

Production Reporting On Microsoft SQL Server

Prepared For

MASON-GREY

Mason-Grey Corporation
400 Galleria Parkway, Suite 1500
Atlanta, Georgia 30339
Phone (678) 385-7470 / Fax (678) 385-7471
www.mason-grey.com

PRODUCTION REPORTING ON SQL SERVER

Purpose	4
Report System Flow	4
SQL Server	4
<i>DTS Package</i>	4
Nightly Report Data Mover – MES	4
Nightly Report Data Mover – Peak	6
Running DTS Packages	7
<i>Tables</i>	7
ss.grade_spec	7
ss.invent_audit	8
ss.jumbo	8
ss.sum_duration_day	8
ss.ReelWidth	8
<i>Views</i>	8
vwGross	8
vwGood	9
vwCull	11
vwTrim	13
vwResale	14
vwDowntime	15
<i>Stored Procedures</i>	15
spRptStdTonsByMachineByGradeSpec	15
spRptStdTonsByMachine	17
spRptActTonsByMachineByGradeSpec	18
spRptActTonsByMachine	19
spRptReelWidth	20
spGetAllReelWidths	20
spInsertReelWidth	21
spDeleteReelWidth	21
<i>User-defined Functions</i>	22
GMTDateToOVTimestamp	22
OVTimestampToGMTDate	22
GMTOffsetSeconds	22
IsDaylightTime	22
OVTimestampToEasternDate	23
SecondsToHHMM	23
Production Access Database	23
<i>Report Flow</i>	23
<i>Reports</i>	24
rptStdTonsByMachineByGradeSpec	24
rptStdTonsByMachine	24
rptActTonsByMachineByGradeSpec	25
rptActTonsByMachine	25
rptReelWidth	25

<i>rptStdTonsByMachineFactorSub</i>	25
<i>Queries</i>	25
<i>Tables</i>	26
tblOption	26
<i>Forms</i>	26
<i>Modules</i>	28
Configuration	28
<i>SQL Server</i>	28
MES	Error! Bookmark not defined.
Peak	29
SQL Server	29
<i>User Computer</i>	29
Appendix A	31

Purpose

The purpose of this document is to describe and document the flow and the components of the Atlanta Client production reporting system. The system currently supports four production reports.

Throughout this document, an “ov” prefix indicates a column or table from the MES database and an “ss” prefix indicates a column or table from the SQL Server database.

Report System Flow

See Appendix A for a high-level data flow diagram of the reporting system.

- Throughout the day, production data is stored in the MES system’s Informix database.
- Each morning at 4:00 am, the Nightly Report Data Mover – MES Data Transformation Services (DTS) package selects the previous day’s information and stores it in a SQL Server database so that reports and queries on the data will not impact the production system.
- A user opens the Production Reporting Access database and selects a report from the Production Reporting form
- The selected report dates are passed to a function which:
 - Issues a query that executes a SQL Server stored procedure via a “pass-through” query to retrieve the report data and store it in a local Access table
 - Opens the report, which uses the local Access table as its data source
 - Prints and/or displays the report depending on option settings

SQL Server

DTS Package

Every morning at 4:00 am, two SQL Server Agent jobs start DTS packages to move the previous day’s data from production databases to a SQL Server reporting database. One package is named “Nightly Report Data Mover – MES” and the other is “Nightly Report Data Mover – Peak.” This document addresses the MES data.

As of this writing, the reporting database is named ClientDataStore on the MACSRV3 SQL Server. See the section on Configuration for instructions on how to change the data location.

The schedule and job history of the packages can be viewed and modified in the SQL Server Enterprise Manager under Management->SQL Server Agent->Jobs. To view job history, right-click the job and select “View Job History...” from the menu. The packages can be run manually using the instructions in Running DTS Packages below.

The packages can be viewed and edited in the SQL Server Enterprise Manager under Data Transformation Services->Local Packages. Each package runs as a single database transaction – if a part of the package fails, then all the operations for that package are rolled back. However, each package is independent of the other – if one fails, the other will still try to run.

Nightly Report Data Mover – MES

Global Variables

Global variables can be accessed by editing the package and selecting Package->Properties from the menu bar and clicking the Global Variables tab.

This package has four global variables:

- **StartDate** – the start Eastern time zone date/time (inclusive) of the data to be extracted. The default value should be 1/1/1970.
- **EndDate** – the end Eastern time zone date/time (exclusive) of the data to be extracted. The default value should be 1/1/1970.
- **StartTS** – the start timestamp corresponding to the StartDate. This timestamp represents a GMT date because the timestamps in MES are in GMT. The default value is 0.
- **EndTS** – the end timestamp corresponding to the EndDate. This timestamp represents a GMT date because the timestamps in MES are in GMT. The default value is 0.

Connections

This package contains two database connection definitions. One is a connection to the MES Informix database and the other is to the SQL Server data store.

MES

The MES connection uses a System DSN (ODBC Connection) named MESAtlanta. This DSN must be defined on the SQL Server for the scheduled job to run correctly. It should be defined on any machine that uses Enterprise Manager to execute the package manually.

SQL Server

The SQL Server connection uses the Microsoft OLE DB Provider for SQL Server.

Tasks

The MES package consists of a sequential flow of tasks.

- **Set Date Range** – The first task is an ActiveX Script Task that determines the date range for this report run. If StartDate or EndDate are 1/1/1970, then this task will use the prior day (midnight to midnight) as the date range. This step ALWAYS calculates the timestamps based on the dates. There is a command line utility for running DTS packages (described below) that will take start and end dates as parameters. This step is written in VBScript and is commented in the code.
- **Copy table data** – There are multiple sets of connections and tasks to copy each table's data from MES to SQL Server. In general, these steps are:
 - A SQL task to remove any previous rows from SQL Server for the timeframe of the current run. Where appropriate, this step uses the StartTS and EndTS global variables as parameters for the delete query. To edit one of these steps, double-click it. The property dialog has the SQL statement and a Parameters button. If the SQL statement has "?" in it, then the Parameters button defines the global variables assigned to the question marks.
 - A Transform Data task to copy the data from the MES connection to the SQL Server connection. To view the data transformation, double-click the arrow between the two connections. The Source tab contains the query to select the data from the MES system and uses the same Parameters scheme as above. The Destination tab shows the destination table in SQL Server. The Transformation tab shows the relationship between the source columns and destination columns.
- **Reset the StartDate and EndDate global variables to 1/1/1970.** During development, it was very easy to accidentally save the package with the last run's dates. This step eliminates that problem.

