m_relPer +
"%\n";
            //m_sqlRerank.Append("DELETE FROM PRECOM_Rerank WHERE feeder='" +
m_feeder + "' AND version=" + tbVersion.Text.Trim() +
            //   " AND [load_relief_per]=" + m_loadPer +
            //   " AND pilc_strategy='" + m_pilcStrat + "' " + " AND [pilc_per]=" + m_pilcPer +
```

```
                    //    " AND rel_strategy='" + m_relStrat + "' AND [rel_per]=" + m_relPer + ";" +

                    //    "INSERT INTO PRECOM_Rerank (id, version, feeder, load_relief_per, pilc_strategy,
pilc_per, " +

                    //    "rel_strategy, rel_per, actual_per, sec_changed, score, rank, seq_id, date) " +

                    //    "VALUES(newid(), " + tbVersion.Text.Trim() + ",'" + m_feeder + "'," + m_loadPer +
",'" + m_pilcStrat +

                    //     "'," + m_pilcPer + ",'" + m_relStrat + "'," + m_relPer + "," + m_actualPer + "," +
m_runTot + "," +

                    //    "0,0," + m_seqID + ",getdate()); ");



                    m_sqlRerank.Append("INSERT INTO PRECOM_Rerank (id, version, feeder,
load_relief_per, pilc_strategy, pilc_per, " +

                        "rel_strategy, rel_per, actual_per, sec_changed, score, rank, seq_id, date) " +

                        "VALUES(newid(), " + tbVersion.Text.Trim() + ",'" + m_feeder + "'," + m_loadPer +
",'" + m_pilcStrat +

                        "'," + m_pilcPer + ",'" + m_relStrat + "'," + m_relPer + "," + m_actualPer + "," +
m_runTot + "," +

                        "0,0," + m_seqID + ",getdate()); ");

                }

                private void PrepFdrAttrsSql(Hashtable ht, int seq_id)
```

```
{
    m_Error = "ERROR at PrepFdrAttrsSql.\nFeeder: " + m_feeder + "\nLoad Relief " +
m_loadPer + "%\nPILC: " + m_pilcStrat + " " + m_pilcPer + "%\n" + m_relStrat + " " + m_relPer + "%\n";
    StringBuilder colValu = new StringBuilder();

    if (chkALaCart.Checked)
      m_sqlFdrAttrs.Append("INSERT INTO PRECOM_FeedersAttributes_ALaCart (seq_id,
date");
    else
      m_sqlFdrAttrs.Append("INSERT INTO PRECOM_FeedersAttributes (seq_id,date");

    colValu.AppendFormat("VALUES ({0}, getdate()", seq_id);
    foreach (string key in ht.Keys)
    {
      if (key == "borough" || key == "network" || key == "feeder")
        continue;

      string value = nullValueCheck(Convert.ToDecimal(ht[key]));
```

```
        if (value != "")
        {
            m_sqlFdrAttrs.AppendFormat(",{0}", key);
            colValu.AppendFormat(",{0}", value);
        }
    }

    colValu.Append("); ");
    m_sqlFdrAttrs.Append(") ");
    m_sqlFdrAttrs.Append(colValu);
}
private string nullValueCheck(decimal value)
{
    m_Error = "ERROR at nullValueCheck.\n";
    if (value == Convert.ToDecimal(-.9999))
        return "";
    else
        return value.ToString();
```

```
}

private bool ValidateFields()

{

  if (chkHipotBase.Checked)

    return true;



  bool flag = false;

  if (tbVersion.Text.Trim() == "")

  {

    ErrorMessage("Please enter the version #.");

    tbVersion.Focus();

  }



  else if (!(chkLoadRelief.Checked || chkPilc.Checked || chkRel.Checked ||
chkALaCart.Checked || chkHipotBase.Checked || chkRel.Checked))

    ErrorMessage("Please select tier(s) to be computed.");



  else if (chkRel.Checked)
```

```
        {
            if ((chkLoadRelief.Checked && !chkPilc.Checked) || (!chkLoadRelief.Checked &&
chkPilc.Checked))
                ErrorMessage("Invalid combination of strategies selected.");
            else
                flag = true;
        }
        else
            flag = true;

        return flag;
    }
    private int CheckFdrAttrs(Hashtable ht)
    {
        //create value key for hashtable
        bool first_time = true;
        int new_key = m_maxSeqID + 1;
        StringBuilder new_value = new StringBuilder(); //will use this as the key
```

```
decimal value;
foreach (string key in ht.Keys)
{
  if (key != "feeder" && key != "network" && key != "borough")
  {
    value = Convert.ToDecimal(ht[key]);
    if (first_time)
    {
      new_value.Append(value);
      first_time = false;
    }
    else
      new_value.AppendFormat(",{0}", value);
  }
}

//check if the value already in hashtable; add if not
```

```
    int seq_id = 0;
    if (!FdrAttrsHt.ContainsKey(new_value.ToString()))
    {
        FdrAttrsHt.Add(new_value, new_key);
        m_maxSeqID++;
        seq_id = new_key;
        PrepFdrAttrsSql(ht, seq_id);
    }
    else
        seq_id = Convert.ToInt32(FdrAttrsHt[new_value]);

    return seq_id;
}
private void writeOut_JointsLs(string feeder, ArrayList jntsLs)
{
    //write to output file
    string path = ConfigurationManager.AppSettings["logPath"].ToString();
    string fileLoc = path + "precomOut_Joints_" + feeder + ".txt";
```

```
    using (StreamWriter writer = File.AppendText(fileLoc))
    {
      for (int i = 0; i < jntsLs.Count; i++)
      {
        FeederModelMgr.Joint jnt = (Joint)jntsLs[i];
        writer.WriteLine(jnt.JointID);
      }

      writer.Close();
    }
  }
  private void ErrorMessage() { ErrorMessage(m_Error); }
  private void ErrorMessage(string sError)
  {
    StringBuilder sbAlert = new StringBuilder("");
    sbAlert.Append("\n<script language=javascript>\n");
    sbAlert.Append("function ErrorMessage(){\n");
    sbAlert.AppendFormat("alert('{0}');\n", clientWriter(sError));
```

```
                sbAlert.Append("}\n\n");



                //Add the script to call the function

                sbAlert.Append("ErrorMessage();\n");

                sbAlert.Append("</script>");



                if (!ClientScript.IsClientScriptBlockRegistered("ErrorMessage"))

                   ClientScript.RegisterClientScriptBlock(this.GetType(),          "ErrorMessage",
sbAlert.ToString());

                   return;

               }

               private string clientWriter(string sValue)

               {

/*****************************************************************************

                //To handle writing out to a client script in the page, special characters must

                //be handled.

*****************************************************************************/
```

```
    string sTmp = sValue.Replace("\n", @"\n");
    sValue = sTmp.Replace("\r", @"\r");
    sTmp = sValue.Replace("\t", @"\t");
    return sTmp.Replace("'", @"\'");
}
private void clearGrp2()
{
    chkALaCart.Checked = false;
    ddlRunID.Enabled = false;
    chkHipotBase.Checked = false;
}
private void clearGrp1()
{
    chkPilc.Checked = false;
    chkLoadRelief.Checked = false;
    chkRel.Checked = false;

    ddlLoadPer.Enabled = false;
```

```
    ddlPilePer.Enabled = false;
    ddlRelPer.Enabled = false;

    ddlPileStrat.Enabled = false;
    ddlRelStrat.Enabled = false;

    chk5.Checked = false;
    chk5.Enabled = false;
    ddlPilePer5.Enabled = false;
}

protected void ddlBorough_SelectedIndexChanged(object sender, EventArgs e)
{
    if (ddlBorough.SelectedIndex > 0)
        GetNetworks(ddlBorough.Items[ddlBorough.SelectedIndex].Value.ToString());
    else
        ddlNetwork.Items.Clear();
```

```
            ddlFeeder.Items.Clear();

            ddlRunID.Items.Clear();

        }

        protected void ddlNetwork_SelectedIndexChanged(object sender, EventArgs e)

        {

            /* Get feeders for this network */

            if (ddlNetwork.SelectedIndex > 0)

                GetFeeders(ddlBorough.Items[ddlBorough.SelectedIndex].Value.Trim(),
ddlNetwork.Items[ddlNetwork.SelectedIndex].Value.Trim());

            else

                ddlFeeder.Items.Clear();



            ddlRunID.Items.Clear();

        }

        protected void chkRebuildFreq_CheckedChanged(object sender, EventArgs e)

        {

            if (chkHipotBase.Checked)

            {
```

```
        clearGrp1();
        chkALaCart.Checked = false;
        ddlRunID.Enabled = false;
        //GetMaxVersionNum();
    }
}
protected void chkALaCart_CheckedChanged(object sender, EventArgs e)
{
    if (chkALaCart.Checked)
    {
        clearGrp1();
        chkHipotBase.Checked = false;
        ddlRunID.Enabled = true;
        //GetMaxVersionNum();
    }
    else
        ddlRunID.Enabled = false;
```

```
}
protected void chk5_CheckedChanged(object sender, EventArgs e)
{
  if (chk5.Checked)
  {
    ddlPilcPer.Enabled = false;
    ddlPilcPer.SelectedIndex = 0;
    ddlPilcPer5.Enabled = true;
  }
  else
  {
    ddlPilcPer.Enabled = true;
    ddlPilcPer5.Enabled = false;
    ddlPilcPer5.SelectedIndex = 0;
  }
}
protected void chkLoadRelief_CheckedChanged(object sender, EventArgs e)
{
```

```
    if (chkLoadRelief.Checked)
    {
      ddlLoadPer.Enabled = true;
      if (!chkPilc.Checked && !chkRel.Checked)
      {
        clearGrp2();
        //if ((!chkRel.Checked) && (!chkPilc.Checked))
        //   GetMaxVersionNum();
      }
    }
    else
    {
      ddlLoadPer.Enabled = false;
    }
  }
  protected void chkPilc_CheckedChanged(object sender, EventArgs e)
  {
    if (chkPilc.Checked)
```

```
{
    ddlPilcPer.Enabled = true;
    ddlPilcStrat.Enabled = true;

    if (!chkLoadRelief.Checked && !chkRel.Checked)
    {
        clearGrp2();
        chk5.Enabled = true;

        //if ((!chkLoadRelief.Checked) && (!chkRel.Checked))
        //    GetMaxVersionNum();
    }
}
else
{
    ddlPilcPer.Enabled = false;
    ddlPilcStrat.Enabled = false;
    ddlPilcPer.SelectedIndex = 0;
```

```
        chk5.Enabled = false;
        chk5.Checked = false;
        ddlPilePer5.Enabled = false;
        ddlPilePer5.SelectedIndex = 0;
    }
}
protected void ddlFeeder_SelectedIndexChanged(object sender, EventArgs e)
{
    if (ddlFeeder.SelectedIndex > 0)
    {
        m_feeder = ddlFeeder.Items[ddlFeeder.SelectedIndex].Value;
        GetRunID();
    }
    else
        ddlRunID.Items.Clear();
}
protected void chkReliablilty_CheckedChanged(object sender, EventArgs e)
```

```
{
  if (chkRel.Checked)
  {
    ddlRelStrat.Enabled = true;
    ddlRelPer.Enabled = true;

    if (!chkLoadRelief.Checked && !chkPilc.Checked)
    {
      clearGrp2();
      //if ((!chkLoadRelief.Checked) && (!chkPilc.Checked))
      //   GetMaxVersionNum();
    }
  }
  else
  {
    ddlRelStrat.Enabled = false;
    ddlRelPer.Enabled = false;
  }
```

```
}

//---------------------------------------------

private int SaveDataSet(Hashtable ht)

{

  m_Error = "ERROR at SaveDataSet.\nFeeder: " + m_feeder + "\nLoad Relief " + m_loadPer + "%\nPILC: " + m_pilcStrat + " " + m_pilcPer + "%\nReliabilitiy: " + m_relStrat + " " + m_relPer + "%\n";

  bool first_time = true;

  StringBuilder dataSel = new StringBuilder();

  StringBuilder colName = new StringBuilder();

  StringBuilder colValu = new StringBuilder();


  if (chkALaCart.Checked)

  {

    dataSel.Append("SELECT seq_id FROM PRECOM_FeedersAttributes_ALaCart WHERE ");

    colName.Append("INSERT INTO PRECOM_FeedersAttributes_ALaCart (date");
```

```
}
else
{
   dataSel.Append("SELECT seq_id FROM PRECOM_FeedersAttributes WHERE ");
   colName.Append("INSERT INTO PRECOM_FeedersAttributes (date");
}

colValu.Append("VALUES (getdate()");

foreach (string key in ht.Keys)
{
   if (key != "borough" && key != "network" && key != "feeder")
   {
      string value = nullValueCheck(Convert.ToDecimal(ht[key]));
      if (value != "")
      {
         if (first_time)
         {
```

```
            dataSel.AppendFormat("{0}={1}", key, value);
            first_time = false;
        }
        else
            dataSel.AppendFormat(" AND {0}={1}", key, value);

        colName.AppendFormat(", {0}", key);
        colValu.AppendFormat(", {0}", value);
    }
    else
    {
        if (first_time)
        {
            dataSel.AppendFormat("{0} IS NULL", key);
            first_time = false;
        }
        else
            dataSel.AppendFormat(" AND {0} IS NULL", key);
```

```
        }
    }
}

/* Check to see if the same data set existed.  If yes get the seq_id else get the next seq_id */
m_Error = "ERROR retrieving data from table ";
if (chkALaCart.Checked)
    m_Error += "PRECOM_FeedersAttributes_ALaCart.\n";
else
    m_Error += "PRECOM_FeedersAttributes.\n";

m_Error += "Feeder: " + m_feeder + "\nLoad Relief %: " + m_loadPer +
    "\nPILC Strategy: " + m_pilcStrat + ", PILC %: " + m_pilcPer + "\n";

SqlCommand cmd = new SqlCommand(dataSel.ToString(), m_conn);
SqlDataReader dr = cmd.ExecuteReader();

int seq_id = 0;
```

```
if (dr.HasRows)
{
   dr.Read();
   seq_id = Convert.ToInt32(dr["seq_id"]);
}

dr.Close();
cmd = null;

if (seq_id == 0)
{
   //seq_id = GetNextSeqNum();

   colName.Append(", seq_id) ");
   colValu.Append(", " + seq_id + ")");
   colName.Append(colValu);

   /* Save new data set */
```

```
            m_Error = "ERROR inserting data into table ";
            if (chkALaCart.Checked)
              m_Error += "PRECOM_FeedersAttributes_ALaCart.\n";
            else
              m_Error += "PRECOM_FeedersAttributes.\n";

            m_Error += "Feeder: " + m_feeder + "\nLoad Relief %: " + m_loadPer +
              "\nPILC Strategy: " + m_pilcStrat + ", PILC %: " + m_pilcPer + "\n";

            /* Execute sql command */
            cmd = new SqlCommand(colName.ToString(), m_conn);
            cmd.ExecuteNonQuery();
            cmd = null;
          }

          return seq_id;
        }
      }
```

APPENDIX C

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.SqlClient;
using System.Data;
using System.Collections;
using System.Configuration;


namespace UpdateAttributes
{
   public class UpdateAttributes : Master
   {
      #region protected and private fields
      private int CurrYear;
      private bool IsHipotBase, IsRerank;
      private string Strategy;
      private ArrayList VisitedRuns;
      private DataView VectorsDv, FromToDv;
```

```
        private DataView mCablesDv, mJointsDv, CablesDv, CablesDv_2, JointsDv;

        private DataView VisitedStructsDv, VisitedJointsDv;

        private SqlConnection Conn;

        #endregion



        /// <summary>

        /// entry point for bulk processing

        /// </summary>

        /// <param name="fdrNwkBoro_type">type: ALL, FEEDER, NETWORK, BORO</param>

        /// <param name="fdrNwkBoro">filter: 01B51, 1B, B</param>

        /// <param name="isHipotBase">true if computing new hipot attributes based on most current
state of feeder</param>

        public UpdateAttributes(string fdrNwkBoro_type, string fdrNwkBoro, bool isHipotBase)
//called by Precomp

        {

            InitMe(isHipotBase, false);

            Conn.Open();
```

```
            mCablesDv = new DataView(GetCablesJoints(fdrNwkBoro_type, fdrNwkBoro, true ));
//get cables

            mJointsDv = new DataView(GetCablesJoints(fdrNwkBoro_type, fdrNwkBoro, false));
//get joints



            Conn.Close();

        }



        /// <summary>

        /// interactive mode entry point

        /// </summary>

        public UpdateAttributes()                    //called by Rerank

        {

            bool isHipotBase = false;

            InitMe(isHipotBase, true);

            Strategy = "rerank";

        }
```

```
public DataRow CountMe()                    //called by HipotBase
{
   /* Add new row to store data and initialize it */
   Strategy = "hipotBase";
   InitVectorsRow();

   for (int i = 0; i < CablesDv.Count; i++)
      CountCables(CablesDv[i]);

   return VectorsDv[VectorsDv.Count - 1].Row;
}

public DataRow CountMe(int run_id)          //called by aLaCart
{
   /* Add new row to store data and initialize it */
   Strategy = "aLaCart";
   InitVectorsRow();
```

```
    /* Replace A La Cart */
    CablesDv.RowFilter = "run_id=" + run_id;
    for (int j = 0; j < CablesDv.Count; j++)
       CountCables(CablesDv[j]);

    /* Reset rowfilter */
    CablesDv.RowFilter = "";
    JointsDv.RowFilter = "";

    /* Clear visited structures and joints for next run_id */
    VisitedJointsDv.Table.Clear();
    VisitedStructsDv.Table.Clear();

    return VectorsDv[VectorsDv.Count-1].Row;
}

public DataRow CountMe(int id, bool isSection)        //called by Rerank
{
```

```
    ArrayList idLs = new ArrayList();
    idLs.Add(id);
    CountMe(idLs, isSection);
    return VectorsDv[VectorsDv.Count - 1].Row;
}

public DataRow CountMe(ArrayList idLs, bool isSection, string strategy)  //called by Precom
{
    Strategy = strategy;
    CountMe(idLs, isSection);

    if (VectorsDv.Count > 1)
        return SumVectors();
    else
        return VectorsDv[0].Row;
}

private void CountMe(ArrayList idLs, bool isSection)
```

```
{
  /* Add new row to store data and initialize it */
  InitVectorsRow();

  for (int i = 0; i < idLs.Count; i++)
  {
    /* If run_id already done, skip */
    if (isSection)
      CablesDv.RowFilter = "vision_id=" + Convert.ToInt32(idLs[i]);
    else
      CablesDv.RowFilter = "run_id=" + Convert.ToInt32(idLs[i]);

    for (int j = 0; j < CablesDv.Count; j++)
      CountCables(CablesDv[j]);
  }

  /* Reset rowfilter */
  CablesDv.RowFilter = "";
```

```
    JointsDv.RowFilter = "";
}

public DataTable GetFromToStruct(int id, bool isSection)
{
    if (isSection)
        FromToDv.RowFilter = "vision_id=" + id;
    else
        FromToDv.RowFilter = "run_id=" + id;

    DataTable dt = FromToDv.ToTable();
    FromToDv.RowFilter = "";
    return dt;
}

public void CheckCablesJointsCnt(ref int cables_cnt, ref int joints_cnt)
{
    //call this after calling ChangeFeeder
```

```
    cables_cnt = CablesDv.Count;

    joints_cnt = JointsDv.Count;

}


private DataTable CreateFromToTbl()

{

    DataTable dt = new DataTable();


    dt.Columns.Add("feeder", System.Type.GetType("System.String"));

    dt.Columns.Add("run_id", System.Type.GetType("System.Int32"));

    dt.Columns.Add("vision_id", System.Type.GetType("System.Int32"));

    dt.Columns.Add("from_struct", System.Type.GetType("System.String"));

    dt.Columns.Add("to_struct", System.Type.GetType("System.String"));


    return dt;

}


public DataRow SumVectors()
```

```
{
    DataView dv = new DataView(VectorsDv.Table.Clone());

    /* Create table to store results */
    DataRowView newRow = dv.AddNew();

    /* Sum data */
    newRow.Row["iPaperGT40"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(iPaperGT40)", ""));
    newRow.Row["iPaperLE40"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(iPaperLE40)", ""));

    newRow.Row["iSolidLT16"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(iSolidLT16)" , ""));
    newRow.Row["iSolid70_75"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(iSolid70_75)", ""));
    newRow.Row["iSolidGE16"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(iSolidGE16)" , ""));
```

```
                newRow.Row["papLT8"]     =
Convert.ToInt32(VectorsDv.Table.Compute("sum(papLT8)"     , ""));

                newRow.Row["papBet8A16"]  =
Convert.ToInt32(VectorsDv.Table.Compute("sum(papBet8A16)" , ""));

                newRow.Row["papBet16A24"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(papBet16A24)", ""));

                newRow.Row["papBet24A32"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(papBet24A32)", ""));

                newRow.Row["papBet32A40"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(papBet32A40)", ""));

                newRow.Row["papGE40"]     =
Convert.ToInt32(VectorsDv.Table.Compute("sum(papGE40)", ""));



                newRow.Row["solLT8"]     = Convert.ToInt32(VectorsDv.Table.Compute("sum(solLT8)"
, ""));

                newRow.Row["solBet8A16"]  =
Convert.ToInt32(VectorsDv.Table.Compute("sum(solBet8A16)" , ""));

                newRow.Row["solBet16A24"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(solBet16A24)", ""));

                newRow.Row["solGT24"]     =
Convert.ToInt32(VectorsDv.Table.Compute("sum(solGT24)"    , ""));
```

```
newRow.Row["elasGT1993"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(elasGT1993)", ""));

newRow.Row["elasLE1993"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(elasLE1993)", ""));

newRow.Row["raychem3W1W"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(raychem3W1W)", ""));

newRow.Row["rayGT1993"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(rayGT1993)" , ""));

newRow.Row["rayLE1993"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(rayLE1993)" , ""));

newRow.Row["raychem1W1W"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(raychem1W1W)", ""));

newRow.Row["other1W1W"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(other1W1W)" , ""));

newRow.Row["leadWipeCount"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(leadWipeCount)", ""));
```

```
newRow.Row["oStopLT7"]   =
Convert.ToInt32(VectorsDv.Table.Compute("sum(oStopLT7)"    , ""));

newRow.Row["oStopBet7A12"]=
Convert.ToInt32(VectorsDv.Table.Compute("sum(oStopBet7A12)", ""));

newRow.Row["oStopGT12"]   =
Convert.ToInt32(VectorsDv.Table.Compute("sum(oStopGT12)"   , ""));



newRow.Row["CSR_LE_5percent"]  =
Convert.ToInt32(VectorsDv.Table.Compute("sum(CSR_LE_5percent)" , ""));

newRow.Row["CSR_LE_10percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(CSR_LE_10percent)", ""));

newRow.Row["CSR_LE_15percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(CSR_LE_15percent)", ""));

newRow.Row["CSR_LE_20percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(CSR_LE_20percent)", ""));

newRow.Row["CSR_LE_25percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(CSR_LE_25percent)", ""));

newRow.Row["CSR_LE_30percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(CSR_LE_30percent)", ""));
```

```
                newRow.Row["JR_LE_5percent"]  =
Convert.ToInt32(VectorsDv.Table.Compute("sum(JR_LE_5percent)" , ""));

                newRow.Row["JR_LE_10percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(JR_LE_10percent)", ""));

                newRow.Row["JR_LE_15percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(JR_LE_15percent)", ""));

                newRow.Row["JR_LE_20percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(JR_LE_20percent)", ""));

                newRow.Row["JR_LE_25percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(JR_LE_25percent)", ""));

                newRow.Row["JR_LE_30percent"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(JR_LE_30percent)", ""));



                newRow.Row["strategy"]  = "";

                newRow.Row["cablesCnt"] = Convert.ToInt32(VectorsDv[0].Row["cablesCnt"]);

                newRow.Row["jointsCnt"] = Convert.ToInt32(VectorsDv[0].Row["jointsCnt"]);



                newRow.Row["load_lt80"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_lt80)", ""));
```

```
                newRow.Row["load_gte80_lt90"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_gte80_lt90)", ""));

                newRow.Row["load_gte90_lt100"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_gte90_lt100)", ""));

                newRow.Row["load_gte100_lt110"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_gte100_lt110)", ""));

                newRow.Row["load_gte110_lt120"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_gte110_lt120)", ""));

                newRow.Row["load_gte120_lt130"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_gte120_lt130)", ""));

                newRow.Row["load_gte130_lt140"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_gte130_lt140)", ""));

                newRow.Row["load_gte140"] =
Convert.ToInt32(VectorsDv.Table.Compute("sum(load_gte140)", ""));



                newRow.EndEdit();

                return dv[0].Row;

            }



            /// <summary>
```

```
/// start working with new feeder
/// </summary>
/// <param name="feeder">5-char feeder</param>
/// <param name="isRerank"> </param>
public void ChangeFeeder(string feeder)
{
  ClearMe();

  /* This is used by Rerank application */
  if (IsRerank)
  {
    Conn.Open();

    CablesDv = new DataView(GetCablesJoints(feeder, true ));     //get cables
    JointsDv = new DataView(GetCablesJoints(feeder, false));     //get joints

    Conn.Close();
  }
```

```
    /* This is used by Precom application */
    else
    {
      mCablesDv.RowFilter = "feeder='" + feeder + "'";
      CablesDv   = new DataView(mCablesDv.ToTable());

      mJointsDv.RowFilter = "feeder='" + feeder + "'";
      JointsDv = new DataView(mJointsDv.ToTable());
    }

    /* copy is needed because joints does separate, inner loop over this table */
    CablesDv_2 = new DataView(CablesDv.Table.Copy());
  }

  /// <summary>
  /// initializer
  /// </summary>
```

```
/// <param name="isHipotBase"></param>
private void InitMe(bool isHipotBase, bool isRerank)
{
  IsHipotBase = isHipotBase;
  IsRerank = isRerank;

  CurrYear =  Convert.ToInt32(ConfigurationManager.AppSettings["year"]);
  if (IsRerank)
    FromToDv = new DataView(CreateFromToTbl());

  VectorsDv       = new DataView(CreateVectorsTbl());
  VisitedJointsDv  = new DataView(CreateVisitedTbl());
  VisitedStructsDv = new DataView(CreateVisitedTbl());
  VisitedRuns     = new ArrayList();

  Conn = new SqlConnection(ConfigurationManager.AppSettings["dbConn"]);
}
```

```
private void InitVectorsRow()

{

  DataRowView newRow = VectorsDv.AddNew();



  newRow.Row["iPaperGT40"] = newRow.Row["iPaperLE40"] = 0;

  newRow.Row["iSolidLT16"] = newRow.Row["iSolid70_75"] =
newRow.Row["iSolidGE16"] = 0;



  newRow.Row["papLT8"] = newRow.Row["papBet8A16"] =
newRow.Row["papBet16A24"] = newRow.Row["papBet24A32"] = 0;

  newRow.Row["papBet32A40"] = newRow.Row["papGE40"] = 0;



  newRow.Row["solLT8"] = newRow.Row["solBet8A16"] =
newRow.Row["solBet16A24"] = newRow.Row["solGT24"] = 0;

  newRow.Row["elasGT1993"] = newRow.Row["elasLE1993"] = 0;

  newRow.Row["raychem3w1w"] = newRow.Row["rayGT1993"] =
newRow.Row["rayLE1993"] = 0;

  newRow.Row["raychem1w1w"] = newRow.Row["other1w1w"] = 0;

  newRow.Row["leadWipeCount"] = 0;
```

```
                newRow.Row["oStopLT7"] = newRow.Row["oStopBet7A12"] =
newRow.Row["oStopGT12"] = 0;


                newRow.Row["JR_LE_5percent"]  = newRow.Row["JR_LE_10percent"] =
newRow.Row["JR_LE_15percent"] = 0;

                newRow.Row["JR_LE_20percent"] = newRow.Row["JR_LE_25percent"] =
newRow.Row["JR_LE_30percent"] = 0;


                newRow.Row["CSR_LE_5percent"]  = newRow.Row["CSR_LE_10percent"] =
newRow.Row["CSR_LE_15percent"] = 0;

                newRow.Row["CSR_LE_20percent"] = newRow.Row["CSR_LE_25percent"] =
newRow.Row["CSR_LE_30percent"] = 0;


                newRow.Row["strategy"]  = Strategy;

                newRow.Row["cablesCnt"] = CablesDv.Count;

                newRow.Row["jointsCnt"] = JointsDv.Count;


                newRow.Row["load_lt80"] = 0;

                newRow.Row["load_gte80_lt90"] = newRow.Row["load_gte90_lt100"] =
newRow.Row["load_gte100_lt110"] = 0;
```

```
                newRow.Row["load_gte110_lt120"] = newRow.Row["load_gte120_lt130"] =
newRow.Row["load_gte130_lt140"] = 0;
                newRow.Row["load_gte140"] = 0;

                newRow.EndEdit();
            }

            private void ClearMe()
            {
                VectorsDv.Table.Clear();
                VisitedStructsDv.Table.Clear();
                VisitedJointsDv.Table.Clear();

                if (IsRerank)
                    FromToDv.Table.Clear();

                CablesDv = null;
                JointsDv = null;
```

```
    CablesDv_2 = null;
}

public void ClearMe(string strategy)
{
    VectorsDv.RowFilter = "strategy='" + strategy + "'";
    while (VectorsDv.Count > 0)
        VectorsDv.Delete(0);

    /* Clear duplicate structures */
    VisitedStructsDv.RowFilter = "strategy='" + strategy + "'";
    while (VisitedStructsDv.Count > 0)
        VisitedStructsDv.Delete(0);

    /* Clear duplicate joints */
    VisitedJointsDv.RowFilter = "strategy='" + strategy + "'";
    while (VisitedJointsDv.Count > 0)
        VisitedJointsDv.Delete(0);
```

```
    /* Reset rowfilter */
    CablesDv.RowFilter = "";
    JointsDv.RowFilter = "";

    VectorsDv.RowFilter = "";
    VisitedStructsDv.RowFilter = "";
    VisitedJointsDv.RowFilter = "";
}

private void CountCables(DataRowView cableRow)
{
    /* Replace Cable Section Rank */
    if ((!IsHipotBase) && (cableRow.Row["percentile"] != System.DBNull.Value))
    {
        float percentile = (float)Convert.ToDouble(cableRow.Row["percentile"]);
        CountCSR(percentile);
    }
```

```
/* Insulation */
string insulation;
if (cableRow.Row["insulation"].Equals(DBNull.Value))
   insulation = "EPR";
else
   insulation = cableRow.Row["insulation"].ToString().ToUpper().Trim();


/* Installation date */
int cableYr = 0;
if (cableRow.Row["install_year"].Equals(DBNull.Value))
{
   //impute from mean values
   switch (insulation)
   {
      case "EPR":
         cableYr = 1999;
         break;
```

```
        case "XLP":
            cableYr = 1983;
            break;
        case "PAPER":
            cableYr = 1959;
            break;
    }
}
else
    cableYr = Convert.ToInt32(cableRow.Row["install_year"]);

/* Paper Cable */
DataRowView vectorRow = VectorsDv[VectorsDv.Count - 1];
switch (insulation)
{
    case "PAPER":
        if ((CurrYear - cableYr) >= 40)
```

```
                    vectorRow.Row["iPaperGT40"] =
Convert.ToInt32(vectorRow.Row["iPaperGT40"]) + 1;
                else
                    vectorRow.Row["iPaperLE40"] =
Convert.ToInt32(vectorRow.Row["iPaperLE40"]) + 1;
                break;

            case "XLP":    // three cases: young, in the known-bad period, otherwise old
                if ((CurrYear - cableYr) < 16)
                    vectorRow.Row["iSolidLT16"] =
Convert.ToInt32(vectorRow.Row["iSolidLT16"]) + 1;
                else if ((cableYr >= 1970) && (cableYr <= 1975))
                    vectorRow.Row["iSolid70_75"] =
Convert.ToInt32(vectorRow.Row["iSolid70_75"]) + 1;
                else
                    vectorRow.Row["iSolidGE16"] =
Convert.ToInt32(vectorRow.Row["iSolidGE16"]) + 1;
                break;
```

```
                case "EPR":
                    if ((CurrYear - cableYr) < 16)
                        vectorRow.Row["iSolidLT16"] =
Convert.ToInt32(vectorRow.Row["iSolidLT16"]) + 1;
                    else
                        vectorRow.Row["iSolidGE16"] =
Convert.ToInt32(vectorRow.Row["iSolidGE16"]) + 1;
                    break;
                }

                /* Load Percentage */
                if ((!IsHipotBase) && (cableRow.Row["load_percentage"] != DBNull.Value))
                {
                    int load_pct = Convert.ToInt32(cableRow.Row["load_percentage"]);
                    if (load_pct < 80)
                        vectorRow.Row["load_lt80"] = Convert.ToInt32(vectorRow.Row["load_lt80"]) + 1;
                    else if ((load_pct >= 80) && (load_pct < 90))
                        vectorRow.Row["load_gte80_lt90"] =
Convert.ToInt32(vectorRow.Row["load_gte80_lt90"]) + 1;
```

```
                else if ((load_pct >= 90) && (load_pct < 100))

                    vectorRow.Row["load_gte90_lt100"] =
Convert.ToInt32(vectorRow.Row["load_gte90_lt100"]) + 1;

                else if ((load_pct >= 100) && (load_pct < 110))

                    vectorRow.Row["load_gte100_lt110"] =
Convert.ToInt32(vectorRow.Row["load_gte100_lt110"]) + 1;

                else if ((load_pct >= 110) && (load_pct < 120))

                    vectorRow.Row["load_gte110_lt120"] =
Convert.ToInt32(vectorRow.Row["load_gte110_lt120"]) + 1;

                else if ((load_pct >= 120) && (load_pct < 130))

                    vectorRow.Row["load_gte120_lt130"] =
Convert.ToInt32(vectorRow.Row["load_gte120_lt130"]) + 1;

                else if ((load_pct >= 130) && (load_pct < 140))

                    vectorRow.Row["load_gte130_lt140"] =
Convert.ToInt32(vectorRow.Row["load_gte130_lt140"]) + 1;

                else

                    vectorRow.Row["load_gte140"] = Convert.ToInt32(vectorRow.Row["load_gte140"])
+ 1;

            }
```

```
/* Count Joints */

string from_struct = "";

string to_struct = "";



if (!cableRow.Row["from_structLeg"].Equals(DBNull.Value))

{

   from_struct = cableRow.Row["from_structLeg"].ToString().Trim().ToUpper();

   if (!VisitedStruct(from_struct))

      CountJoints(from_struct);

}



if (!cableRow.Row["to_structLeg"].Equals(DBNull.Value))

{

   to_struct = cableRow.Row["to_structLeg"].ToString().Trim().ToUpper();

   if (!VisitedStruct(to_struct))

      CountJoints(to_struct);

}
```

```
    /* Save from to struct info for Rerank */
    if (IsRerank)
    {
        DataRowView fromTo = FromToDv.AddNew();

        fromTo.Row["feeder"] = cableRow.Row["feeder"].ToString();
        fromTo.Row["run_id"] = Convert.ToInt32(cableRow.Row["run_id"]);
        fromTo.Row["vision_id"] = Convert.ToInt32(cableRow.Row["vision_id"]);
        fromTo.Row["from_struct"] = from_struct;
        fromTo.Row["to_struct"] = to_struct;

        fromTo.EndEdit();
    }
}

private bool VisitedStruct(string structure)
{
    bool flag = false;
```

```
    VisitedStructsDv.RowFilter = "struct='" + structure + "'";



    if (VisitedStructsDv.Count == 0)

    {

      VisitedStructsDv.RowFilter = "";



      DataRowView row = VisitedStructsDv.AddNew();

      row.Row["struct"]   = structure;

      row.Row["strategy"] = Strategy;

      row.EndEdit();

    }

    else

      flag = true;



    return flag;

}



private bool VisitedJoint(int joint)
```

```
{
    bool flag = false;
    VisitedJointsDv.RowFilter = "joint=" + joint;

    if (VisitedJointsDv.Count == 0)
    {
        VisitedJointsDv.RowFilter = "";

        DataRowView row = VisitedJointsDv.AddNew();
        row.Row["joint"]    = joint;
        row.Row["strategy"] = Strategy;
        row.EndEdit();
    }
    else
        flag = true;

    return flag;
}
```

```
private void CountJoints(string structure)

{

  if (structure == "")

    return;



  DataRowView vectorRow = VectorsDv[VectorsDv.Count - 1];



  /* Cable count */

  CablesDv_2.RowFilter = "(from_structLeg='" + structure + "' OR to_structLeg='" +
structure + "') AND insulation='paper'";

  int cnt = CablesDv_2.Count;



  CablesDv_2.RowFilter = "(from_structLeg='" + structure + "' OR to_structLeg='" +
structure + "') AND 1=" + cnt;

  int cableCnt = CablesDv_2.Count;



  /* Joint count */
```

```
JointsDv.RowFilter = "structLeg='" + structure + "'";
for (int i = 0; i < JointsDv.Count; i++)
{
  int joint_id = Convert.ToInt32(JointsDv[i].Row["joint_id"]);

  if (!VisitedJoint(joint_id))
  {
    /* Replace joint rank */
    if ((!IsHipotBase) && (JointsDv[i].Row["percentile"] != DBNull.Value))
    {
      float percentile = (float)(Convert.ToDouble(JointsDv[i].Row["percentile"]));
      CountJR(percentile);
    }

    /* Joint type */
    int jointAge = CurrYear - Convert.ToInt32(JointsDv[i].Row["install_year"]);
    string jointType = JointsDv[i].Row["type"].ToString().ToUpper().Trim();
```

```
switch (jointType)
{
  case "PAPER":
    if (jointAge < 8)
      vectorRow.Row["papLT8"] = Convert.ToInt32(vectorRow.Row["papLT8"]) +
1;
    else if ((jointAge >= 8) && (jointAge < 16))
      vectorRow.Row["papBet8A16"] =
Convert.ToInt32(vectorRow.Row["papBet8A16"]) + 1;
    else if ((jointAge >= 16) && (jointAge < 24))
      vectorRow.Row["papBet16A24"] =
Convert.ToInt32(vectorRow.Row["papBet16A24"]) + 1;
    else if ((jointAge >= 24) && (jointAge < 32))
      vectorRow.Row["papBet24A32"] =
Convert.ToInt32(vectorRow.Row["papBet24A32"]) + 1;
    else if ((jointAge >= 32) && (jointAge < 40))
      vectorRow.Row["papBet32A40"] =
Convert.ToInt32(vectorRow.Row["papBet32A40"]) + 1;
    else
```

```
                    vectorRow.Row["papGE40"] =
Convert.ToInt32(vectorRow.Row["papGE40"]) + 1;
                    break;

                case "SOLID":
                    if (jointAge < 8)
                        vectorRow.Row["solLT8"] = Convert.ToInt32(vectorRow.Row["solLT8"]) +
1;
                    else if ((jointAge >= 8) && (jointAge < 16))
                        vectorRow.Row["solBet8A16"] =
Convert.ToInt32(vectorRow.Row["solBet8A16"]) + 1;
                    else if ((jointAge >= 16) && (jointAge <= 24))
                        vectorRow.Row["solBet16A24"] =
Convert.ToInt32(vectorRow.Row["solBet16A24"]) + 1;
                    else
                        vectorRow.Row["solGT24"] = Convert.ToInt32(vectorRow.Row["solGT24"])
+ 1;
                    break;
                }
```

```
/* Joint stop */
string config = JointsDv[i].Row["config"].ToString().Trim().ToUpper();
string jointStop = JointsDv[i].Row["stop"].ToString().Trim().ToUpper();
int jointYear = Convert.ToInt32(JointsDv[i].Row["install_year"]);

if (jointStop == "S")
{
  /* Elastimold */
  if ((jointType == "ELASTIMOLD") && (config == "2W-1W"))
  {
    if (jointYear > 1993)
      vectorRow.Row["elasGT1993"] =
Convert.ToInt32(vectorRow.Row["elasGT1993"]) + 1;
    else
      vectorRow.Row["elasLE1993"] =
Convert.ToInt32(vectorRow.Row["elasLE1993"]) + 1;
  }
```

```
/* Raychem */

else if ((jointType == "RAYCHEM") && (config == "3W-1W"))

{

   if (cableCnt == 2)

      vectorRow.Row["raychem3w1w"] =
Convert.ToInt32(vectorRow.Row["raychem3w1w"]) + 1;



   if (jointYear > 1993)

      vectorRow.Row["rayGT1993"] =
Convert.ToInt32(vectorRow.Row["rayGT1993"]) + 1;

   else

      vectorRow.Row["rayLE1993"] =
Convert.ToInt32(vectorRow.Row["rayLE1993"]) + 1;

   }



/* Others */

else

{
```

```
if ((config == "1W-1W") && (cableCnt == 2))
{
  if (jointType == "RAYCHEM")
    vectorRow.Row["raychem1w1w"] =
Convert.ToInt32(vectorRow.Row["raychem1w1w"]) + 1;
  else if (jointType == "OTHER")
    vectorRow.Row["other1w1w"] =
Convert.ToInt32(vectorRow.Row["other1w1w"]) + 1;
}

if (jointType == "LW")
  vectorRow.Row["leadWipeCount"] =
Convert.ToInt32(vectorRow.Row["leadWipeCount"]) + 1;

if (jointAge < 7)
  vectorRow.Row["oStopLT7"] =
Convert.ToInt32(vectorRow.Row["oStopLT7"]) + 1;
else if ((jointAge >= 7) && (jointAge <= 12))
```

```
                    vectorRow.Row["oStopBet7A12"] =
Convert.ToInt32(vectorRow.Row["oStopBet7A12"]) + 1;
                else
                    vectorRow.Row["oStopGT12"] =
Convert.ToInt32(vectorRow.Row["oStopGT12"]) + 1;
                }
            }

        } //end if

    } //end for i
}

private DataTable GetCablesJoints(string feeder, bool isCable)
{
    if (Conn.State == ConnectionState.Closed)
        Conn.Open();
```

```
string sql;

if (isCable)

  sql = "SELECT * FROM Precom_Cables_MgrModel WHERE feeder='" + feeder + "'";

else

  sql = "SELECT * FROM Precom_Joints_MgrModel WHERE feeder='" + feeder + "'";



/* Fetch data */

DataTable dt = new DataTable();



SqlDataAdapter da = new SqlDataAdapter(sql, Conn);

da.Fill(dt);

da = null;



return (dt);

}



private DataTable GetCablesJoints(string fdrNwkBoro_type, string fdrNwkBoro, bool
isCable)
```

```
{
  if (Conn.State == ConnectionState.Closed)
    Conn.Open();

  /* Prepare to select by borough, network or feeder */
  string sql;
  if (isCable)
    sql = "SELECT * FROM Precom_Cables_MgrModel ";
  else
    sql = "SELECT * FROM Precom_Joints_MgrModel ";

  switch (fdrNwkBoro_type)
  {
    case "ALL":
      break;

    case "FEEDER":
    case "NETWORK":
```

```
        sql += "WHERE feeder like '%" + fdrNwkBoro + "%' ";

        break;


    case "BORO":

        sql += "WHERE feeder like '%" + fdrNwkBoro + "%' ";


        if (fdrNwkBoro == "B")

            sql += "AND feeder not like '09b%' ";

        else if (fdrNwkBoro == "Q")

            sql += "OR feeder like '09b%' ";


        break;

}


sql += " ORDER BY feeder";


/* Fetch data */

DataTable dt = new DataTable();
```

```
    SqlDataAdapter da = new SqlDataAdapter(sql, Conn);
    da.Fill(dt);
    da = null;

    return (dt);
}

private DataTable CreateVisitedTbl()
{
    DataTable dt = new DataTable();
    dt.Columns.Add("strategy", System.Type.GetType("System.String"));
    dt.Columns.Add("struct", System.Type.GetType("System.String"));
    dt.Columns.Add("joint", System.Type.GetType("System.Int32"));
    return dt;
}

private DataTable CreateVectorsTbl()
```

```
{

  DataTable dt = new DataTable();

  dt.Columns.Add("iPaperGT40", System.Type.GetType("System.Int32"));

  dt.Columns.Add("iPaperLE40", System.Type.GetType("System.Int32"));



  dt.Columns.Add("iSolidLT16", System.Type.GetType("System.Int32"));

  dt.Columns.Add("iSolid70_75", System.Type.GetType("System.Int32"));

  dt.Columns.Add("iSolidGE16", System.Type.GetType("System.Int32"));



  dt.Columns.Add("papLT8", System.Type.GetType("System.Int32"));

  dt.Columns.Add("papBet8A16", System.Type.GetType("System.Int32"));

  dt.Columns.Add("papBet16A24", System.Type.GetType("System.Int32"));

  dt.Columns.Add("papBet24A32", System.Type.GetType("System.Int32"));

  dt.Columns.Add("papBet32A40", System.Type.GetType("System.Int32"));

  dt.Columns.Add("papGE40", System.Type.GetType("System.Int32"));



  dt.Columns.Add("solLT8"    , System.Type.GetType("System.Int32"));

  dt.Columns.Add("solBet8A16" , System.Type.GetType("System.Int32"));
```

```
dt.Columns.Add("solBet16A24", System.Type.GetType("System.Int32"));

dt.Columns.Add("solGT24"    , System.Type.GetType("System.Int32"));


dt.Columns.Add("elasGT1993", System.Type.GetType("System.Int32"));

dt.Columns.Add("elasLE1993", System.Type.GetType("System.Int32"));


dt.Columns.Add("raychem3W1W", System.Type.GetType("System.Int32"));

dt.Columns.Add("rayGT1993"  , System.Type.GetType("System.Int32"));

dt.Columns.Add("rayLE1993"  , System.Type.GetType("System.Int32"));


dt.Columns.Add("raychem1W1W"  , System.Type.GetType("System.Int32"));

dt.Columns.Add("other1W1W"    , System.Type.GetType("System.Int32"));

dt.Columns.Add("leadWipeCount", System.Type.GetType("System.Int32"));


dt.Columns.Add("oStopLT7"    , System.Type.GetType("System.Int32"));

dt.Columns.Add("oStopBet7A12", System.Type.GetType("System.Int32"));

dt.Columns.Add("oStopGT12"   , System.Type.GetType("System.Int32"));
```

```
dt.Columns.Add("CSR_LE_5percent" , System.Type.GetType("System.Decimal"));

dt.Columns.Add("CSR_LE_10percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("CSR_LE_15percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("CSR_LE_20percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("CSR_LE_25percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("CSR_LE_30percent", System.Type.GetType("System.Decimal"));


dt.Columns.Add("JR_LE_5percent" , System.Type.GetType("System.Decimal"));

dt.Columns.Add("JR_LE_10percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("JR_LE_15percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("JR_LE_20percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("JR_LE_25percent", System.Type.GetType("System.Decimal"));

dt.Columns.Add("JR_LE_30percent", System.Type.GetType("System.Decimal"));


dt.Columns.Add("strategy" , System.Type.GetType("System.String"));


dt.Columns.Add("cablesCnt", System.Type.GetType("System.Int32"));

dt.Columns.Add("jointsCnt", System.Type.GetType("System.Int32"));
```

```
    dt.Columns.Add("load_lt80", System.Type.GetType("System.Int32"));

    dt.Columns.Add("load_gte80_lt90", System.Type.GetType("System.Int32"));

    dt.Columns.Add("load_gte90_lt100", System.Type.GetType("System.Int32"));

    dt.Columns.Add("load_gte100_lt110", System.Type.GetType("System.Int32"));

    dt.Columns.Add("load_gte110_lt120", System.Type.GetType("System.Int32"));

    dt.Columns.Add("load_gte120_lt130", System.Type.GetType("System.Int32"));

    dt.Columns.Add("load_gte130_lt140", System.Type.GetType("System.Int32"));

    dt.Columns.Add("load_gte140", System.Type.GetType("System.Int32"));


    return dt;

}


private void CountJR(float percentile)

{

    DataRowView vectorRow = VectorsDv[VectorsDv.Count-1];


    if (percentile <= .3)
```

```
                    vectorRow.Row["JR_LE_30percent"] =
Convert.ToInt32(vectorRow.Row["JR_LE_30percent"]) + 1;

                if (percentile <= .25)

                    vectorRow.Row["JR_LE_25percent"] =
Convert.ToInt32(vectorRow.Row["JR_LE_25percent"]) + 1;

                if (percentile <= .20)

                    vectorRow.Row["JR_LE_20percent"] =
Convert.ToInt32(vectorRow.Row["JR_LE_20percent"]) + 1;

                if (percentile <= .15)

                    vectorRow.Row["JR_LE_15percent"] =
Convert.ToInt32(vectorRow.Row["JR_LE_15percent"]) + 1;

                if (percentile <= .10)

                    vectorRow.Row["JR_LE_10percent"] =
Convert.ToInt32(vectorRow.Row["JR_LE_10percent"]) + 1;

                if (percentile <= .05)

                    vectorRow.Row["JR_LE_5percent"]  =
Convert.ToInt32(vectorRow.Row["JR_LE_5percent" ]) + 1;

                }



                private void CountCSR(float percentile)
```

```
{
    DataRowView vectorRow = VectorsDv[VectorsDv.Count - 1];

    if (percentile <= .3)
      vectorRow.Row["CSR_LE_30percent"] =
Convert.ToInt32(vectorRow.Row["CSR_LE_30percent"]) + 1;
    if (percentile <= .25)
      vectorRow.Row["CSR_LE_25percent"] =
Convert.ToInt32(vectorRow.Row["CSR_LE_25percent"]) + 1;
    if (percentile <= .20)
      vectorRow.Row["CSR_LE_20percent"] =
Convert.ToInt32(vectorRow.Row["CSR_LE_20percent"]) + 1;
    if (percentile <= .15)
      vectorRow.Row["CSR_LE_15percent"] =
Convert.ToInt32(vectorRow.Row["CSR_LE_15percent"]) + 1;
    if (percentile <= .10)
      vectorRow.Row["CSR_LE_10percent"] =
Convert.ToInt32(vectorRow.Row["CSR_LE_10percent"]) + 1;
    if (percentile <= .05)
```

```
                    vectorRow.Row["CSR_LE_5percent"] =
Convert.ToInt32(vectorRow.Row["CSR_LE_5percent"]) + 1;
                }
            }
        }
```