Tables

The tables currently copied from MES are:

- **Grade_spec** – rows are copied if ov.ts_modified or ov.ts_created is in the date range of the current package execution. The value 2147483647 is used as ts_expires for new rows because that is the largest value for a timestamp and translates to a date sometime in 2038.

There are two data transformations for this table that are not straight copies. These may be viewed by editing the package and double-clicking the arrow between MES and SQL Server for the Copy grade_spec task and clicking the Transformations tab:

- DTSTransformation__10 uses the first 3 characters of ov.grade_spec to populate ss.paper_grade
 - DTSTransformation__11 is an ActiveX Script transformation to execute two “Lookups.” The lookup queries can be viewed on the Lookups tab. One query updates ss.ts_expires for the ss.grade_spec row from the previous time period and the other eliminates rows for the same ss.grade_spec and ss.ts_effective as inserted rows (to prevent duplicate rows in ss.grade_spec).
- Jumbo – gets rows for the package execution date range based on the timestamp value in ts_turned_up. Jumbo is also populated from jumbo_hist if the report dates are old enough. Jumbo_hist is populated with archived rows from jumbo.
 - Invent_audit – gets rows for the package execution date range based on the timestamp value in ts_wound or mx_timestamp
 - Sum_duration_day – gets rows for the package execution date range based on the timestamp in prod_date

Nightly Report Data Mover – Peak

This package copies data from the Peak Oracle database for the load and load_batch tables. These tables are used in the Shipping Access database, which has been modified on Client’s computer to access the SQL Server tables. These tables are not detailed in this document.

Global Variables

This package has two global variables:

- StartDate – the start Eastern time zone date/time (inclusive) of the data to be extracted. The default value should be 1/1/1970.
- EndDate – the end Eastern time zone date/time (exclusive) of the data to be extracted. The default value should be 1/1/1970.

Connections

This package contains two database connection definitions. One is a connection to the Peak Oracle database and the other is to the SQL Server data store.

peak

The peak connection uses the Microsoft ODBC Driver for Oracle.

SQL Server

The SQL Server connection uses the Microsoft OLE DB Provider for SQL Server.

Tasks

The MES package consists of a sequential flow of tasks.

- Set Date Range – The first task is an ActiveX Script Task that determines the date range for this report run. If StartDate or EndDate are 1/1/1970, then this task will use the prior day (midnight to midnight) as the date range. There is a command line utility for running DTS packages (described below) that will take start and end dates as parameters. This step is written in VBScript and is commented in the code.
- Copy table data – There are multiple sets of connections and tasks to copy each table’s data from peak to SQL Server. These steps are:
 - A SQL task to remove any previous rows from SQL Server for the timeframe of the current run. Where appropriate, this step uses the StartDate and EndDate global variables as parameters for the delete query. To edit one of these steps, double-click it. The property dialog has the SQL statement and a Parameters button. If the SQL statement has “?” in it, then the Parameters button defines the global variables assigned to the question marks.

- A Transform Data task to copy the data from the peak connection to the SQL Server connection. To view the data transformation, double-click the arrow between the two connections. The Source tab contains the query to select the data from the peak system and uses the same Parameters scheme as above. The Destination tab shows the destination table in SQL Server. The Transformation tab shows the relationship between the source columns and destination columns.
- Reset the StartDate and EndDate global variables to 1/1/1970. During development, it was very easy to accidentally save the package with the last run's dates. This step eliminates that problem.

Tables

The tables currently copied from peak are:

- load – gets rows based on the date value in dt_stamp
- load_batch – gets rows based on the date value in dt_stamp

Running DTS Packages

In order to run the DTS packages from a remote machine, that machine must have the SQL Server Client Utilities installed.

Running From The DTS Designer UI

To execute a package manually from the DTS Designer, edit the package from SQL Server Enterprise Manager->Data Transformation Services->Local Packages. Select Package->Execute from the menu bar.

Running From the dtsrun command-line utility

The dtsrun command-line utility can be used to pass in dates other than the default. To run the package for 9/5/02, follow the following steps:

- Open a command prompt window
- Type: dtsrun /S MACSRV3 /E /N "Nightly Report Data Mover - MES" /A "StartDate:7=9/5/2002 12:00 am" /A "EndDate:7=9/6/2002 12:00 am"

Hit Enter

Tables

The tables stored in SQL Server are not exact copies of the source tables in MES. Many columns are not copied to conserve space and “keep it simple” and some columns have been added to facilitate lookups in views and stored procedures.

ss.grade_spec

This table contains data about the grade classifications of paper being produced. It is mainly used as the source for the standard basis weight for a grade spec. This table differs from the ov.grade_spec table in that it maintains historic basis weight values for a grade spec and contains a derived column based on the grade.

Historic Values

The ov.bwt_sold value for a grade_spec periodically changes in the MES system without recording the historic value. For a reporting system, this means that if a report were run for 1/1/02 on 1/2/02 and 1/9/02, the report would show different results if the ov.bwt_sold value were change in between the times the report was run.