CLAIMS

1. A capital asset planning system for selecting assets for improvement within an infrastructure comprising:

(a) one or more data sources descriptive of the infrastructure;

(b) one or more databases , coupled to the one or more data sources, to compile the one or more data sources;

(c) one or more processors, each coupled to and having respective communication interfaces to receive data from the one or more databases, the one or more processors comprising a predictor to generate:

(i) a first metric of estimated infrastructure effectiveness based, at least in part, on a current status of the infrastructure;

(ii) a second metric of estimated infrastructure effectiveness based, at least in part, on a user-selected, proposed changed configuration of the infrastructure; and

(iii) a net metric of infrastructure effectiveness based, at least in part, on said first metric and said second metric; and

(d) a display, coupled to have the one or more processors, for visually presenting the net metric of infrastructure effectiveness;

wherein the assets for improvement are selected based, at least in part, on the net metric of infrastructure effectiveness.

2. The capital asset planning system of claim 1, wherein the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness are both based, at least in part, on an estimated length of time that the infrastructure produces a commodity.

3. The capital asset planning system of claim 1, wherein the infrastructure is selected from the group consisting of a chemical processing operation, a petroleum refining

operation, a product manufacturing operation, a telecommunication grid, a transportation infrastructure, a gas network, a commodity pipeline network, and a water treatment network.

4. The capital asset planning system of claim 1, wherein the infrastructure is an electrical grid.

5. The capital asset planning system of claim 4, wherein the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness are based, at least in part, on the estimated mean time between failure of one or more components of the electrical grid.

6. The capital asset planning system of claim 4, wherein the one or more data sources descriptive of the infrastructure include data representative of at least one of electrical feeder data, electrical cable data, electrical joint data, electrical transformer data, electrical outage data, electrical test pass or fail data, electrical load data, and past capital improvement cost data.

7. The capital asset planning system of claim 4, wherein the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness is estimated using a feeder index, said index based, at least in part, on a predicted likelihood of feeder failure.

8. The capital asset planning system of claim 7, wherein the predicted likelihood of feeder failure is obtained using machine learning.

9. The capital asset planning system of claim 7, wherein the feeder index is recalculated based, at least in part, on observed and previously predicted feeder data.

10. The capital asset planning system of claim 5, wherein the estimated mean time between failure is the estimated mean time between failure of an electrical feeder.

11. The capital asset planning system of claim 1, wherein the predictor further generates a cost metric based, at least in part, on the cost of the user-selected, proposed change of the infrastructure.

12. The capital asset planning system of claim 11, wherein the cost metric is generated based, at least in part, on a user-specified cost of one or more specific actions encompassed by the user-selected, proposed change of the infrastructure.

13. The capital asset planning system of claim 11, wherein the one or more data sources include past capital improvement cost data, and the cost metric is based, at least in part, on past capital improvement cost data.

14. The capital asset planning system of claim 1, wherein the predictor further comprises a prioritizer to determine the user-selected, proposed changed configuration of the infrastructure that provides the maximum value based, at least in part, on the net metric of infrastructure effectiveness and the cost metric.

15. A method for selecting assets for improvement within an infrastructure comprising:

(a) accessing one or more data sources descriptive of the infrastructure;

(b) compiling the one or more data sources into one or more databases;

(c) generating a first metric of estimated infrastructure effectiveness based, at least in part, on a current status of the infrastructure;

(d) generating a second metric of estimated infrastructure effectiveness based, at least in part, on a user-selected, proposed changed configuration of the infrastructure; and

(e) generating a net metric of infrastructure effectiveness based, at least in part, on said first metric and said second metric; and

(f) displaying the net metric of infrastructure effectiveness,

wherein the assets for improvement are selected based, at least in part, on the net metric of infrastructure effectiveness.

16. The method for selecting assets for improvement of claim 15 wherein the infrastructure is an electrical grid, and said first metric, said second metric and said net metric is based, at least in part, on the estimated mean time between failure of an electrical feeder within the electrical grid.

17. The method for selecting assets for improvement of claim 15, wherein the first metric of infrastructure effectiveness and the second metric of infrastructure effectiveness is estimated using a feeder index, said index based, at least in part, on a predicted likelihood of feeder failure.

18. The method for selecting assets for improvement of claim 15, further comprising generating a cost metric based on the cost of the user-selected, proposed change of the infrastructure.

19. The method for selecting assets for improvement of claim 15, further comprising determining the user-selected, proposed changed configuration of the infrastructure that provides the maximum value based on the net metric of infrastructure effectiveness and the cost metric.

20. A capital asset planning system for selecting assets for improvement within an electrical grid comprising:

(a) one or more data sources descriptive of the infrastructure selected from electrical feeder data, electrical cable data, electrical joint data, electrical transformer data,

electrical outage data, electrical test pass or fail data, electrical load data, and past capital improvement cost data;

(b) one or more databases, coupled to the one or more data sources, to compile the one or more data sources;

(c) one or more processors, each coupled to the one or more databases and having respective communication interfaces to receive data from the one or more databases, the one or more processors comprising a predictor to generate:

(i) an estimated baseline mean time between failure of an electrical feeder;

(ii) an estimated projected mean time between failure of an electrical feeder based, at least in part, on a user-selected, proposed changed configuration of the electrical grid; and

(iii) a net mean time between failure based, at least in part on the estimated baseline mean time between failure and the estimated projected mean time between failure; and

(d) a display, coupled to have the one or more processors, for visually presenting the net metric of infrastructure effectiveness and a cost metric of the proposed changed configuration of the electrical grid,

wherein the capital asset planning system provides the ability to select assets for improvement based, at least in part, on the net mean time between failure and the cost metric.

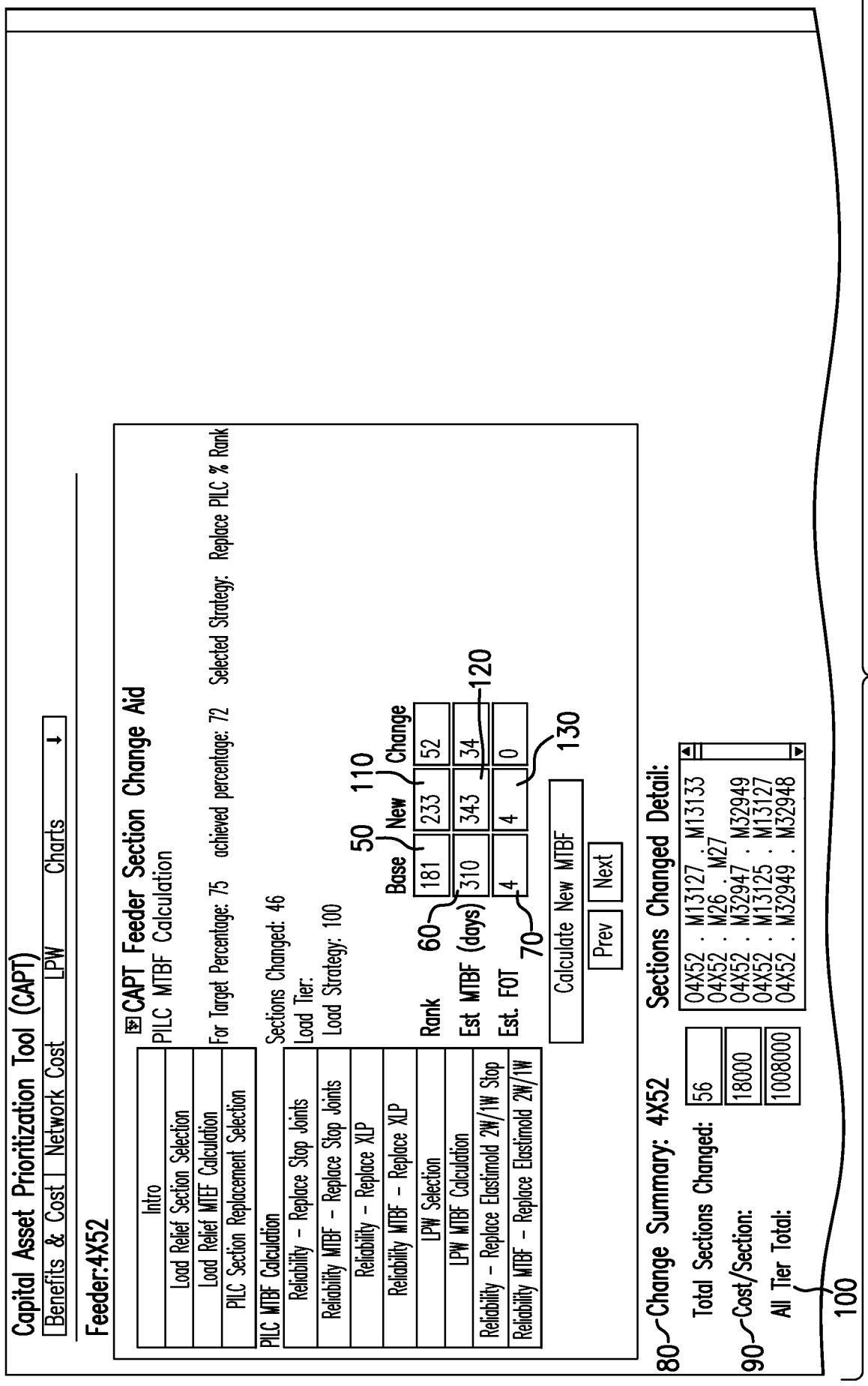

Capital Asset Prioritization Tool (CAPT)
Benefits & Cost
Network Cost
LPW
Charts
Feeder:4X52
Intro
Load Relief Section Selection
Load Relief MTEF Calculation
PILC Section Replacement Selection
PILC MTBF Calculation
Reliability – Replace Stop Joints
Reliability MTBF – Replace Stop Joints
Reliability – Replace XLP
Reliability MTBF – Replace XLP
LPW Selection
LPW MTBF Calculation
Reliability – Replace Elastimold 2W/1W Stop
Reliability MTBF – Replace Elastimold 2W/1W
CAPT Feeder Section Change Aid
PILC MTBF Calculation
For Target Percentage: 75 achieved percentage: 72 Selected Strategy: Replace PILC % Rank
Sections Changed: 46
Load Tier:
Load Strategy: 100
50
110
Base
New
Change
Rank
60
181
233
52
Est MTBF (days)
310
343
34
120
Est. FOT
70
4
4
0
130
Calculate New MTBF
Prev
Next
80
Change Summary: 4X52
Sections Changed Detail:
Total Sections Changed:
56
90
Cost/Section:
18000
All Tier Total:
1008000
04X52 . M13127 . M13133
04X52 . M26 . M27
04X52 . M32947 . M32949
04X52 . M13125 . M13127
04X52 . M32949 . M32948
100
Continued on FIG.1B

FIG.1A

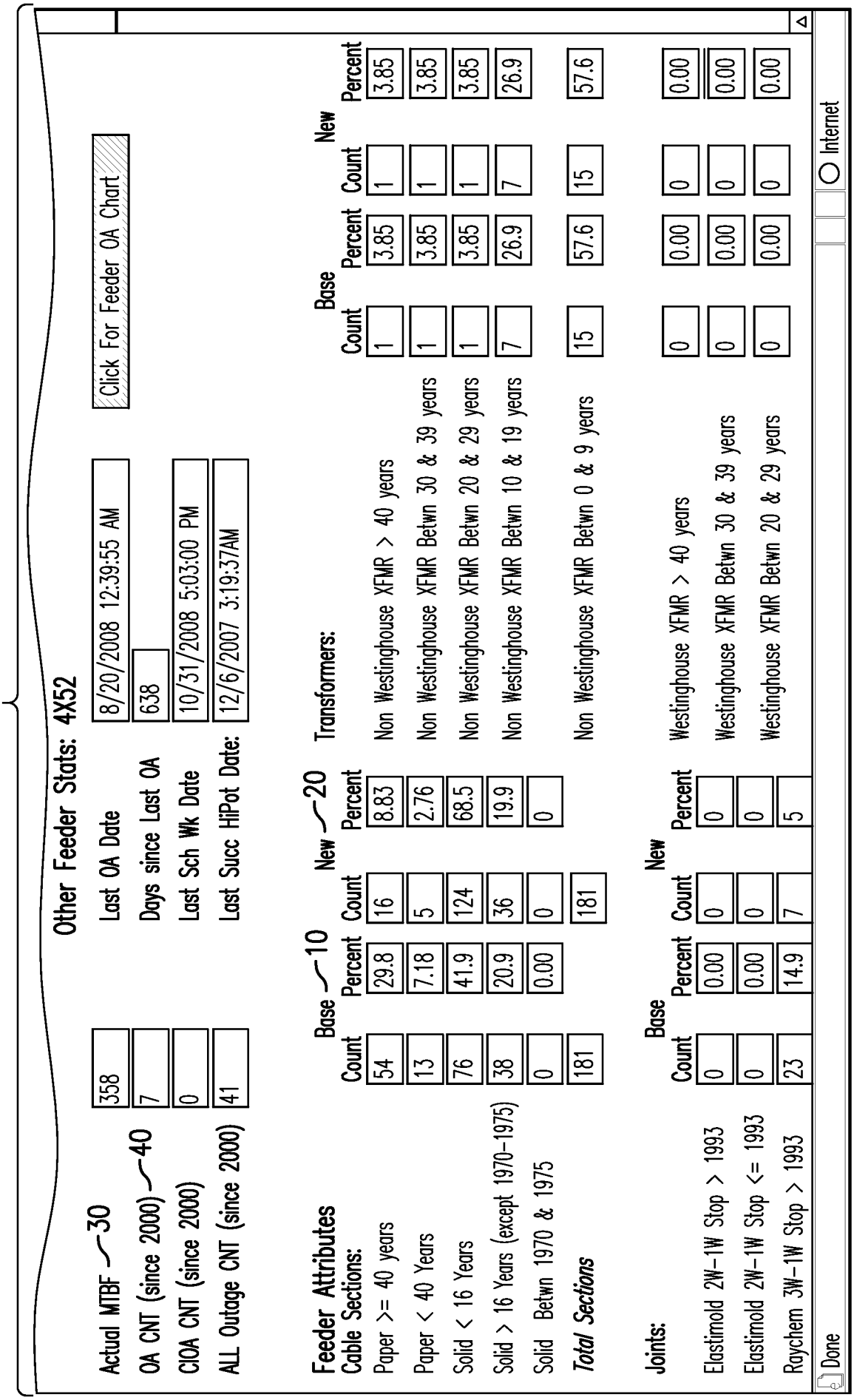

From FIG.1A
Other Feeder Stats: 4X52
Actual MTBF 30
OA CNT (since 2000) 40
ClOA CNT (since 2000)
ALL Outage CNT (since 2000)
358
7
0
41
Last OA Date
Days since Last OA
Last Sch Wk Date
Last Succ HiPot Date:
8/20/2008 12:39:55 AM
638
10/31/2008 5:03:00 PM
12/6/2007 3:19:37AM
Click For Feeder OA Chart
Feeder Attributes
Cable Sections:
Paper >= 40 years
Paper < 40 Years
Solid < 16 Years
Solid > 16 Years (except 1970-1975)
Solid Betwn 1970 & 1975
Total Sections
Base 10
New 20
Count
Percent
54 29.8 16 8.83
13 7.18 5 2.76
76 41.9 124 68.5
38 20.9 36 19.9
0 0.00 0 0
181 181
Joints:
Elastimold 2W-1W Stop > 1993
Elastimold 2W-1W Stop <= 1993
Raychem 3W-1W Stop > 1993
0 0.00 0 0
0 0.00 0 0
23 14.9 7 5
Transformers:
Non Westinghouse XFMR > 40 years
Non Westinghouse XFMR Betwn 30 & 39 years
Non Westinghouse XFMR Betwn 20 & 29 years
Non Westinghouse XFMR Betwn 10 & 19 years
Non Westinghouse XFMR Betwn 0 & 9 years
Westinghouse XFMR > 40 years
Westinghouse XFMR Betwn 30 & 39 years
Westinghouse XFMR Betwn 20 & 29 years
1 3.85 1 3.85
1 3.85 1 3.85
1 3.85 1 3.85
7 26.9 7 26.9
15 57.6 15 57.6
0 0.00 0 0.00
0 0.00 0 0.00
0 0.00 0 0.00
Done
Internet

FIG.1B

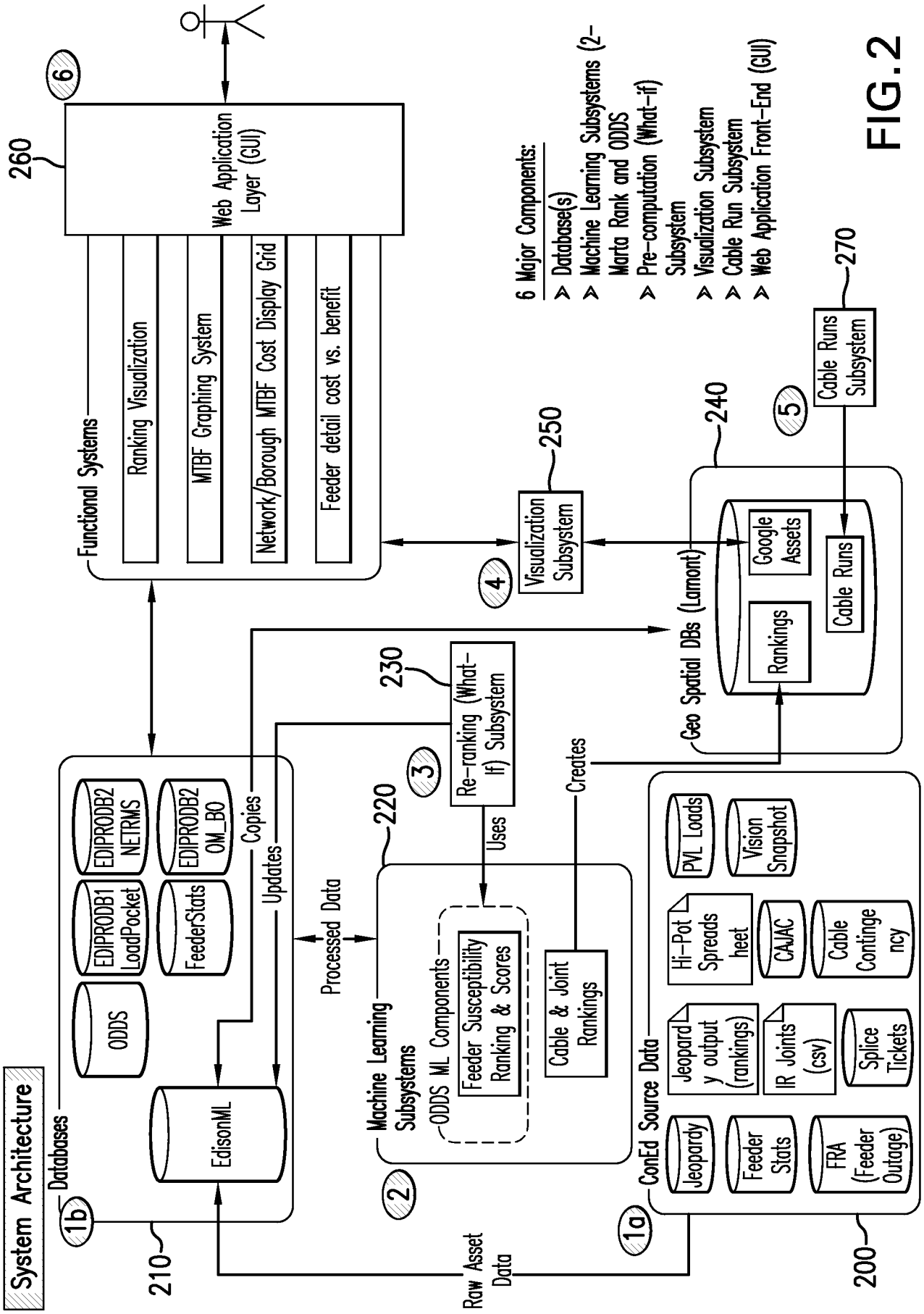

System Architecture
Databases
1b
210
EDIPRODB2 NETRMS
EDIPRODB2 OM_BO
EDIPRODB1 LoadPocket
FeederStats
ODDS
EdisonML
Copies
Updates
Processed Data
Machine Learning Subsystems
ODDS ML Components
Feeder Susceptibility Ranking & Scores
Cable & Joint Rankings
2
220
Uses
Creates
3
Re-ranking (What-If) Subsystem
230
Raw Asset Data
1a
ConEd Source Data
Jeopardy
Feeder Stats
FRA (Feeder Outage)
Jeopardy output (rankings)
IR Joints (csv)
Splice Tickets
Hi-Pot Spreadsheet
CAIAC
Cable Contingency
PVL Loads
Vision Snapshot
200
Geo Spatial DBs (Lamont)
Rankings
Google Assets
Cable Runs
240
4
Visualization Subsystem
250
5
Cable Runs Subsystem
270
Functional Systems
Ranking Visualization
MTBF Graphing System
Network/Borough MTBF Cost Display Grid
Feeder detail cost vs. benefit
6
Web Application Layer (GUI)
260
6 Major Components:
» Database(s)
» Machine Learning Subsystems (2- Marta Rank and ODDS
» Pre-computation (What-if) Subsystem
» Visualization Subsystem
» Cable Run Subsystem
» Web Application Front-End (GUI)

FIG. 2

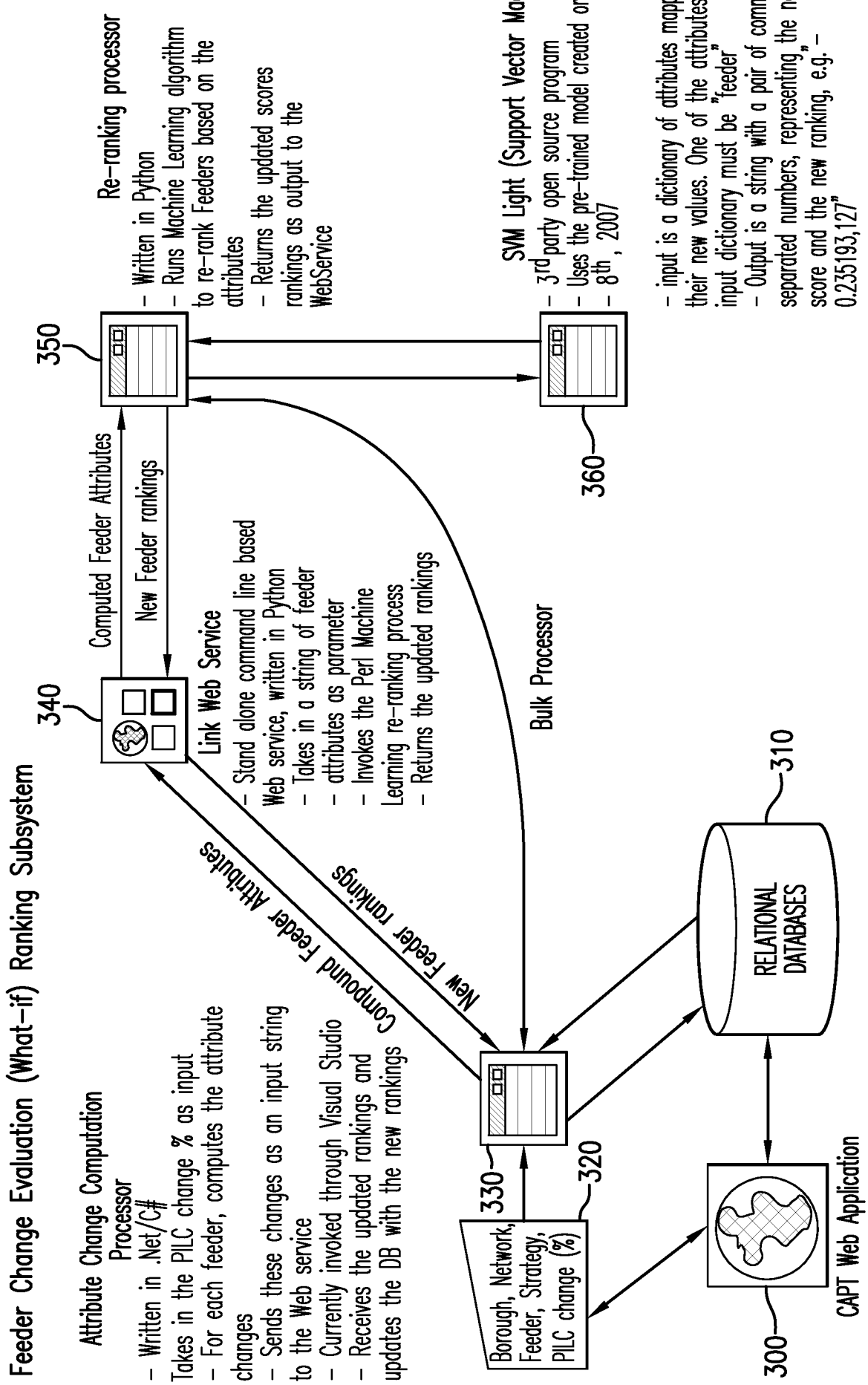

Feeder Change Evaluation (What-if) Ranking Subsystem
Attribute Change Computation Processor
- Written in .Net/C#
Takes in the PILC change % as input
- For each feeder, computes the attribute changes
- Sends these changes as an input string to the Web service
- Currently invoked through Visual Studio
- Receives the updated rankings and updates the DB with the new rankings
Compound Feeder Attributes
New Feeder rankings
340
Link Web Service
- Stand alone command line based Web service, written in Python
- Takes in a string of feeder attributes as parameter
- Invokes the Perl Machine Learning re-ranking process
- Returns the updated rankings
Computed Feeder Attributes
New Feeder rankings
350
Re-ranking processor
- Written in Python
- Runs Machine Learning algorithm to re-rank Feeders based on the attributes
- Returns the updated scores rankings as output to the WebService
Bulk Processor
360
SVM Light (Support Vector Machine)
- 3rd party open source program
- Uses the pre-trained model created on August 8th, 2007
- input is a dictionary of attributes mapped to their new values. One of the attributes in the input dictionary must be "feeder"
- Output is a string with a pair of comma-separated numbers, representing the new score and the new ranking, e.g.- 0.235193,127"
330
320
Borough, Network, Feeder, Strategy, PILC change (%)
310
RELATIONAL DATABASES
300
CAPT Web Application