The DTS package uses the existing ov.ts_modified and ov.ts_created columns to determine whether to insert a new row into ss.grade_spec for newly added or changed rows in ov.grade_spec. The ss.ts_effective and ss.ts_expires columns in ss.grade_spec are used to record the period of time in which the values in a particular row are valid. For example, the following shows part of two rows in

ss.grade_spec representing the "lifetime" of ss.grade_spec (the column) 974018. At timestamp 1033905600, the ss.bwt_sold value was changed. When reports are run that use the ss.bwt_sold value to calculate standard amounts, the timestamp of the event is joined to the correct time period of ss.grade_spec.

grade_spec	paper_grade	bwt_sold	ts_effective	ts_expires
974018	974	74.0000	0	1033905600
974018	974	74.5000	1033905600	2147483647

Paper Grade

The above table also contains an ss.paper_grade column that is not present in MES. This column is simply derived from the first three characters of ss.grade_spec during the DTS package execution. All of the SQL statements used to run the production reports use different values depending on the paper grade. Rather than incur the overhead of parsing the left three characters for each row every time the queries are run, the value is figured out ahead of time and then used directly.

ss.invent_audit

The ss.invent_audit table represents events that take place in the production of paper rolls. In MES, a row is inserted for each action that takes place. Only a small number of columns are copied to SQL Server from MES by the DTS package.

ss.jumbo

Each row in the ss.jumbo table represents a "jumbo roll." New rows are not added for each action that takes place. Rather, values are set for various columns as events happen. Only a small number of columns are copied to SQL Server from MES by the DTS package.

ss.sum_duration_day

The ss.sum_duration_day table is populated from an MES summary table (ov.sum_duration_day) during the DTS package execution. This table represents machine downtime.

ss.ReelWidth

ss.ReelWidth contains rows representing the historic reel widths by grade spec used during reporting. Like ss.grade_spec, the reel width values change periodically and must be recorded. ss.ts_effective and ss.ts_expires represent the valid time period for rows in this table. ss.ReelWidth is not populated by the DTS package. The Production Report Access database contains a Reel Width administration form where a user can manage reel widths and their time periods.

Views

In SQL Server, a view is a virtual table representing the data in one or more tables. The reporting system uses views to gather distributed data into a form that the report stored procedures can more easily use. The views are used as the basis for report stored procedures.

In all of the views below, [factor] represents either 72000000 (12 * 2000 * 3000) or 24000000 (12 * 2000 * 1000). 72000000 is used when the paper grade is 110, 115 or 320 and 24000000 is used when it is not. Most of the queries are in pairs where the only difference between Query x and Query x+1 is that [factor] is different because of the paper grade.

vwGross

This view unions two queries to combine information from the jumbo, grade_spec and ReelWidth tables to calculate actual and standard "gross" values.

Columns

- Prod_mach_id – The machine id associated with the jumbo roll. This column is included so queries that use vwGross can be grouped.
- Grade_spec – The grade_spec of the jumbo roll. This column is included so queries that use vwGross can be grouped.
- ActGross – The actual gross value for each row in jumbo and jumbo_hist as calculated by $(\text{length_turnup} * \text{bsw_act} * \text{stdwidth}) / [\text{factor}]$. Length_turnup and bsw_act are from the jumbo or jumbo_hist table and stdwidth is from the ReelWidth table.
- StdGross – The gross value based on a standard basis weight for the grade spec. This value is calculated by $(\text{length_turnup} * \text{bwt_sold} * \text{stdwidth}) / [\text{factor}]$. Length_turnup is from the jumbo or jumbo_hist table, bwt_sold is from the grade_spec table and stdwidth is from the ReelWidth table.
- Ts_turned_up – The timestamp when the jumbo roll was turned up. This column is included so queries that use vwGross can be filtered for report dates.

Queries

- Query 1 – Selects information from jumbo where the grade_spec.paper_grade is 110, 115 or 320. Grade_spec and ReelWidth are joined on jumbo.ts_turned_up within the effective range for their pertinent historic values. ReelWidth is also joined on the grade_spec column.
- Query 2 – Selects information from jumbo where the grade_spec.paper_grade is NOT 110, 115 or 320. Grade_spec and ReelWidth are joined on jumbo.ts_turned_up within the effective range for their pertinent historic values. ReelWidth is also joined on the grade_spec column.

vwGood

This view unions six queries to combine information from the invent_audit and grade_spec tables to calculate actual and standard “good” values.

Columns

- Prod_mach_id – The machine id associated with the invent_audit row. This column is included so queries that use vwGood can be grouped.
- Grade_spec – The grade_spec of the roll. This column is included so queries that use vwGood can be grouped.
- ActGood – The sum of actual good values for rows in invent_audit grouped by prod_machine_id, grade_spec and mx_timestamp as calculated by $(\text{length_lineal} * \text{bsw_act} * \text{width_act}) / [\text{factor}]$. Length_lineal, bsw_act and width_act are from the invent_audit table.
- StdGood – The sum of good values based on a standard basis weight for the grade spec, grouped by prod_machine_id, grade_spec and mx_timestamp. This value is calculated by $(\text{length_lineal} * \text{bsw_act} * \text{width_act}) / [\text{factor}]$. Length_lineal and width_act are from the invent_audit table and bwt_sold is from the grade_spec table.
- BwtAct – The sum of bsw_act from invent_audit grouped by prod_machine_id, grade_spec and mx_timestamp. This value will be used by the report stored procedure to calculate actual basis weight for an entire grade spec.
- BwtStd – The sum of bwt_sold from grade_spec grouped by prod_machine_id, grade_spec and mx_timestamp. This value will be used by the report stored procedure to calculate standard basis weight for an entire grade spec. It is necessary to sum this data due to historic grade_spec values, since there could be no “one” standard basis weight over a report period.
- Rows – The count of rows that grouped together. This count will be used to average the basis weights above.
- Mx_timestamp – The timestamp of the source invent_audit row. This column is included so queries that use vwGood can be filtered for report dates.