FIG.3

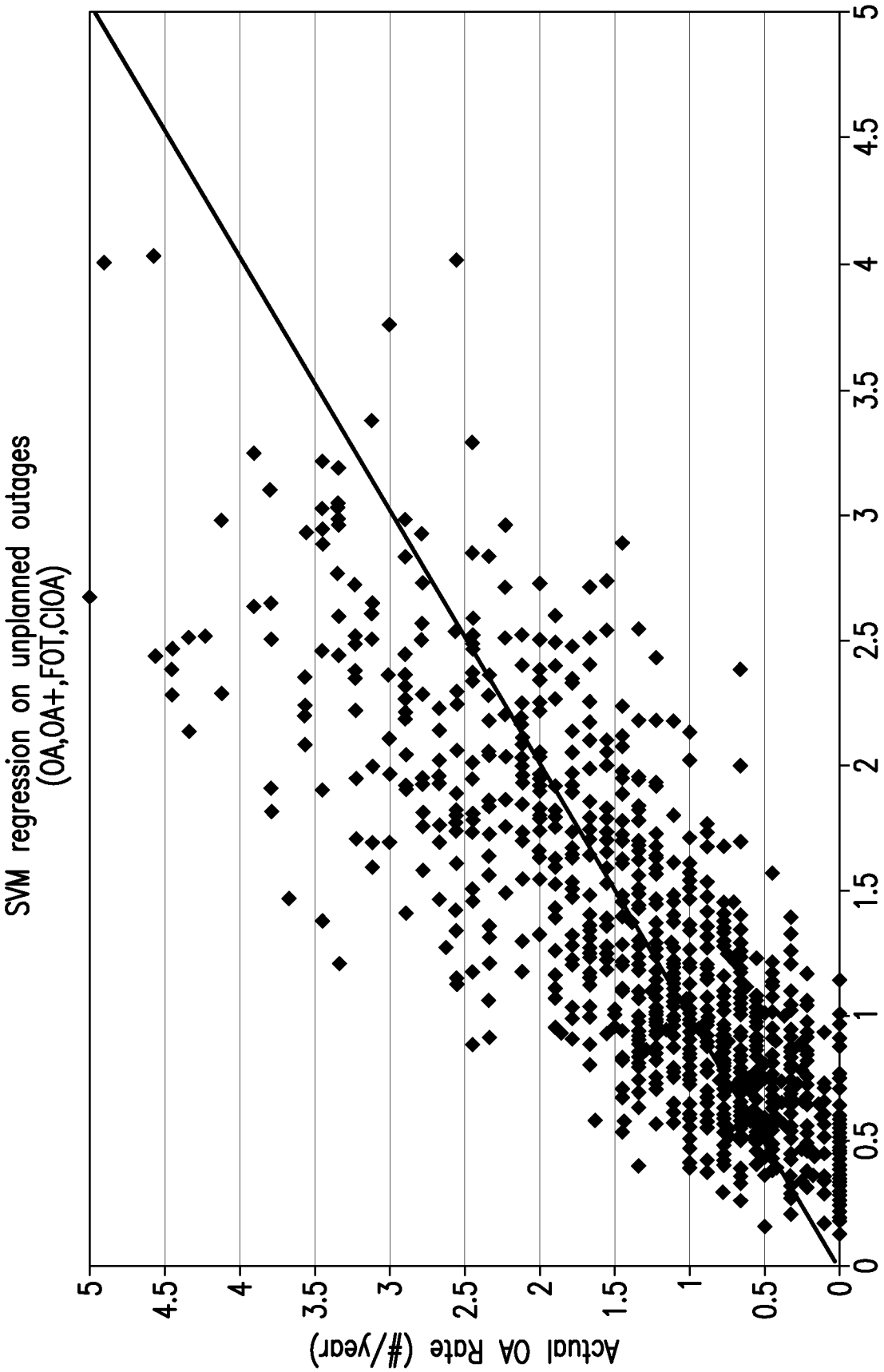

SVM regression on unplanned outages
(OA,OA+,FOT,ClOA)
Actual OA Rate (#/year)
Predicted OA Rate (#/year)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

FIG.4

Capital Asset Prioritization Tool (CAPT)
Benefits & Cost
Network Cost
LPW
Charts
Feeder:4X40
Intro
Load Relief Section Selection
Load Relief MTBF Calculation
PILC Section Replacement Selection
PILC MTBF Calculation
Reliability – Replace Stop Joints
Reliability MTBF – Replace Stop Joints
Reliability – Replace XLP
Reliability MTBF – Replace XLP
LPW Selection
LPW MTBF Calculation
Reliability – Replace Elastimold 2W/1W Stop
Reliability MTBF – Replace Elastimold 2W/1W
CAPT Feeder Section Change Aid
Introduction
The next set of steps will help you calculate the costs and benefits of replacing cable sections (and joints) for Load Relief, PILC Replacement, and Reliability goals. First select the feeder of interest below
Feeder Selection
Select Borough:
Bronx
Feeder maps
720
Select Network:
Central Bronx (4X)
Feeder maps
Criteria:
All
Percentage >=
100
710
Select Feeder:
4X40
Feeder maps
700
Start
Change Summary: 4X40
Total Sections Changed:
20
Cost/Section:
18000
All Tier Total:
360000
730
Sections Changed Detail:
04X40 . M16205 . M16210
04X40 . M16160 . M16164
04X40 . M16191 . M16194
04X40 . M23077 . M25916
04X40 . CM2_V2288
04X40 . M13590 . M21002
740
Other Feeder Stats: 4X40
Actual MTBF
413
OA CNT (since 2000)
5
CIOA CNT (since 2000)
0
ALL Outage CNT (since 2000)
31
Last OA Date
2/18/2007 6:01:06 AM
Days since Last OA
1187
Last Sch Wk Date
8/22/2008 8:14:00 AM
Last Succ HiPot Date:
1/21/2008 7:27:38 AM
750
Click For Feeder OA Chart
Continued on FIG.5B

FIG.5A

From FIG.5A

Feeder Attributes

Cable Sections:	Base Count	Base Percent	New Count	New Percent
Paper >= 40 years	36	27.0	20	15.0
Paper < 40 years	9	6.77	5	3.75
760 Solid < 16 Years	55	41.3	75	56.4
Solid > 16 Years (except 1970-1975)	33	24.8	33	24.8
Solid Betwn 1970 & 1975	0	0.00	0	0
Total Sections	133		133	

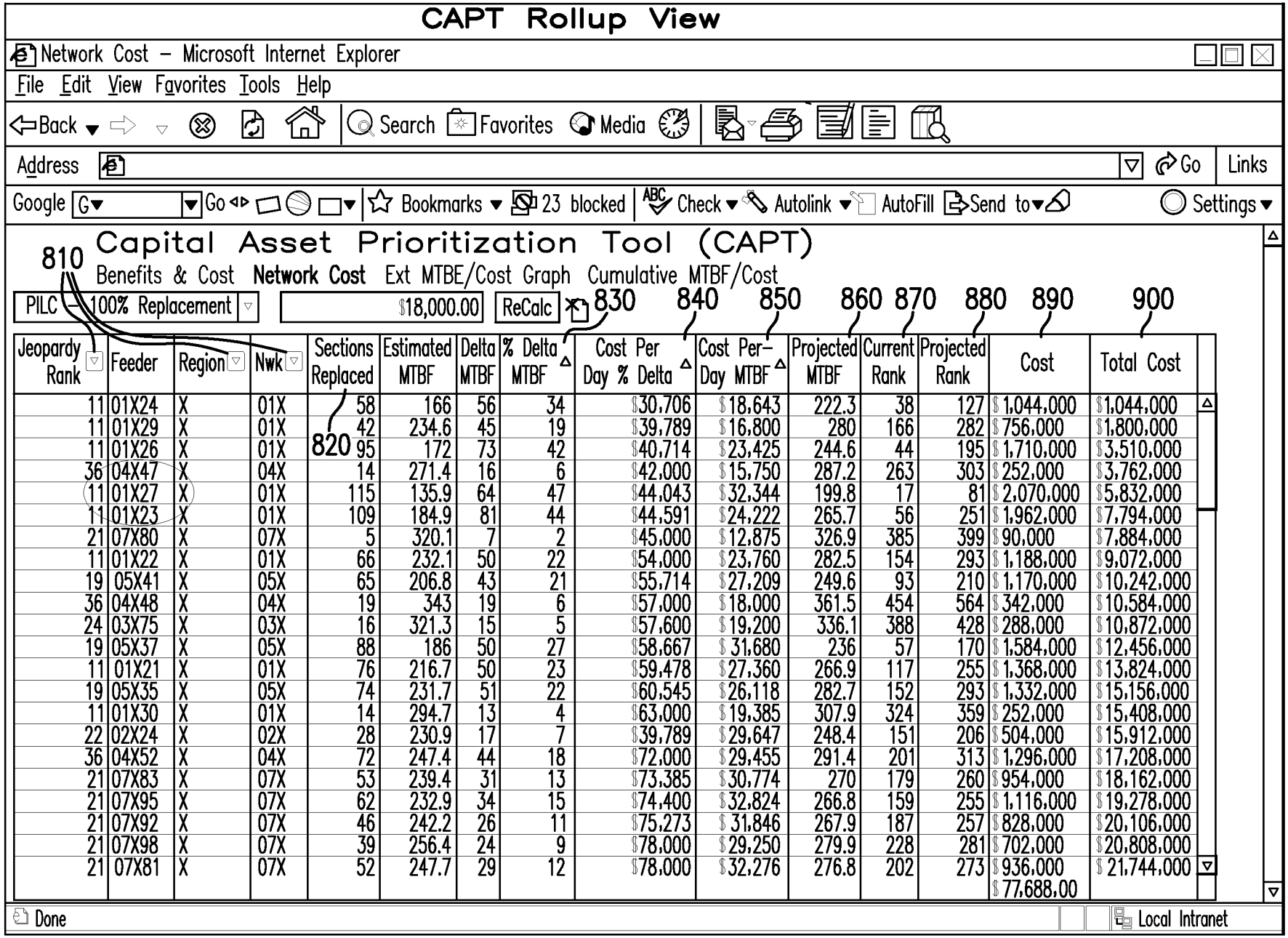
Joints:	Base Count	Base Percent	New Count	New Percent
Elastimold 2W-1W Stop > 1993	0	0.00	0	0
Elastimold 2W-1W Stop <= 1993	0	0.00	0	0
Raychem 3W-1W Stop > 1993	13	10.5	8	7.20
Raychem 3W-1W Stop <= 1993	11	9.17	5	4.50
Other Stop Joints < 7 years	5	4.17	3	2.70
Other Stop Joints Betwn 7 & 12 years	0	0.00	0	0
Other Stop Joints > 12 years	5	4.17	3	2.70
770 *Stop Joint Total*	34	28.2	19	17.1
Paper Joints < 8 years	0	0.00	0	0
Paper Joints Betwn 8 & 16 years	0	0.00	0	0
Paper Joints Betwn 16 & 24 years	0	0.00	0	0
Paper Joints Betwn 24 & 32 years	1	0.83	0	0
Paper Joints Betwn 32 & 40 years	10	8.33	5	4.50
Paper Joints > 40 years	17	14.1	12	10.8
Total Paper Joints	26	23.3	17	15.3

Transformers:	Base Count	Base Percent	New Count	New Percent
Non Westinghouse XFMR > 40 years	1	4.55	1	4
Non Westinghouse XFMR Betwn 30 & 39 years	2	9.09	2	9
Non Westinghouse XFMR Betwn 20 & 29 years	0	0.00	0	0 780
Non Westinghouse XFMR Betwn 10 & 19 years	4	18.1	4	1
Non Westinghouse XFMR Betwn 0 & 9 years	14	63.6	14	6
Westinghouse XFMR > 40 years	1	4.55	1	4
Westinghouse XFMR Betwn 30 & 39 years	0	0.00	0	0
Westinghouse XFMR Betwn 20 & 29 years	0	0.00	0	0
Westinghouse XFMR Betwn 10 & 19 years	0	0.00	0	0
Westinghouse XFMR Betwn 0 & 9 years	0	0.00	0	0
Total XFMR	22		22	

Load: (Requires external programs: Jeopardy and PVL)	Base Count	Base Percent	New Count	New Percent
Feeder Load	385		385	
Normal Feeder Rating	490		490	
Emergency Feeder Rating	755		755	
Emergency Feeder Load	555		555	790
4KV Feeder Count	0		0	
Load Pocket Weight: Total and Avg for Feeder	11	0.36	11	0
Feeder Importance to Jeopardy	8.75		8.75	

FIG.5B

SUBSTITUTE SHEET (RULE 26)



Jeopardy Rank	Feeder	Region	Nwk	Sections Replaced	Estimated MTBF	Delta MTBF	% Delta MTBF	Cost Per Day % Delta	Cost Per-Day MTBF	Projected MTBF	Current Rank	Projected Rank	Cost	Total Cost
11	01X24	X	01X	58	166	56	34	$30,706	$18,643	222.3	38	127	$1,044,000	$1,044,000
11	01X29	X	01X	42	234.6	45	19	$39,789	$16,800	280	166	282	$756,000	$1,800,000
11	01X26	X	01X	95	172	73	42	$40,714	$23,425	244.6	44	195	$1,710,000	$3,510,000
36	04X47	X	04X	14	271.4	16	6	$42,000	$15,750	287.2	263	303	$252,000	$3,762,000
11	01X27	X	01X	115	135.9	64	47	$44,043	$32,344	199.8	17	81	$2,070,000	$5,832,000
11	01X23	X	01X	109	184.9	81	44	$44,591	$24,222	265.7	56	251	$1,962,000	$7,794,000
21	07X80	X	07X	5	320.1	7	2	$45,000	$12,875	326.9	385	399	$90,000	$7,884,000
11	01X22	X	01X	66	232.1	50	22	$54,000	$23,760	282.5	154	293	$1,188,000	$9,072,000
19	05X41	X	05X	65	206.8	43	21	$55,714	$27,209	249.6	93	210	$1,170,000	$10,242,000
36	04X48	X	04X	19	343	19	6	$57,000	$18,000	361.5	454	564	$342,000	$10,584,000
24	03X75	X	03X	16	321.3	15	5	$57,600	$19,200	336.1	388	428	$288,000	$10,872,000
19	05X37	X	05X	88	186	50	27	$58,667	$31,680	236	57	170	$1,584,000	$12,456,000
11	01X21	X	01X	76	216.7	50	23	$59,478	$27,360	266.9	117	255	$1,368,000	$13,824,000
19	05X35	X	05X	74	231.7	51	22	$60,545	$26,118	282.7	152	293	$1,332,000	$15,156,000
11	01X30	X	01X	14	294.7	13	4	$63,000	$19,385	307.9	324	359	$252,000	$15,408,000
22	02X24	X	02X	28	230.9	17	7	$39,789	$29,647	248.4	151	206	$504,000	$15,912,000
36	04X52	X	04X	72	247.4	44	18	$72,000	$29,455	291.4	201	313	$1,296,000	$17,208,000
21	07X83	X	07X	53	239.4	31	13	$73,385	$30,774	270	179	260	$954,000	$18,162,000
21	07X95	X	07X	62	232.9	34	15	$74,400	$32,824	266.8	159	255	$1,116,000	$19,278,000
21	07X92	X	07X	46	242.2	26	11	$75,273	$31,846	267.9	187	257	$828,000	$20,106,000
21	07X98	X	07X	39	256.4	24	9	$78,000	$29,250	279.9	228	281	$702,000	$20,808,000
21	07X81	X	07X	52	247.7	29	12	$78,000	$32,276	276.8	202	273	$936,000	$21,744,000
													$77,688,00	

FIG.6

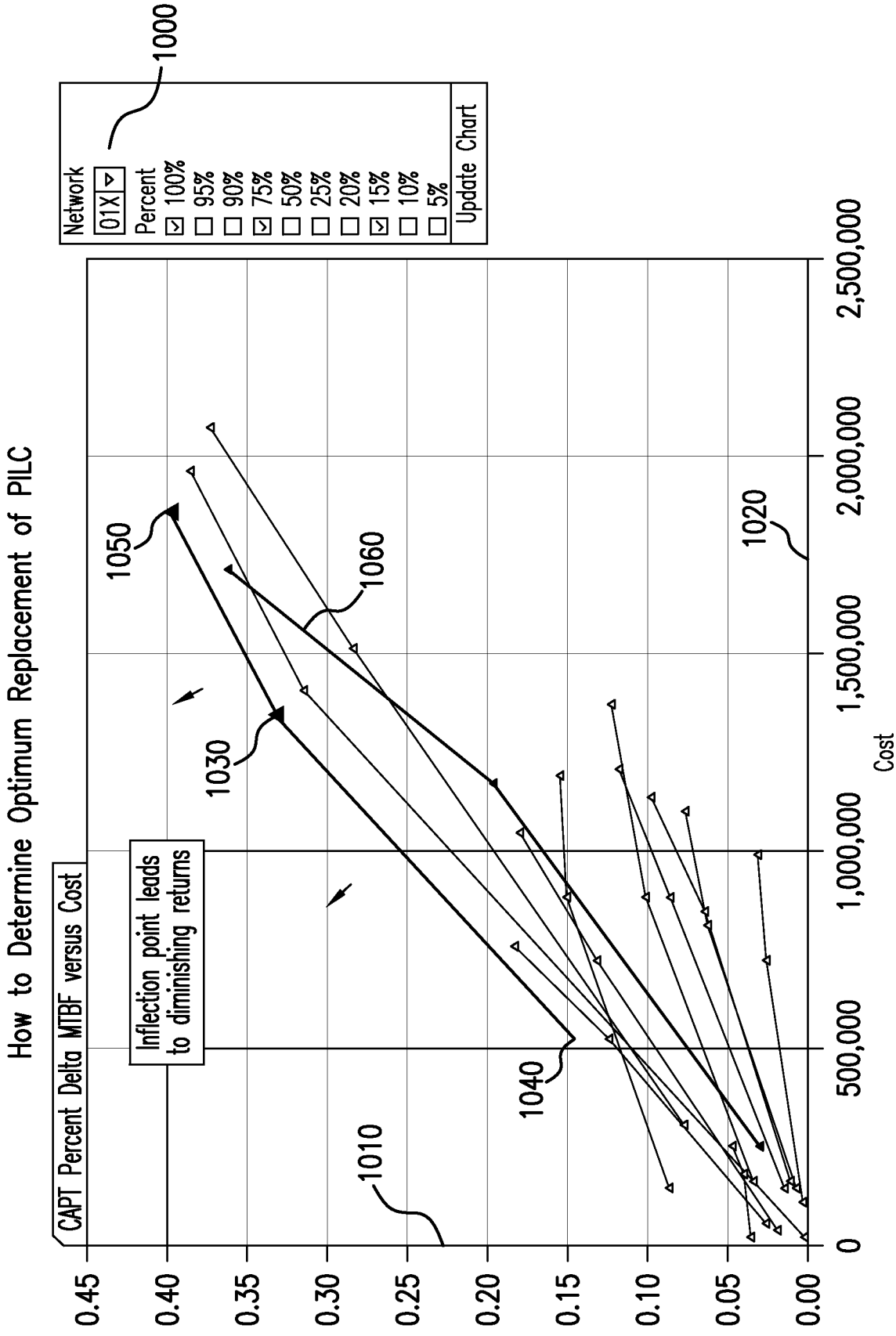

How to Determine Optimum Replacement of PILC
CAPT Percent Delta MTBF versus Cost
Inflection point leads to diminishing returns
Network
01X
Percent
100%
95%
90%
75%
50%
25%
20%
15%
10%
5%
Update Chart
1000
1010
1020
1030
1040
1050
1060
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
500,000
1,000,000
1,500,000
2,000,000
2,500,000
Cost

FIG. 7

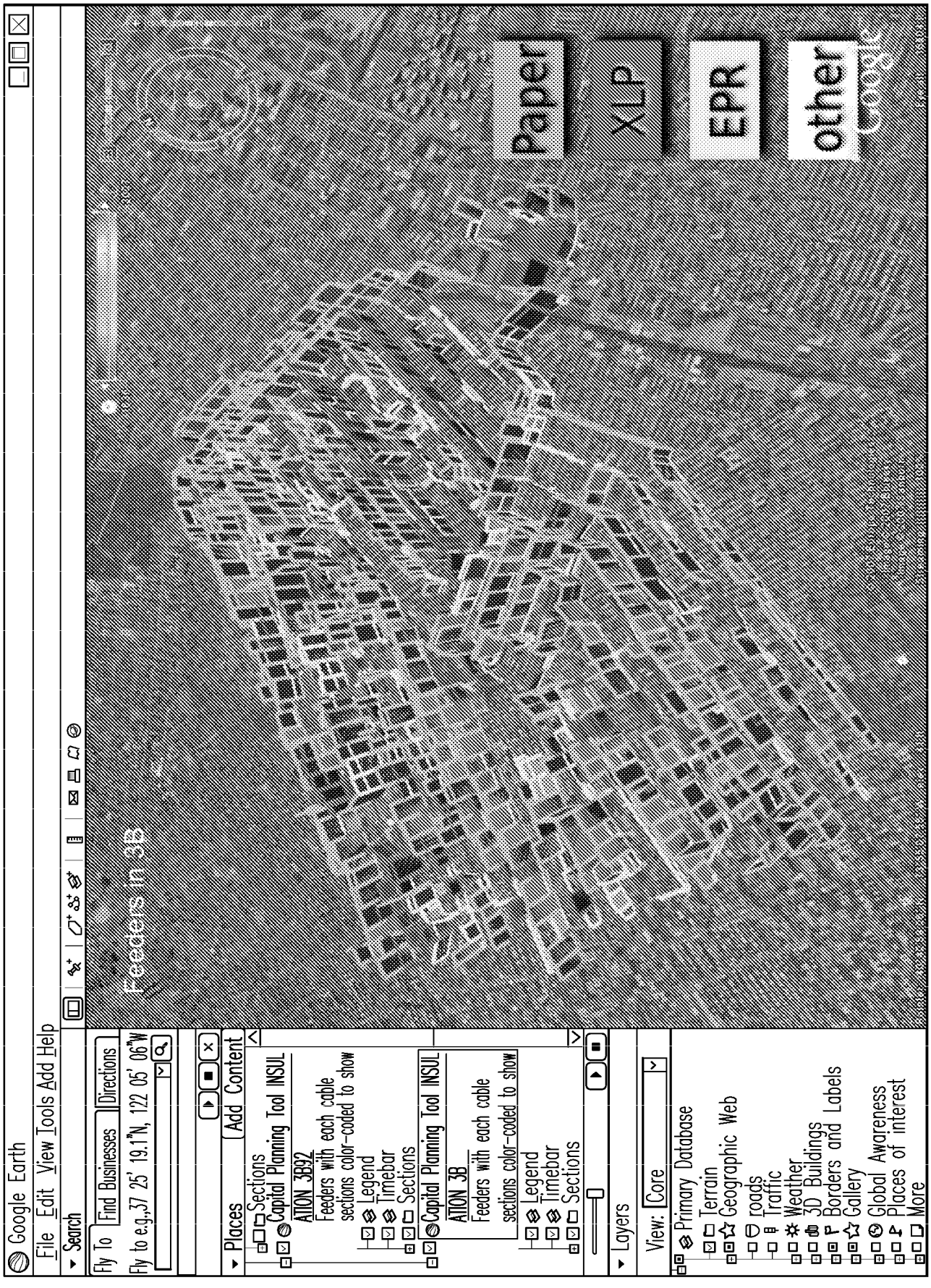

Google Earth
File Edit View Tools Add Help
Search
Fly To
Find Businesses
Directions
Fly to e.g., 37 25' 19.1"N, 122 05' 06"W
Places
Add Content
Sections
Capital Planning Tool INSUL
ATION 3B92
Feeders with each cable sections color-coded to show
Legend
Timebar
Sections
Capital Planning Tool INSUL
ATION 3B
Feeders with each cable sections color-coded to show
Legend
Timebar
Sections
Layers
View: Core
Primary Database
Terrain
Geographic Web
roads
Traffic
Weather
3D Buildings
Borders and Labels
Gallery
Global Awareness
Places of interest
More
Feeders in 3B
Paper
XLP
EPR
other

FIG. 8

Google Earth
File Edit View Tools Add Help
Search
Fly To
Find Businesses
Directions
Fly to e.g.,37 25' 19.1"N, 122 05' 06"W
Places
Add Content
Sections
Capital Planning Tool INSUL
ATION 3B92
Feeders with each cable sections color-coded to show
Legend
Timebar
Sections
Capital Planning Tool INSUL
ATION 3B
Feeders with each cable sections color-coded to show
Legend
Timebar
Sections
Layers
View: Core
Primary Database
Terrain
Geographic Web
roads
Traffic
Weather
3D Buildings
Borders and Labels
Gallery
Global Awareness
Places of interest
More
Feeder 3B92 ranked cable sections
Paper
XLP
EPR
other

FIG.9

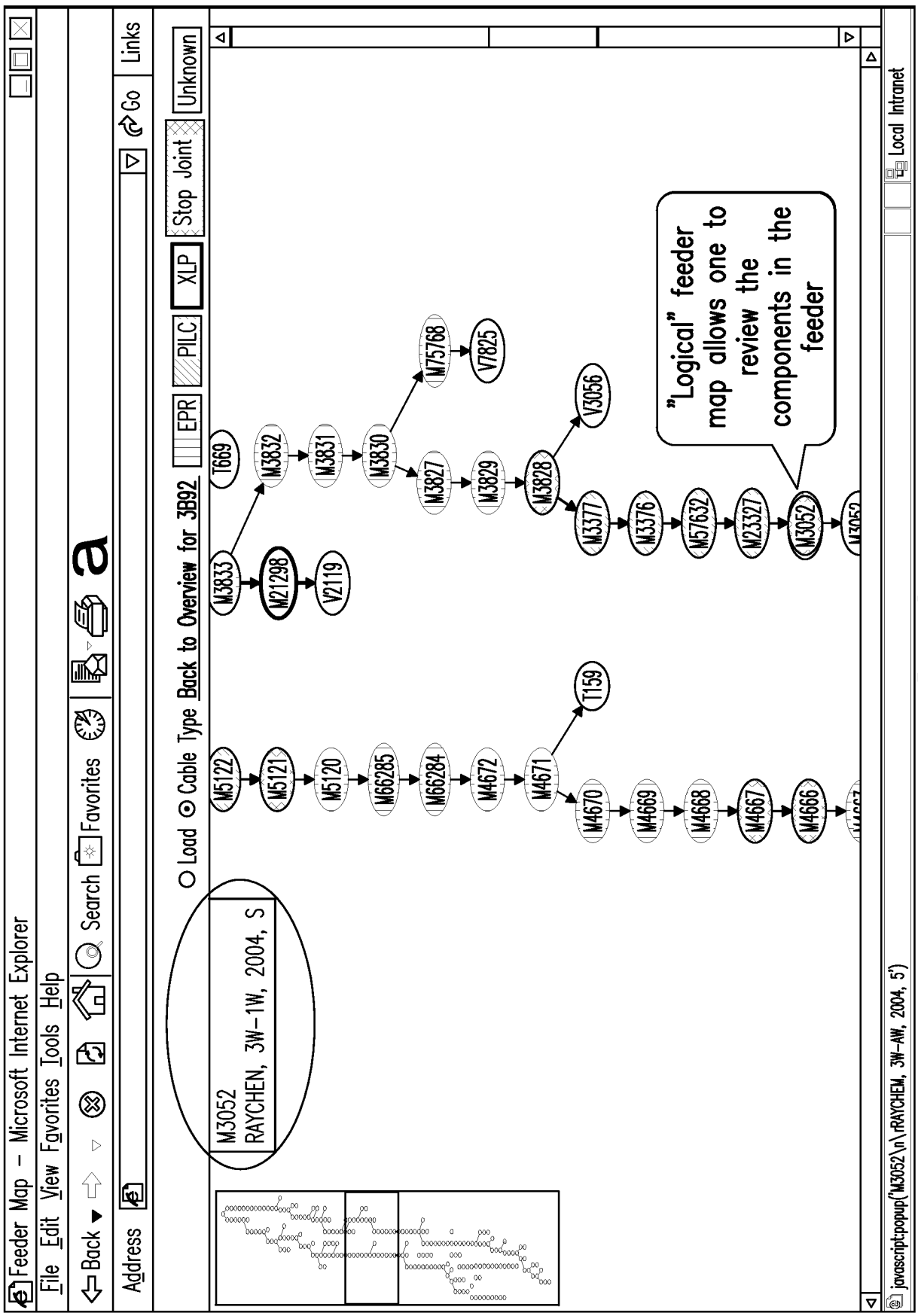

Feeder Map - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Links
Go
Load Cable Type Back to Overview for 3B92
EPR
PILC
XLP
Stop Joint
Unknown
M3052
RAYCHEN, 3W-1W, 2004, S
M5122
M5121
M5120
M66285
M66284
M4672
M4671
T159
M4670
M4669
M4668
M4667
M4666
M3833
M21298
V2119
T669
M3832
M3831
M3830
M3827
M3829
M3828
M75768
V7825
V3056
M3377
M3376
M57632
M23327
M3052
"Logical" feeder map allows one to review the components in the feeder
Local Intranet
javascriptpopup('M3052\n\RAYCHEM, 3W-AW, 2004, 5')

FIG.10

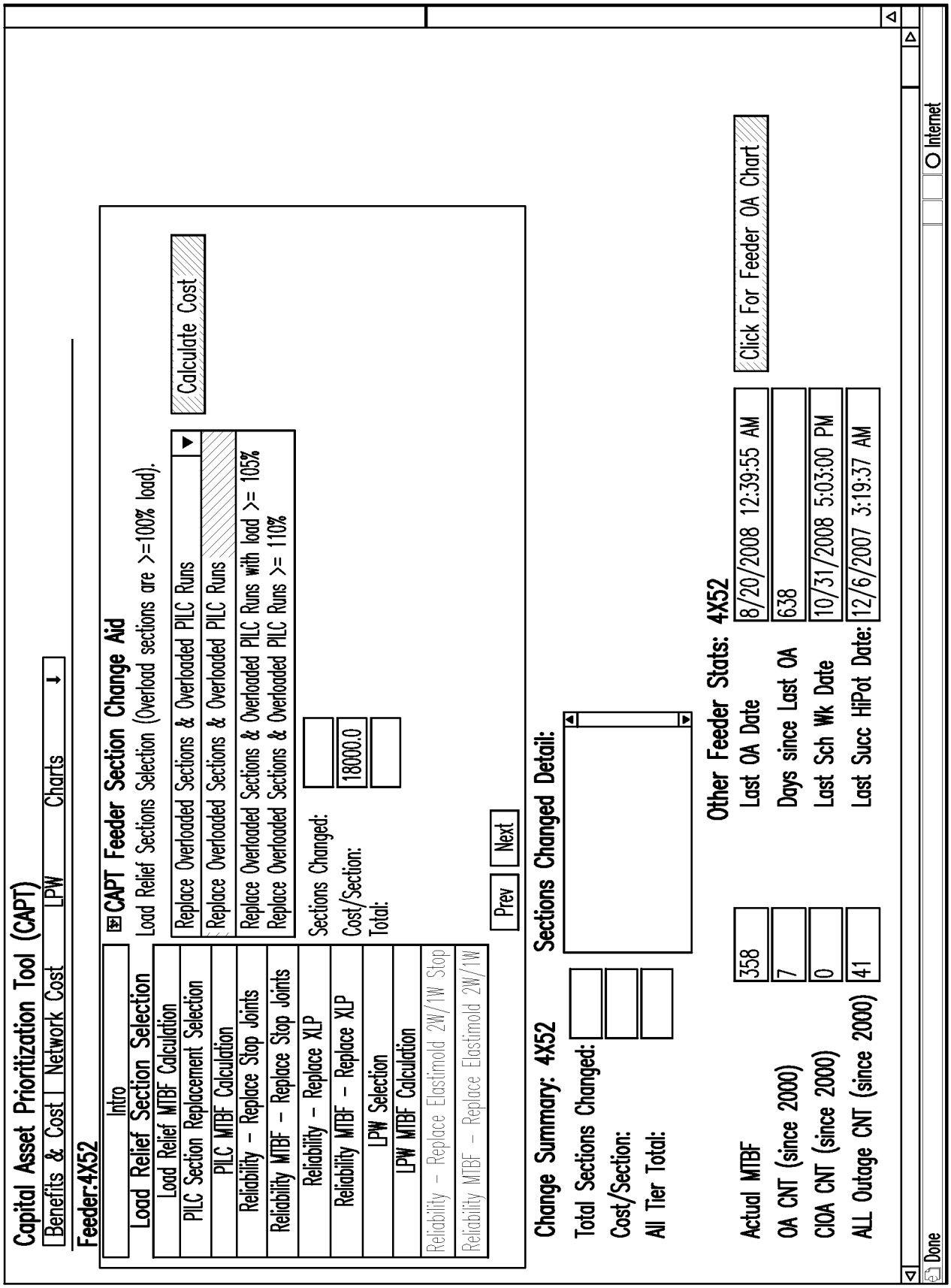

Capital Asset Prioritization Tool (CAPT)
Benefits & Cost
Network Cost
LPW
Charts
Feeder:4X52
Intro
Load Relief Section Selection
Load Relief MTBF Calculation
PILC Section Replacement Selection
PILC MTBF Calculation
Reliability – Replace Stop Joints
Reliability MTBF – Replace Stop Joints
Reliability – Replace XLP
Reliability MTBF – Replace XLP
LPW Selection
LPW MTBF Calculation
Reliability – Replace Elastimold 2W/1W Stop
Reliability MTBF – Replace Elastimold 2W/1W
CAPT Feeder Section Change Aid
Load Relief Sections Selection (Overload sections are >=100% load).
Replace Overloaded Sections & Overloaded PILC Runs
Replace Overloaded Sections & Overloaded PILC Runs
Replace Overloaded Sections & Overloaded PILC Runs with load >= 105%
Replace Overloaded Sections & Overloaded PILC Runs >= 110%
Calculate Cost
Sections Changed:
Cost/Section:
18000.0
Total:
Prev
Next
Change Summary: 4X52
Sections Changed Detail:
Total Sections Changed:
Cost/Section:
All Tier Total:
Other Feeder Stats: 4X52
Actual MTBF
358
OA CNT (since 2000)
7
CIOA CNT (since 2000)
0
ALL Outage CNT (since 2000)
41
Last OA Date
8/20/2008 12:39:55 AM
Days since Last OA
638
Last Sch Wk Date
10/31/2008 5:03:00 PM
Last Succ HiPot Date:
12/6/2007 3:19:37 AM
Click For Feeder OA Chart
Done
Internet

FIG. 11

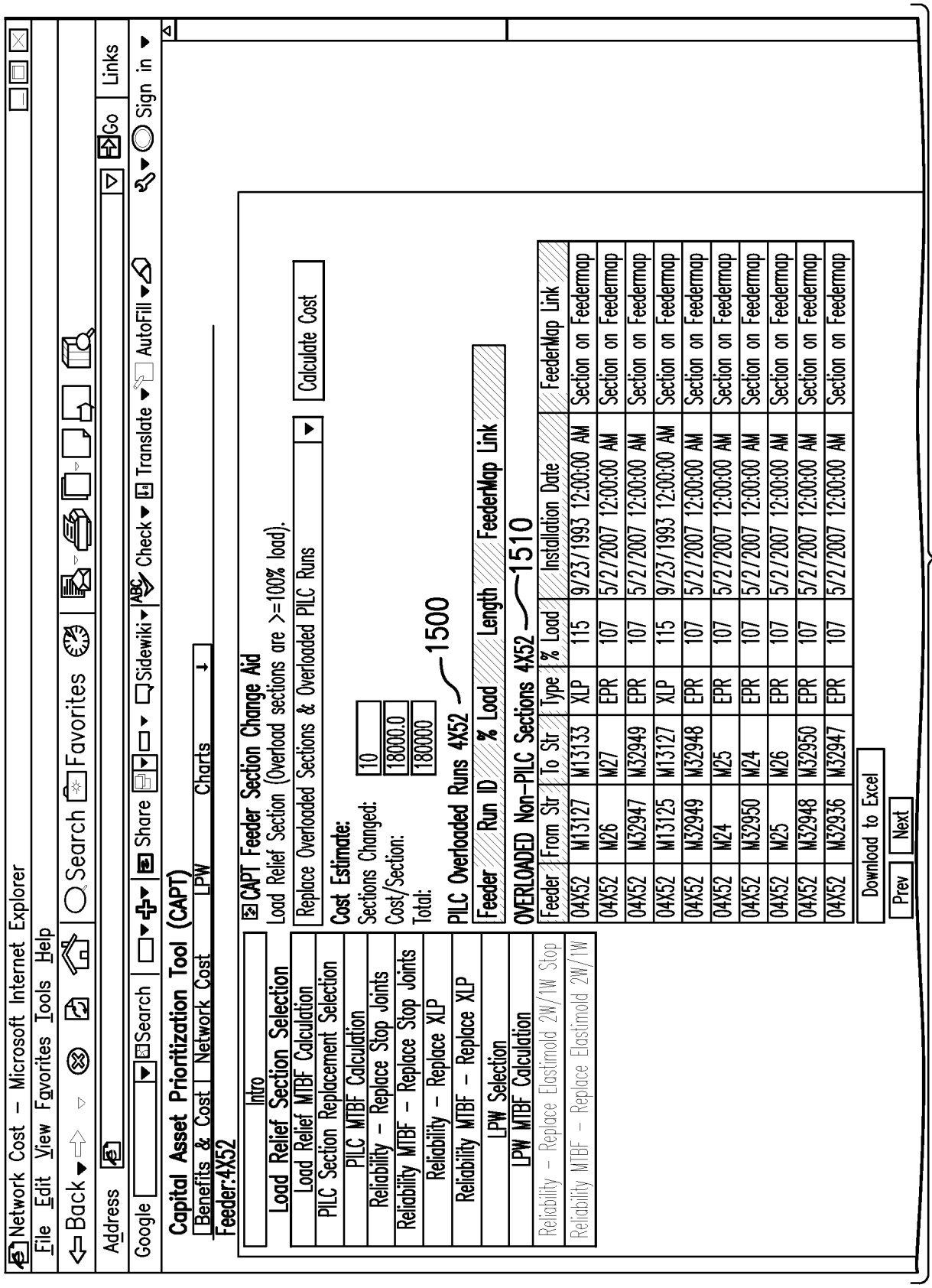

Network Cost - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Google Search Share Sidewiki Check Translate AutoFill Sign in
Capital Asset Prioritization Tool (CAPT)
Benefits & Cost | Network Cost | LPW | Charts
Feeder:4X52
Intro
Load Relief Section Selection
Load Relief MTBF Calculation
PILC Section Replacement Selection
PILC MTBF Calculation
Reliability - Replace Stop Joints
Reliability MTBF - Replace Stop Joints
Reliability - Replace XLP
Reliability MTBF - Replace XLP
LPW Selection
LPW MTBF Calculation
Reliability - Replace Elastimold 2W/1W Stop
Reliability MTBF - Replace Elastimold 2W/1W
CAPT Feeder Section Change Aid
Load Relief Section (Overload sections are >=100% load).
Replace Overloaded Sections & Overloaded PILC Runs
Calculate Cost
Cost Estimate:
Sections Changed: 10
Cost/Section: 18000.0
Total: 180000
1500
PILC Overloaded Runs 4X52
Feeder Run ID % Load Length FeederMap Link
OVERLOADED Non-PILC Sections 4X52
1510
Feeder From Str To Str Type % Load Installation Date FeederMap Link
04X52 M13127 M13133 XLP 115 9/23/1993 12:00:00 AM Section on Feedermap
04X52 M26 M27 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
04X52 M32947 M32949 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
04X52 M13125 M13127 XLP 115 9/23/1993 12:00:00 AM Section on Feedermap
04X52 M32949 M32948 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
04X52 M24 M25 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
04X52 M32950 M24 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
04X52 M25 M26 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
04X52 M32948 M32950 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
04X52 M32936 M32947 EPR 107 5/2/2007 12:00:00 AM Section on Feedermap
Download to Excel
Prev Next
Continued on FIG.12B