Queries

- Query 1 – Selects information from `invent_audit` where:
 - the `grade_spec.paper_grade` is 110, 115 or 320
 - the action code is 'TURNUP'
 - the status is 'G'
 - the `grade_spec` is ≥ 100000 and ≤ 990000

`Grade_spec` is joined on `invent_audit.mx_timestamp` within the effective range for their pertinent historic values.
- Query 2 – Selects information from `invent_audit` where:
 - the `grade_spec.paper_grade` is NOT 110, 115 or 320
 - the action code is 'TURNUP'
 - the status is 'G'
 - the `grade_spec` is ≥ 100000 and ≤ 990000

`Grade_spec` is joined on `invent_audit.mx_timestamp` within the effective range for their pertinent historic values.
- Query 3 – Selects information from `invent_audit` where:
 - the `grade_spec.paper_grade` is 110, 115 or 320
 - the action code is NOT 'UNTURNSUP' or 'STATBEFORE'
 - the status is 'G'
 - the `grade_spec` is ≥ 100000 and ≤ 990000
 - `mx_station` is 'CHECKER' or 'ROLLSTATUS'

`Grade_spec` is joined on `invent_audit.mx_timestamp` within the effective range for their pertinent historic values.
- Query 4 – Selects information from `invent_audit` where:
 - the `grade_spec.paper_grade` is NOT 110, 115 or 320
 - the action code is NOT 'UNTURNSUP' or 'STATBEFORE'
 - the status is 'G'
 - the `grade_spec` is ≥ 100000 and ≤ 990000
 - `mx_station` is 'CHECKER' or 'ROLLSTATUS'

`Grade_spec` is joined on `invent_audit.mx_timestamp` within the effective range for their pertinent historic values.
- Query 5 – This query backs out calculations when rolls were unturned up by multiplying the `ActGood` and `StdGood` values by -1 . Selects information from `invent_audit` where:
 - the `grade_spec.paper_grade` is 110, 115 or 320
 - the action code is 'UNTURNSUP' or 'STATBEFORE'
 - the status is 'G'
 - the `grade_spec` is ≥ 100000 and ≤ 990000
 - `mx_station` is 'CHECKER' or 'ROLLSTATUS'

`Grade_spec` is joined on `invent_audit.mx_timestamp` within the effective range for their pertinent historic values.
- Query 6 – This query backs out calculations when rolls were unturned up by multiplying the `ActGood` and `StdGood` values by -1 . Selects information from `invent_audit` where:
 - the `grade_spec.paper_grade` is NOT 110, 115 or 320
 - the action code is 'UNTURNSUP' or 'STATBEFORE'
 - the status is 'G'
 - the `grade_spec` is ≥ 100000 and ≤ 990000
 - `mx_station` is 'CHECKER' or 'ROLLSTATUS'

`Grade_spec` is joined on `invent_audit.mx_timestamp` within the effective range for their pertinent historic values.

vwCull

This view unions 12 queries to combine information from the invent_audit and grade_spec tables to calculate actual and standard “cull” values.

Columns

- Prod_mach_id – The machine id associated with the invent_audit row. This column is included so queries that use vwCull can be grouped.
- Grade_spec – The grade_spec of the roll. This column is included so queries that use vwCull can be grouped.
- ActCull – The sum of actual cull values for rows in invent_audit grouped by prod_machine_id, grade_spec and mx_timestamp as calculated by $(\text{length_lineal} * \text{bsw_act} * \text{width_act}) / [\text{factor}]$. Length_lineal, bsw_act and width_act are from the invent_audit table.
- StdCull – The sum of cull values based on a standard basis weight for the grade spec, grouped by prod_machine_id, grade_spec and mx_timestamp. This value is calculated by $(\text{length_lineal} * \text{bsw_act} * \text{width_act}) / [\text{factor}]$. Length_lineal and width_act are from the invent_audit table and bwt_sold is from the grade_spec table.
- Mx_timestamp – The timestamp of the source invent_audit row. This column is included so queries that use vwCull can be filtered for report dates.

Queries

- Query 1 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is ‘TURNUP’
 - (the status is NOT ‘G’ and the grade_spec is ≥ 100000 and ≤ 990000)
OR the grade_spec is > 990000Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 2 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is ‘TURNUP’
 - (the status is NOT ‘G’ and the grade_spec is ≥ 100000 and ≤ 990000)
OR the grade_spec is > 990000Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 3 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is ‘TURNUP’
 - the grade_spec is < 100000Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 4 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is ‘TURNUP’
 - the grade_spec is < 100000Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 5 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is NOT ‘UNTURNUP’ or ‘STATBEFORE’
 - the mx_station is ‘CHECKER’ or ‘ROLLSTATUS’
 - (the status is NOT ‘G’ and the grade_spec is ≥ 100000 and ≤ 990000)
OR the grade_spec is > 990000

Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

- Query 6 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is NOT 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - (the status is NOT 'G' and the grade_spec is >= 100000 and <= 990000)
OR the grade_spec is > 990000

Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

- Query 7 – This query backs out calculations when rolls were unturned up by multiplying the ActCull and StdCull values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - (the status is NOT 'G' and the grade_spec is >= 100000 and <= 990000)
OR the grade_spec is > 990000

Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

- Query 8 – This query backs out calculations when rolls were unturned up by multiplying the ActCull and StdCull values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - (the status is NOT 'G' and the grade_spec is >= 100000 and <= 990000)
OR the grade_spec is > 990000

Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

- Query 9 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is NOT 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - the grade_spec is < 100000

Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

- Query 10 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is NOT 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - the grade_spec is < 100000

Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

- Query 11 – This query backs out calculations when rolls were unturned up by multiplying the ActCull and StdCull values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - the grade_spec is < 100000

Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

- Query 12 – This query backs out calculations when rolls were unturned up by multiplying the ActCull and StdCull values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320

- the action code is 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - the grade_spec is < 100000
- Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

vwTrim

This view unions four queries to combine information from the invent_audit, grade_spec and ReelWidth tables to calculate actual and standard "trim" values.