FIG.12A

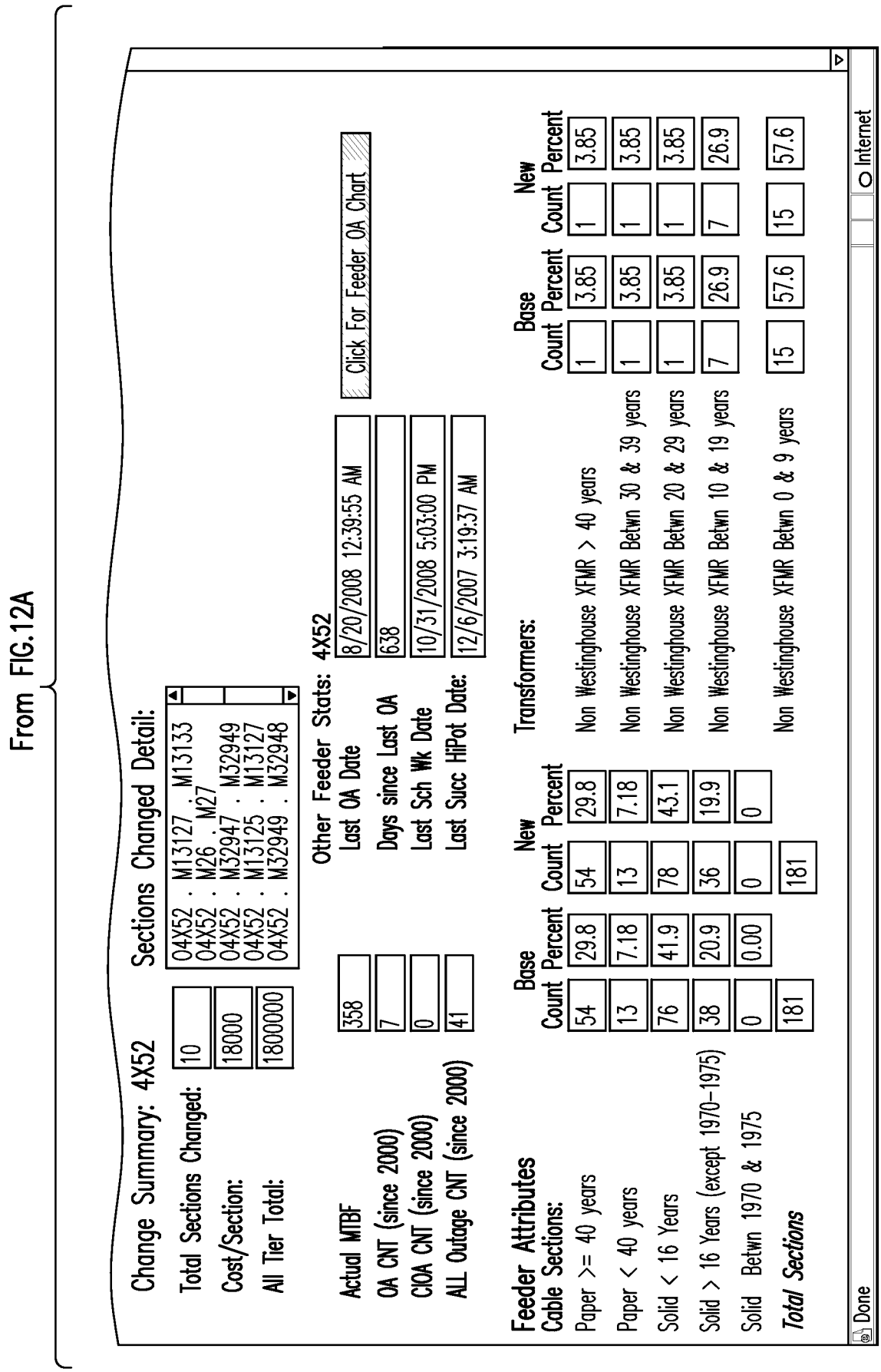
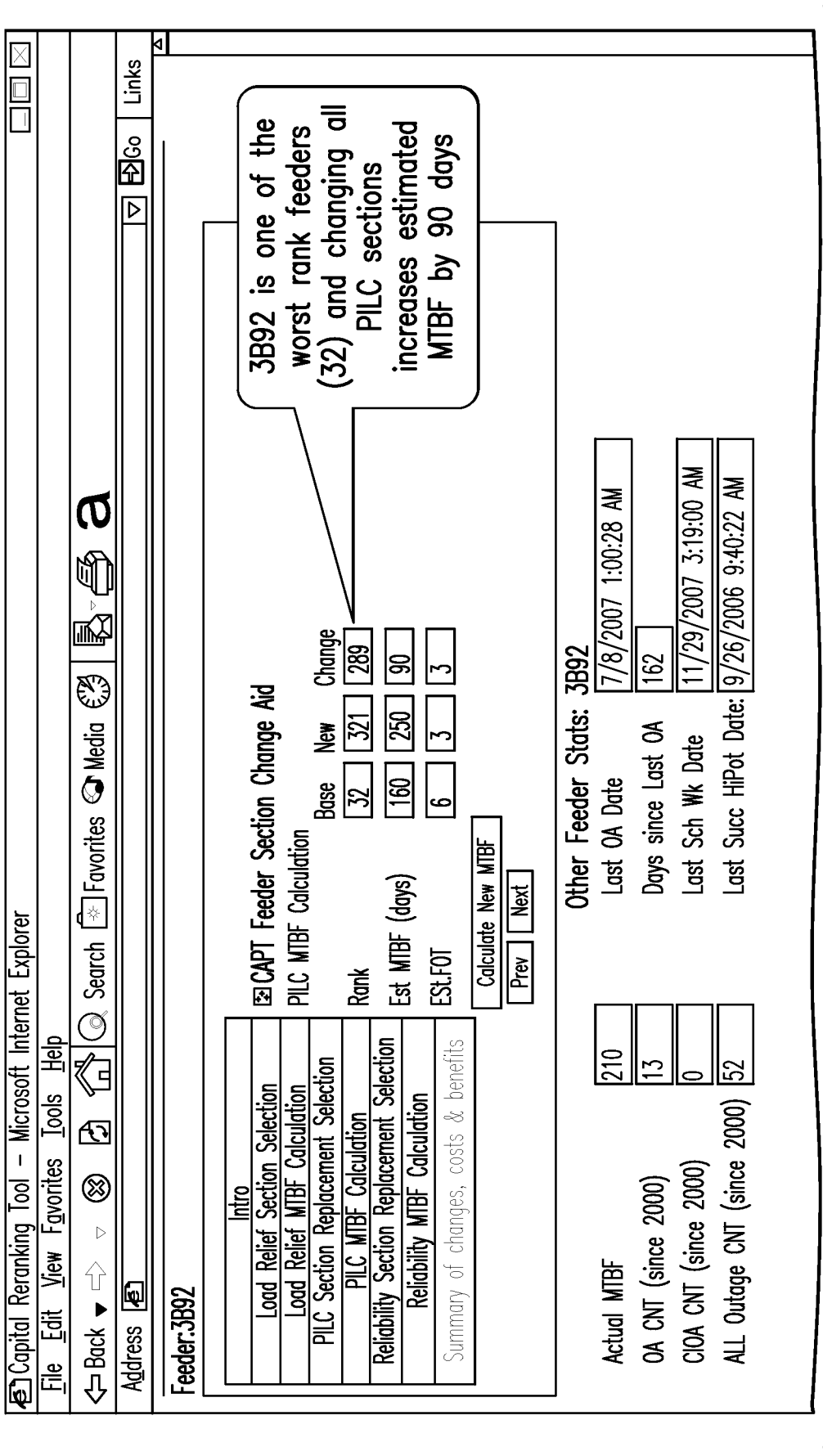

From FIG.12A
Change Summary: 4X52
Sections Changed Detail:
Total Sections Changed: 10
Cost/Section: 18000
All Tier Total: 1800000
04X52 . M13127 . M13133
04X52 . M26 . M27
04X52 . M32947 . M32949
04X52 . M13125 . M13127
04X52 . M32949 . M32948
Other Feeder Stats: 4X52
Actual MTBF 358
OA CNT (since 2000) 7
CIOA CNT (since 2000) 0
ALL Outage CNT (since 2000) 41
Last OA Date 8/20/2008 12:39:55 AM
Days since Last OA 638
Last Sch Wk Date 10/31/2008 5:03:00 PM
Last Succ HiPot Date: 12/6/2007 3:19:37 AM
Click For Feeder OA Chart
Feeder Attributes
Cable Sections:
Base Count Percent
New Count Percent
Transformers:
Base Count Percent
New Count Percent
Paper >= 40 years 54 29.8 54 29.8
Paper < 40 years 13 7.18 13 7.18
Solid < 16 Years 76 41.9 78 43.1
Solid > 16 Years (except 1970-1975) 38 20.9 36 19.9
Solid Betwn 1970 & 1975 0 0.00 0 0
Total Sections 181 181
Non Westinghouse XFMR > 40 years 1 3.85 1 3.85
Non Westinghouse XFMR Betwn 30 & 39 years 1 3.85 1 3.85
Non Westinghouse XFMR Betwn 20 & 29 years 1 3.85 1 3.85
Non Westinghouse XFMR Betwn 10 & 19 years 7 26.9 7 26.9
Non Westinghouse XFMR Betwn 0 & 9 years 15 57.6 15 57.6
Done
Internet

FIG.12B



Continued on FIG.13B

FIG.13A

From FIG.13A

Feeder Attributes

Cable Sections:	Base Count	Base Percent	New Count	New Percent
Paper >= 40 years	45	19.9	0	0
Paper < 40 years	10	4.42	0	0
Solid < 16 Years	131	57.9	186	82.3
Solid > 16 Years (except 1970–1975)	34	15.0	34	15.0
Solid Betwn 1970 & 1975	6	2.65	6	2.65
Total Sections	226		226	

Joints:	Base Count	Base Percent	New Count	New Percent
Elastimold 2W–1W Stop > 1993	0	0.00	0	0
Elastimold 2W–1W Stop <= 1993	0	0.00	0	0
Raychem 3W–1W Stop > 1993	20	10.5	0	0
Raychem 3W–1W Stop <= 1993	3	1.58	0	0
Other Stop Joints < 7 years	2	1.05	0	0
Other Stop Joints Betwn 7 & 12 years	3	1.58	0	0

Transformers:	Base Count	Base Percent	New Count	New Percent
Non Westinghouse XFMR > 40 years	1	2.33	1	2.33
Non Westinghouse XFMR Betwn 30 & 39 years	3	6.98	3	6.98
Non Westinghouse XFMR Betwn 20 & 29 years	3	7.00	3	7.00
Non Westinghouse XFMR Betwn 10 & 19 years	14	32.5	14	32.5
Non Westinghouse XFMR Betwn 0 & 9 years	19	44.1	19	44.1
Westinghouse XFMR > 40 years	1	2.33	1	2.33
Westinghouse XFMR Betwn 30 & 39 years	1	2.33	1	2.33
Westinghouse XFMR Betwn 20 & 29 years	0	0.00	0	0.00
Westinghouse XFMR Betwn 10 & 19 years	1	2.33	1	2.33
Westinghouse XFMR Betwn 0 & 9 years	0	0.00	0	0.00
Total XFMR	43		43	

Done | Local intranet

FIG.13B

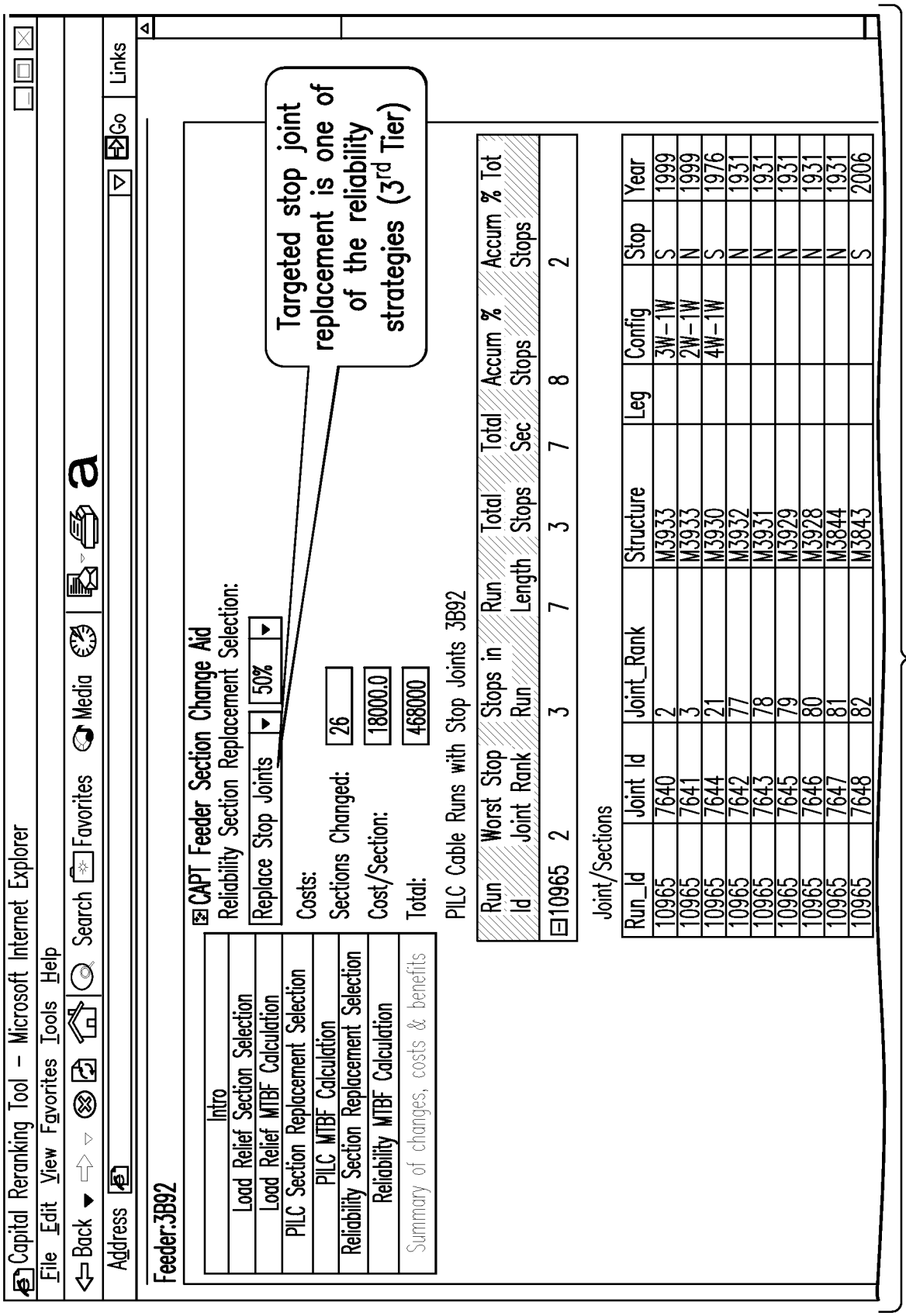

Capital Reranking Tool – Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links
Feeder:3B92
CAPT Feeder Section Change Aid
Reliability Section Replacement Selection:
Replace Stop Joints
50%
Intro
Load Relief Section Selection
Load Relief MTBF Calculation
PILC Section Replacement Selection
PILC MTBF Calculation
Reliability Section Replacement Selection
Reliability MTBF Calculation
Summary of changes, costs & benefits
Costs:
Sections Changed: 26
Cost/Section: 18000.0
Total: 468000
Targeted stop joint replacement is one of of the reliability strategies (3rd Tier)
PILC Cable Runs with Stop Joints 3B92
Run Id
Worst Stop Joint Rank
Stops in Run
Run Length
Total Stops
Total Sec
Accum % Stops
Accum % Tot Stops
10965 2 3 7 3 7 8 2
Joint/Sections
Run_Id | Joint Id | Joint_Rank | Structure | Leg | Config | Stop | Year
10965 | 7640 | 2 | M3933 | | 3W-1W | S | 1999
10965 | 7641 | 3 | M3933 | | 2W-1W | N | 1999
10965 | 7644 | 21 | M3930 | | 4W-1W | S | 1976
10965 | 7642 | 77 | M3932 | | | N | 1931
10965 | 7643 | 78 | M3931 | | | N | 1931
10965 | 7645 | 79 | M3929 | | | N | 1931
10965 | 7646 | 80 | M3928 | | | N | 1931
10965 | 7647 | 81 | M3844 | | | N | 1931
10965 | 7648 | 82 | M3843 | | | S | 2006
Continued on FIG.14B

FIG.14A

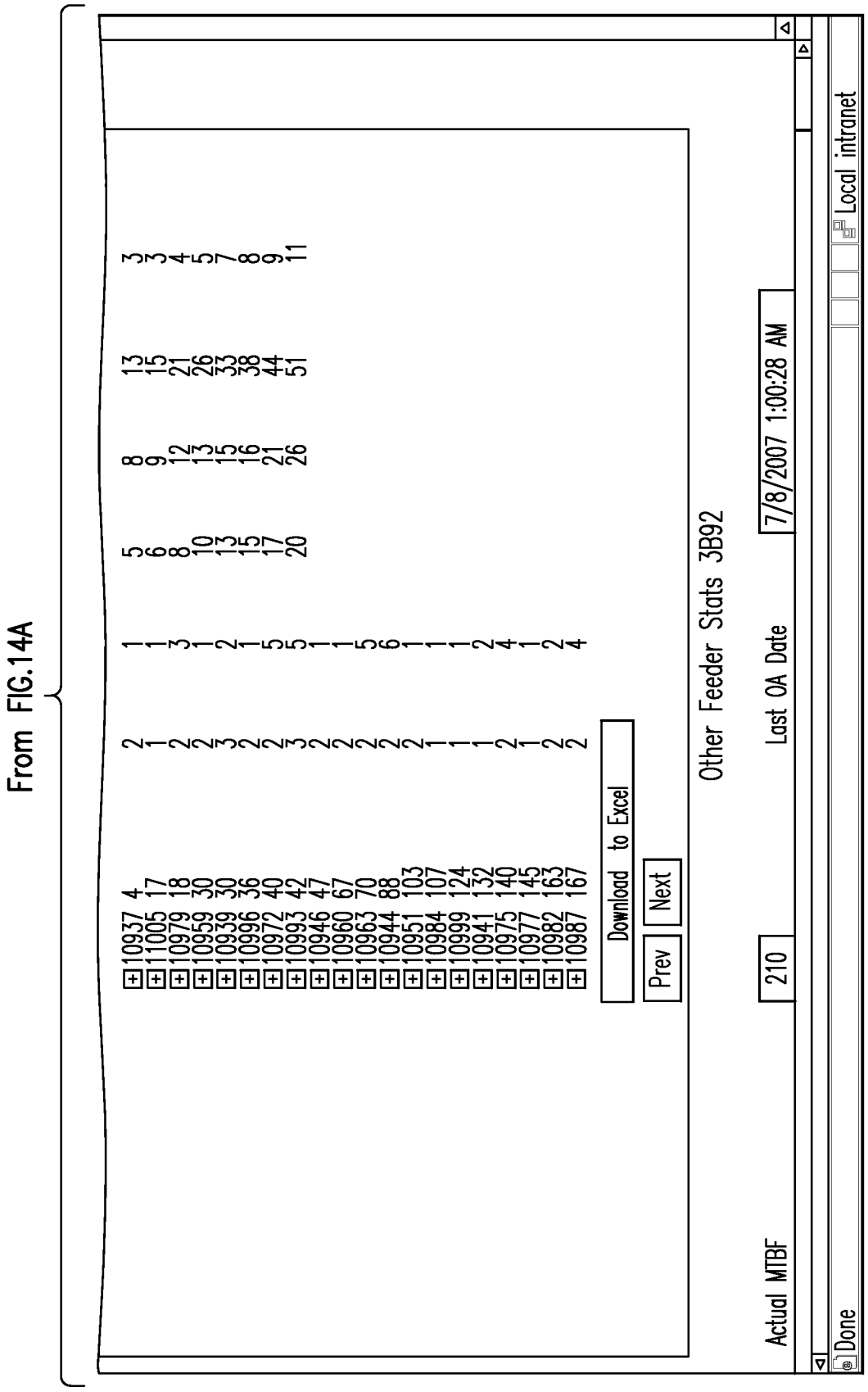

From FIG.14A
Other Feeder Stats 3B92
Download to Excel
Prev
Next
Actual MTBF
210
Last OA Date
7/8/2007 1:00:28 AM
Done
Local intranet

FIG.14B

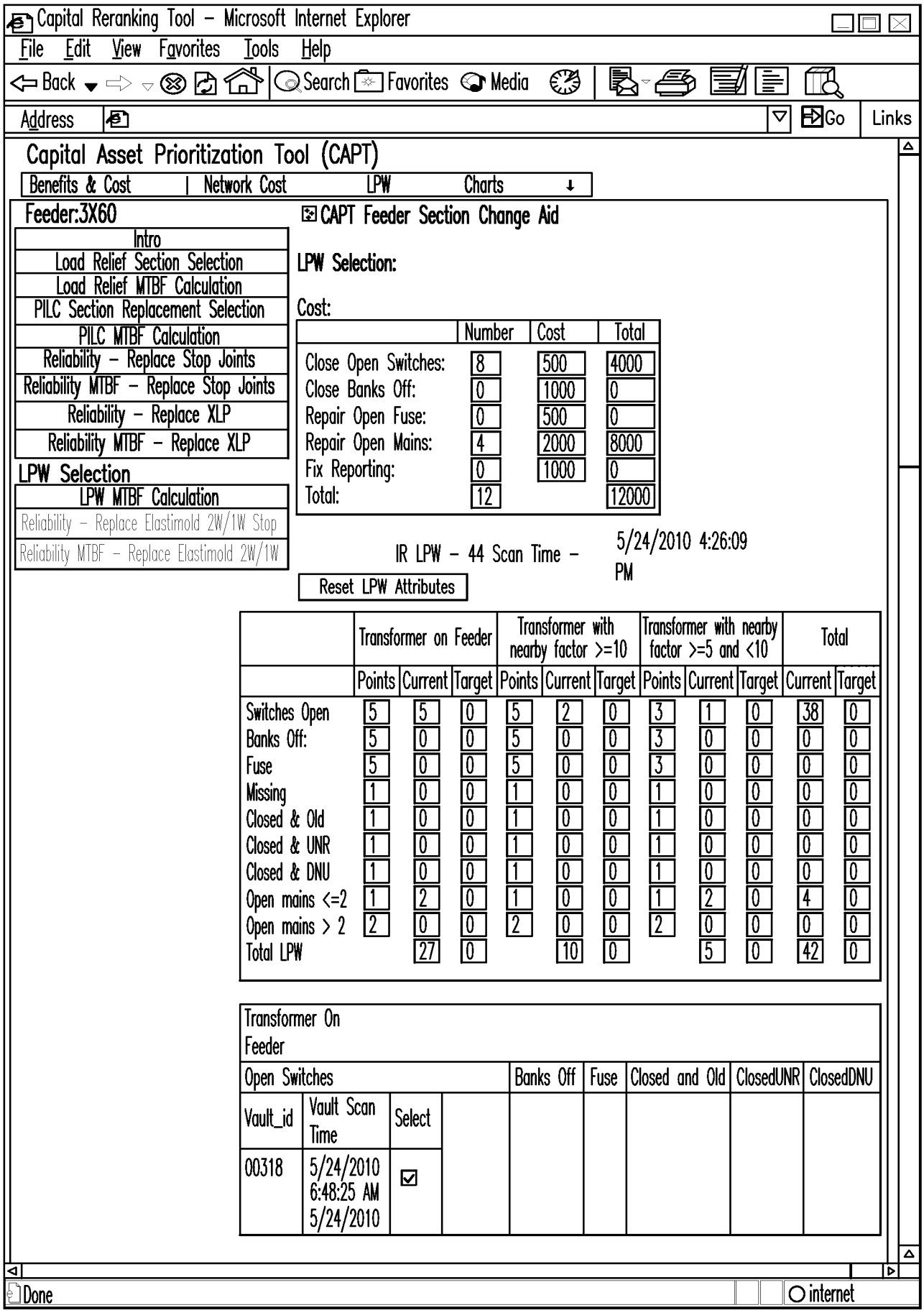

Capital Reranking Tool - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Links
Capital Asset Prioritization Tool (CAPT)
Benefits & Cost | Network Cost LPW Charts
Feeder:3X60
Intro
Load Relief Section Selection
Load Relief MTBF Calculation
PILC Section Replacement Selection
PILC MTBF Calculation
Reliability - Replace Stop Joints
Reliability MTBF - Replace Stop Joints
Reliability - Replace XLP
Reliability MTBF - Replace XLP
LPW Selection
LPW MTBF Calculation
Reliability - Replace Elastimold 2W/1W Stop
Reliability MTBF - Replace Elastimold 2W/1W
CAPT Feeder Section Change Aid
LPW Selection:
Cost:
Number Cost Total
Close Open Switches: 8 500 4000
Close Banks Off: 0 1000 0
Repair Open Fuse: 0 500 0
Repair Open Mains: 4 2000 8000
Fix Reporting: 0 1000 0
Total: 12 12000
IR LPW - 44 Scan Time - 5/24/2010 4:26:09 PM
Reset LPW Attributes
Transformer on Feeder
Transformer with nearby factor >=10
Transformer with nearby factor >=5 and <10
Total
Points Current Target Points Current Target Points Current Target Current Target
Switches Open 5 5 0 5 2 0 3 1 0 38 0
Banks Off: 5 0 0 5 0 0 3 0 0 0 0
Fuse 5 0 0 5 0 0 3 0 0 0 0
Missing 1 0 0 1 0 0 1 0 0 0 0
Closed & Old 1 0 0 1 0 0 1 0 0 0 0
Closed & UNR 1 0 0 1 0 0 1 0 0 0 0
Closed & DNU 1 0 0 1 0 0 1 0 0 0 0
Open mains <=2 1 2 0 1 0 0 1 2 0 4 0
Open mains > 2 2 0 0 2 0 0 2 0 0 0 0
Total LPW 27 0 10 0 5 0 42 0
Transformer On Feeder
Open Switches
Banks Off
Fuse
Closed and Old
ClosedUNR
ClosedDNU
Vault_id
Vault Scan Time
Select
00318
5/24/2010 6:48:25 AM 5/24/2010
Done
internet

FIG.15

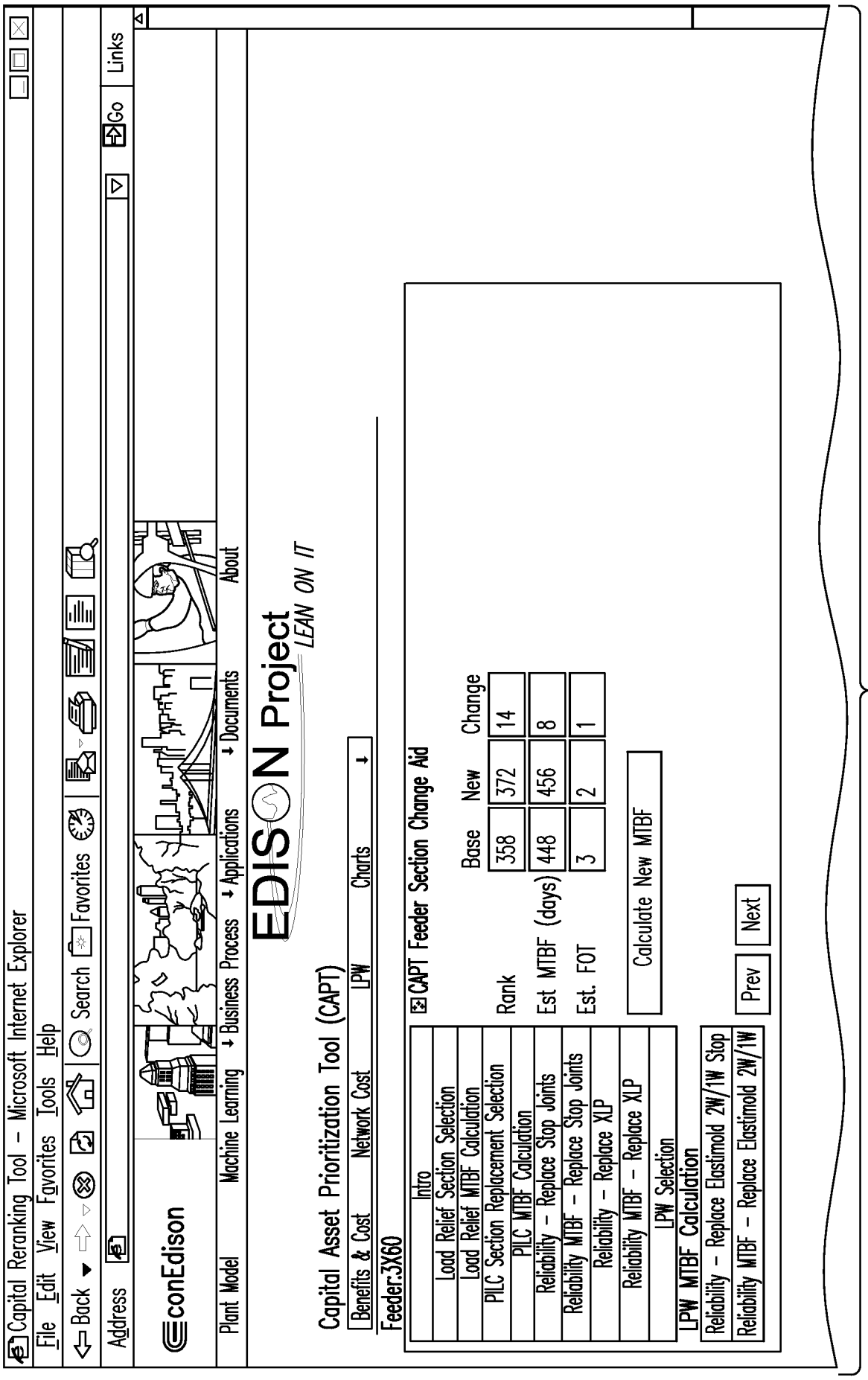

Capital Reranking Tool – Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
conEdison
Plant Model Machine Learning Business Process Applications Documents About
EDISON Project
LEAN ON IT
Capital Asset Prioritization Tool (CAPT)
Benefits & Cost Network Cost LPW Charts
Feeder:3X60
Intro
Load Relief Section Selection
Load Relief MTBF Calculation
PILC Section Replacement Selection
PILC MTBF Calculation
Reliability – Replace Stop Joints
Reliability MTBF – Replace Stop Joints
Reliability – Replace XLP
Reliability MTBF – Replace XLP
LPW Selection
LPW MTBF Calculation
Reliability – Replace Elastimold 2W/1W Stop
Reliability MTBF – Replace Elastimold 2W/1W
CAPT Feeder Section Change Aid
Base New Change
Rank 358 372 14
Est MTBF (days) 448 456 8
Est. FOT 3 2 1
Calculate New MTBF
Prev Next
Continued on FIG.16B

FIG.16A

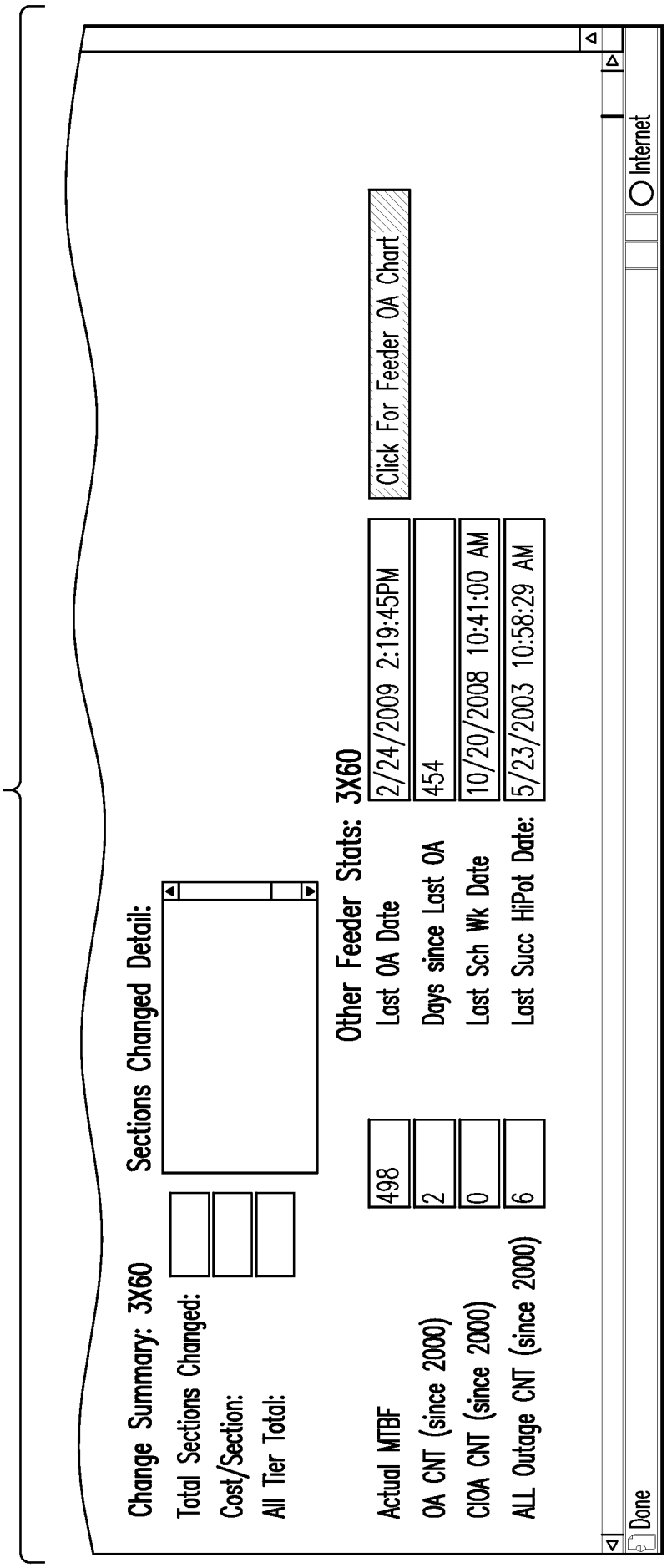

From FIG.16A
Change Summary: 3X60
Total Sections Changed:
Cost/Section:
All Tier Total:
Sections Changed Detail:
Other Feeder Stats: 3X60
Actual MTBF
OA CNT (since 2000)
CIOA CNT (since 2000)
ALL Outage CNT (since 2000)
498
2
0
6
Last OA Date
Days since Last OA
Last Sch Wk Date
Last Succ HiPot Date:
2/24/2009 2:19:45PM
454
10/20/2008 10:41:00 AM
5/23/2003 10:58:29 AM
Click For Feeder OA Chart
Done
Internet

FIG.16B

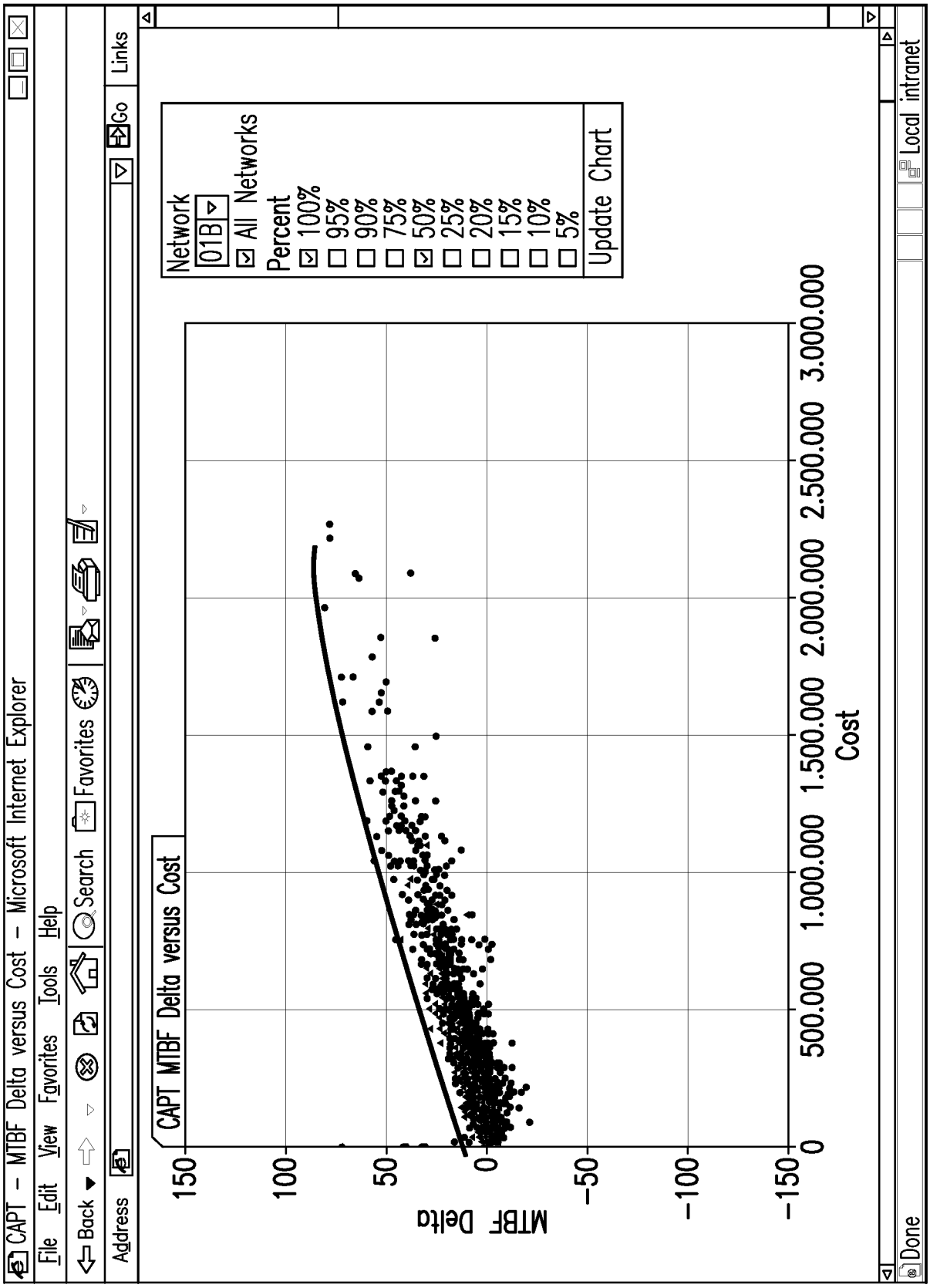

CAPT – MTBF Delta versus Cost – Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Links
Go
CAPT MTBF Delta versus Cost
Network
01B
All Networks
Percent
100%
95%
90%
75%
50%
25%
20%
15%
10%
5%
Update Chart
MTBF Delta
150
100
50
0
−50
−100
−150
0
500.000
1.000.000
1.500.000
2.000.000
2.500.000
3.000.000
Cost
Done
Local intranet

FIG.17

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 10/36717

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 17/00 (2010.01)
USPC - 705/7
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8): G06F 17/00 (2010.01)
USPC: 705/7

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 705/7, 412; 707/603; 700/286 (keyword limited; terms below)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWest; Google Scholar; Google Patents; FreePatentsOnline. Search terms used: asset planning, capital, improvement, infrastructure, effectiveness, database, metric measure measurement, status state, configure initialize parameter, display image screen user-interface UI graphic-user-interface ...

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2008/0319923 A1 (CASEY et al.) 25 December 2008 (25.12.2008) entire document, especially Abstract; Fig.4; para [0013], [0060], [0073], [0097], [0101], [0109], [0111], [0113], [0187]	1 - 20
Y	US 2004/0158772 A1 (PAN et al.) 12 August 2004 (12.08.2004) entire document, especially Abstract; para [0006], [0008], [0011], [0012], [0033], [0051], [0062], [0076], [0091], [0095], [0097], Table 1	1 - 20
Y	US 2008/0177678 A1 (DI MARTINI et al.) 24 July 2008 (24.07.2008) entire document, especially Abstract; para [0001], [0025], [0051], [0088], [0104], [0133], [0134], [0165]	1 - 20
A	US 2008/0126171 A1 (BALDWIN et al.) 29 May 2008 (29.05.2008) entire document	1 - 20

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 July 2010 (20.07.2010)	28 JUL 2010
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774