Columns

- Prod_mach_id – The machine id associated with the invent_audit row. This column is included so queries that use vwTrim can be grouped.
- Grade_spec_made – The grade_spec_made of the roll. This column is included so queries that use vwTrim can be grouped.
- ActTrim – The actual trim values for rows in invent_audit grouped by prod_machine_id, grade_spec_made, ts_wound and mx_action_code as calculated by $(\text{avg}(\text{length_lineal}) * \text{avg}(\text{bsw_act}) * (\text{avg}(\text{stdwidth}) - \text{sum}(\text{width_ord}))) / [\text{factor}]$. Length_lineal, bsw_act and width_ord are from the invent_audit table. stdwidth is from the ReelWidth table.

The idea is to multiply the trimmed area (trim length * trim widths) by the basis weight. For rolls made with the same ts_wound, the length, basis weight, and stdwidth (total width) will be the same, so the average of these values for the group is used. By summing the rolls' widths and subtracting it from the stdwidth, you get the width of trim across the roll.

- StdTrim – The standard trim values for rows in invent_audit grouped by prod_machine_id, grade_spec_made, ts_wound and mx_action_code as calculated by $(\text{avg}(\text{length_lineal}) * \text{avg}(\text{bwt_sold}) * (\text{avg}(\text{stdwidth}) - \text{sum}(\text{width_ord}))) / [\text{factor}]$. Length_lineal and width_ord are from the invent_audit table. bwt_sold is from the grade_spec table. stdwidth is from the ReelWidth table.

The idea is to multiply the trimmed area (trim length * trim widths) by the basis weight. For rolls made with the same ts_wound, the length, basis weight, and stdwidth (total width) will be the same, so the average of these values for the group is used. By summing the rolls' widths and subtracting it from the stdwidth, you get the width of trim across the roll.

- Mx_timestamp – The timestamp of the source invent_audit row. This column is included so queries that use vwTrim can be filtered for report dates.

Queries

- Query 1 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is 'TURNUP'

Grade_spec and ReelWidth are joined on invent_audit.ts_wound within the effective range for their pertinent historic values. grade_spec is joined to invent_audit from grade_spec_made. ReelWidth is also joined on the grade_spec column.
- Query 2 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is 'TURNUP'

Grade_spec and ReelWidth are joined on invent_audit.ts_wound within the effective range for their pertinent historic values. grade_spec is joined to invent_audit from grade_spec_made. ReelWidth is also joined on the grade_spec column.
- Query 3 – This query backs out calculations when rolls were unturned up by multiplying the ActTrim and StdTrim values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is 'UNTURNUP'

Grade_spec and ReelWidth are joined on invent_audit.ts_wound within the effective range for their pertinent historic values. grade_spec is joined to invent_audit from grade_spec_made. ReelWidth is also joined on the grade_spec column.

- Query 4 – This query backs out calculations when rolls were unturned up by multiplying the ActTrim and StdTrim values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is 'UNTURNUP'

Grade_spec and ReelWidth are joined on invent_audit.ts_wound within the effective range for their pertinent historic values. grade_spec is joined to invent_audit from grade_spec_made. ReelWidth is also joined on the grade_spec column.

vwResale

This view unions six queries to combine information from the invent_audit and grade_spec tables to calculate actual and standard "resale" values.

Columns

- Prod_mach_id – The machine id associated with the invent_audit row. This column is included so queries that use vwResale can be grouped.
- Grade_spec – The grade_spec of the roll. This column is included so queries that use vwResale can be grouped.
- ActResale – The sum of actual resale values for rows in invent_audit grouped by prod_machine_id, grade_spec and mx_timestamp as calculated by $(length_lineal * bsw_act * width_act) / [factor]$. Length_lineal, bsw_act and width_act are from the invent_audit table.
- StdResale – The sum of resale values based on a standard basis weight for the grade spec, grouped by prod_machine_id, grade_spec and mx_timestamp. This value is calculated by $(length_lineal * bsw_act * width_act) / [factor]$. Length_lineal and width_act are from the invent_audit table and bwt_sold is from the grade_spec table.
- Mx_timestamp – The timestamp of the source invent_audit row. This column is included so queries that use vwResale can be filtered for report dates.

Queries

- Query 1 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is 'TURNUP'
 - the grade_spec is > 990000Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 2 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is 'TURNUP'
 - the grade_spec is > 990000Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 3 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is NOT 'UNTURNUP' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - the grade_spec is > 990000Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 4 – Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is NOT 'UNTURNUP' or 'STATBEFORE'

- the mx_station is 'CHECKER' or 'ROLLSTATUS'
- the grade_spec is > 990000
Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 5 – This query backs out calculations when rolls were unturned up by multiplying the ActResale and StdResale values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is 110, 115 or 320
 - the action code is 'UNTURNU' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - the grade_spec is > 990000
Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.
- Query 6 – This query backs out calculations when rolls were unturned up by multiplying the ActResale and StdResale values by -1. Selects information from invent_audit where:
 - the grade_spec.paper_grade is NOT 110, 115 or 320
 - the action code is 'UNTURNU' or 'STATBEFORE'
 - the mx_station is 'CHECKER' or 'ROLLSTATUS'
 - the grade_spec is > 990000
Grade_spec is joined on invent_audit.mx_timestamp within the effective range for their pertinent historic values.

vwDowntime

This view consists of one query to select information from the sum_duration_day table to calculate machine downtime.

Columns

- machine_id – The machine id associated with the sum_duration_day row. This column is included so queries that use vwResale can be grouped.
- Grade_spec – The grade_spec of the sum_duration_day row. This column is included so queries that use vwResale can be grouped.
- DownTimeSeconds – The sum of total seconds of downtime grouped by machine_id, grade_spec and prod_date
- Prod_date – The timestamp of the source sum_duration_day row. This column is included so queries that use vwDowntime can be filtered for report dates.

Queries

- Query 1 – Selects information from sum_duration_day where:
 - the machine_type is 'PM'
 - the dt_class_id >= 11 and <= 25

Stored Procedures

spRptStdTonsByMachineByGradeSpec

This stored procedure compiles the data for the Standard Tons Summary by Machine By Grade Spec report.

Parameters

@StartDate – The inclusive start date of the report in Eastern time. If this parameter is null, the beginning of the previous day is used.

@EndDate – The non-inclusive end date of the report in Eastern time. If this parameter is null, the beginning of the current day is used.

For example, to generate the report data for 9/4/02, the parameters should be @StartDate = '9/4/2002 00:00' and @EndDate = '9/5/2002 00:00'.

Operation

After the start and end dates have been determined, a GMT timestamp is generated from them since all the timestamps in the data tables are based on GMT. The GMT timestamp is created by generating a timestamp based on the Eastern time and then adding the number of seconds in 4 or 5 hours depending on daylight or standard time of the specified date.

A temporary table is created to hold intermediate results from the view. This table contains the following columns, which are used in the report:

```
ProdMachId varchar(2),  
GradeSpec varchar(6),  
StdGross decimal(19, 4),  
StdTrim decimal(19, 4),  
StdCull decimal(19, 4),  
StdGood decimal(19, 4),  
ActGood decimal(19, 4),  
DownTimeSeconds int,  
StdResale decimal(19, 4),  
BwtAct decimal(19,4),  
BwtStd decimal(19, 4),  
[Rows] int
```

The values for these columns are inserted from querying the appropriate views for the time period defined by the start and end timestamps. As an example,

```
insert into #RptStdTonsByMachineByGradeSpec (ProdMachId, GradeSpec, StdGross)  
select prod_mach_id, grade_spec, sum(StdGross)  
from vwGross  
where ts_turned_up >= @StartTS and ts_turned_up < @EndTS  
group by prod_mach_id, grade_spec
```

inserts gross values grouped by machine and grade_spec. Similar queries are performed to populate data for the other columns.

Finally, the result set is returned by selecting the sums of these rows from the temporary table, grouped by ProdMachId and GradeSpec. Other columns are generated as part of the final result set. For example, the start and end dates of the report are added. The final result set consists of:

- StartDate – Start date of the report
- EndDate – End date of the report
- ProdMachId – The production machine id
- GradeSpec – The grade spec
- BwtSold – The average “standard” basis weight for a GradeSpec (sum(BwtStd) / sum([Rows])).
- BwtAct – The actual basis weight calculated from (ActGood / StdGood) * (sum(BwtStd) / sum([Rows]))
- StdGross
- StdTrim
- StdCull
- StdGood
- ActGood
- StdBalance = StdGross – StdGood – StdCull – StdTrim
- StdYield = (StdGood / StdGross) * 100

- DownTime – Downtime in the form of HH:MM
- StdResale

spRptStdTonsByMachine

This stored procedure compiles the data for the Standard Tons Summary by Machine report.

Parameters

@StartDate – The inclusive start date of the report in Eastern time. If this parameter is null, the beginning of the previous day is used.

@EndDate – The non-inclusive end date of the report in Eastern time. If this parameter is null, the beginning of the current day is used.

For example, to generate the report data for 9/4/02, the parameters should be @StartDate = '9/4/2002 00:00' and @EndDate = '9/5/2002 00:00'.

Operation

After the start and end dates have been determined, a GMT timestamp is generated from them since all the timestamps in the data tables are based on GMT. The GMT timestamp is created by generating a timestamp based on the Eastern time and then adding the number of seconds in 4 or 5 hours depending on daylight or standard time of the specified date.

A temporary table is created to hold intermediate results from the view. This table contains the following columns, which are used in the report:

```
ProdMachId varchar(2),  
GradeSpec varchar(6),  
StdGross decimal(19, 4),  
StdTrim decimal(19, 4),  
StdCull decimal(19, 4),  
StdGood decimal(19, 4),  
ActGood decimal(19, 4),  
DownTimeSeconds int,  
StdResale decimal(19, 4)
```

The values for these columns are inserted from querying the appropriate views for the time period defined by the start and end timestamps. As an example,

```
insert into #RptStdTonsByMachine (ProdMachId, StdGross)  
select prod_mach_id, sum(StdGross)  
from vwGross  
where ts_turned_up >= @StartTS and ts_turned_up < @EndTS  
group by prod_mach_id
```

inserts gross values grouped by machine. Similar queries are performed to populate data for the other columns.

Finally, the result set is returned by selecting the sums of these rows from the temporary table, grouped by ProdMachId. Other columns are generated as part of the final result set. For example, the start and end dates of the report are added. The final result set consists of:

- StartDate – Start date of the report
- EndDate – End date of the report
- ProdMachId – The production machine id
- StdGross
- StdTrim

- StdCull
- StdGood
- ActGood
- StdBalance = StdGross – StdGood – StdCull – StdTrim
- StdYield = (StdGood / StdGross) * 100
- UpTime – Uptime in the form of HH:MM
- DownTime – Downtime in the form of HH:MM
- MachineAvailability – Total time of the report – total downtime / total time * 100
- StdResale

spRptActTonsByMachineByGradeSpec

This stored procedure compiles the data for the Actual Tons Summary by Machine By Grade Spec report.

Parameters

@StartDate – The inclusive start date of the report in Eastern time. If this parameter is null, the beginning of the previous day is used.

@EndDate – The non-inclusive end date of the report in Eastern time. If this parameter is null, the beginning of the current day is used.

For example, to generate the report data for 9/4/02, the parameters should be @StartDate = '9/4/2002 00:00' and @EndDate = '9/5/2002 00:00'.

Operation

After the start and end dates have been determined, a GMT timestamp is generated from them since all the timestamps in the data tables are based on GMT. The GMT timestamp is created by generating a timestamp based on the Eastern time and then adding the number of seconds in 4 or 5 hours depending on daylight or standard time of the specified date.

A temporary table is created to hold intermediate results from the view. This table contains the following columns, which are used in the report:

```
ProdMachId varchar(2),  
GradeSpec varchar(6),  
ActGross decimal(19, 4),  
ActTrim decimal(19, 4),  
ActCull decimal(19, 4),  
ActGood decimal(19, 4),  
DownTimeSeconds int,  
ActResale decimal(19, 4)
```

The values for these columns are inserted from querying the appropriate views for the time period defined by the start and end timestamps. As an example,

```
insert into #RptActTonsByMachineByGradeSpec (ProdMachId, GradeSpec, ActGross)  
select prod_mach_id, grade_spec, sum(ActGross)  
from vwGross  
where ts_turned_up >= @StartTS and ts_turned_up < @EndTS  
group by prod_mach_id, grade_spec
```

inserts gross values grouped by machine and grade_spec. Similar queries are performed to populate data for the other columns.

Finally, the result set is returned by selecting the sums of these rows from the temporary table, grouped by ProdMachId and GradeSpec. Other columns are generated as part of the final result set. For example, the start and end dates of the report are added. The final result set consists of:

- StartDate – Start date of the report
- EndDate – End date of the report
- ProdMachId – The production machine id
- GradeSpec – The grade spec
- ActGross
- ActTrim
- ActCull
- ActGood
- ActBalance = ActGross – ActGood – ActCull – ActTrim
- ActYield = (ActGood / ActGross) * 100
- DownTime – Downtime in the form of HH:MM
- ActResale

spRptActTonsByMachine

This stored procedure compiles the data for the Actual Tons Summary by Machine report.

Parameters

@StartDate – The inclusive start date of the report in Eastern time. If this parameter is null, the beginning of the previous day is used.

@EndDate – The non-inclusive end date of the report in Eastern time. If this parameter is null, the beginning of the current day is used.

For example, to generate the report data for 9/4/02, the parameters should be @StartDate = '9/4/2002 00:00' and @EndDate = '9/5/2002 00:00'.

Operation

After the start and end dates have been determined, a GMT timestamp is generated from them since all the timestamps in the data tables are based on GMT. The GMT timestamp is created by generating a timestamp based on the Eastern time and then adding the number of seconds in 4 or 5 hours depending on daylight or standard time of the specified date.

A temporary table is created to hold intermediate results from the view. This table contains the following columns, which are used in the report:

```
ProdMachId varchar(2),  
GradeSpec varchar(6),  
ActGross decimal(19, 4),  
ActTrim decimal(19, 4),  
ActCull decimal(19, 4),  
ActGood decimal(19, 4),  
DownTimeSeconds int,  
ActResale decimal(19, 4)
```

The values for these columns are inserted from querying the appropriate views for the time period defined by the start and end timestamps. As an example,

```
insert into #RptActTonsByMachineByGradeSpec (ProdMachId, GradeSpec, ActGross)  
select prod_mach_id, grade_spec, sum(ActGross)  
from vwGross  
where ts_turned_up >= @StartTS and ts_turned_up < @EndTS
```

group by prod_mach_id, grade_spec

inserts gross values grouped by machine and grade_spec. Similar queries are performed to populate data for the other columns.

Finally, the result set is returned by selecting the sums of these rows from the temporary table, grouped by ProdMachId and GradeSpec. Other columns are generated as part of the final result set. For example, the start and end dates of the report are added. The final result set consists of:

- StartDate – Start date of the report
- EndDate – End date of the report
- ProdMachId – The production machine id
- ActGross
- ActTrim
- ActCull
- ActGood
- ActBalance = ActGross – ActGood – ActCull – ActTrim
- ActYield = (ActGood / ActGross) * 100
- UpTime – Uptime in the form of HH:MM
- DownTime – Downtime in the form of HH:MM
- MachineAvailability – Total time of the report – total downtime / total time * 100
- ActResale

spRptReelWidth

This stored procedure compiles the data for the Reel Width sub report – NO LONGER USED.

Parameters

@StartDate – The inclusive start date of the report in Eastern time. If this parameter is null, the beginning of the previous day is used.

@EndDate – The non-inclusive end date of the report in Eastern time. If this parameter is null, the beginning of the current day is used.

For example, to generate the report data for 9/4/02, the parameters should be @StartDate = '9/4/2002 00:00' and @EndDate = '9/5/2002 00:00'.

Operation

After the start and end dates have been determined, a GMT timestamp is generated from them since all the timestamps in the data tables are based on GMT. The GMT timestamp is created by generating a timestamp based on the Eastern time and then adding the number of seconds in 4 or 5 hours depending on daylight or standard time of the specified date.

The machine id and reel widths are selected from the Reel Width table based on the specified time period.

spGetAllReelWidths

This stored procedure simply returns all rows from the ReelWidth table ordered by machine id, grade_spec and effective date. It is used by the reel width administration screen to display the existing reel widths.

Parameters

None

spInsertReelWidth

This stored procedure inserts a new row into the ReelWidth table. It is used by the reel width administration screen to add a reel width for a future time.

Parameters

@PM – The machine id of the production machine.

@grade_spec – The grade_spec for the row.

@stdwidth – The reel width to set for this machine and grade_spec.

@EffectiveDate – The inclusive effective date in Eastern time of this reel width. If this parameter is null, the current time is used.

Operation

After the effective date has been determined, a GMT timestamp is generated from it since all the effective dates in ReelWidth are based on GMT. The GMT timestamp is created by generating a timestamp based on the Eastern time and then adding the number of seconds in 4 or 5 hours depending on daylight or standard time of the specified date.

In a single transaction:

- Any rows with the same PM, grade_spec and ts_effective as the passed parameters are removed from ReelWidth.
- The PM, grade_spec, stdwidth, EffectiveDate parameters are inserted into ReelWidth
- The rows with the same PM and grade_spec are updated to reflect the updated effective and expiration dates. For example, the following two figures show how the rows might start and end for machine 02, grade_spec 000013 (2147483647 is the maximum timestamp):

PM	grade_spec	stdwidth	ts_effective	ts_expires
02	000013	186.5000	0	2147483647

After inserting a value effective midnight 10/11/2002 Eastern, the table looks like:

PM	grade_spec	stdwidth	ts_effective	ts_expires
02	000013	186.5000	0	1034308800
02	000013	190.0000	1034308800	2147483647

spDeleteReelWidth

This stored procedure removes a row from the ReelWidth table. It is used by the reel width administration screen to remove a reel width with a future effective date.

Parameters

@PM – The machine id of the production machine.

@grade_spec – The grade_spec for the row.

@EffectiveDate – The effective date in Eastern time of the reel width to remove.

Operation

After the effective date has been determined, a GMT timestamp is generated from it since all the effective dates in ReelWidth are based on GMT. The GMT timestamp is created by generating a timestamp based on the Eastern time and then adding the number of seconds in 4 or 5 hours depending on daylight or standard time of the specified date.

In a single transaction:

- Any rows with the same PM, grade_spec and ts_effective as the passed parameters are removed from ReelWidth.
- The rows with the same PM, grade_spec are updated to reflect the updated effective and expiration dates.

User-defined Functions

Several user-defined utility functions are used by the reporting system. These functions encapsulate several of the calculations that are performed in many places.

GMTDateToOVTimestamp

Calculates the number of seconds since 1/1/1970 12:00 am to the passed in time. The name of this function may be confusing. It doesn't really matter whether the time entered is local, GMT, or whatever. The name alludes to the fact that it produces timestamps in the same format as the MES timestamps and that the MES timestamps represent GMT time.

Parameters

@dtDate – datetime

Return Value

An int representing the number of seconds that have passed from 1/1/1970 00:00 to @dtDate.

OVTimestampToGMTDate

Calculates the date based on the passed in number of seconds since 1/1/1970 12:00 am.

Parameters

@ITimestamp – int

Return Value

A datetime representing the date based on @ITimestamp.

GMTOffsetSeconds

Calculates the number of seconds that the passed in date is offset from GMT. During Daylight Savings Time, this offset is four hours. During Standard time, this offset is five hours.

Parameters

@dtDate – datetime

Return Value

An int containing the number of seconds @dtDate, based on Eastern time is offset from GMT.

IsDaylightTime

Determines whether or not the passed in Eastern time zone date is in Daylight time or Standard time.

Parameters

@dtDate – datetime

Return Value

A bit indicating whether @dtDate is in Daylight time or not. 1 indicates a Daylight time, 0 indicates a Standard time.

OVTimestampToEasternDate

Calculates the Eastern time from which a GMT timestamp was generated.

Parameters

@ITimestamp – int

Return Value

A datetime representing the Eastern time zone date based on @ITimestamp (a GMT value).

SecondsToHHMM

Converts a number of seconds into a string with the format HH:MM, where HH can be any length representing the number of hours.

Parameters

@seconds – int

Return Value

A varchar(15). As an example, SecondsToHHMM(3600) returns “01:00.”

Production Access Database

The Production Reporting system uses Access as the user interface and report engine to access the data on SQL Server. The Access database contains Tables, Queries (both local and pass-through), Forms, Reports and Modules to support the reporting system.

Report Flow

The following steps describe the flow of the report system within Access to generate a report. It describes the path for the Standard Tons Summary By Machine report:

- The user selects the dates for the report from the Production Reporting form and clicks the button corresponding to the desired report.
- The Click event handler procedure calls a report-specific function (RptStdTonsByMachine) in modReports with the report dates.
- RptStdTonsByMachine calls a more abstract report runner function (RunReportStartEnd) for any sub reports with the pass-through query text, report dates and the sub report name (rptReelWidths). The details of the sub report from this point are similar to the details of the main report from this point, so they are omitted.
- RptStdTonsByMachine calls RunReportStartEnd for the main report with the report dates and the report name (rptStdTonsByMachine).
- RunReportStartEnd:
 - Set the name of the pass-through query for the report (“sp” + report name = “spRptStdTonsByMachine”)
 - Remove the rows from the report-specific work table (tblRptStdTonsByMachine). This table will be filled with data from the pass-through query.
 - Set the text of the pass-through query, including the parameters.
 - Execute a local query (qryRptStdTonsByMachine) that uses the pass-through query as its source. The local query inserts the returned rows into the work table.
 - Run the report (rptStdTonsByMachine).
 - The report uses the work table as its record source.
 - The report’s PageHeader format event sets the caption for the date range in the report using the modUtilities.DateRangeString function.
 - Several functions in modUtilities are called to format data as it is being added to the report.

- If the report is not a sub report, option settings control whether the report is sent to the printer and/or displayed.

Reports

There are four main reports and two sub reports in the Access database.

rptStdTonsByMachineByGradeSpec

This report shows the Standard Tons Summary By Machine By Grade Spec. It contains the rptReelWidth sub report.

There is a direct relationship between the detail information shown in the report and the information returned from the corresponding SQL Server stored procedure – the report merely formats the data and aggregates it for group and report totals:

Report Column	Stored Procedure Column
SPEC #	GradeSpec
Target Basis Weight	BwtSold
Actual Basis Weight	BwtAct
Std Gross	StdGross
Std Trim Loss	StdTrim
Std Culled	StdCull
Std Good Tons	StdGood
Actual Good Tons	ActGood
Std Balance	StdBalance
Yield %	StdYield
Down Time	DownTime
Resale	StdResale

rptStdTonsByMachine

This report shows the Standard Tons Summary By Machine. It contains the rptReelWidth and rptStdTonsByMachineFactorSub sub reports.

There is a direct relationship between the detail information shown in the report and the information returned from the corresponding SQL Server stored procedure – the report merely formats the data and aggregates it for group and report totals:

Report Column	Stored Procedure Column
ID	ProdMachId
Gross On Reel	StdGross
Trim Loss	StdTrim
Culled Rolls	StdCull
Good	StdGood
Balance	StdBalance
Yield %	StdYield
Machine Availability	UpTime
Down Time	DownTime
Machine Availability %	MachineAvailability
Resale	StdResale

rptActTonsByMachineByGradeSpec

This report shows the Actual Tons Summary By Machine By Grade Spec. It contains the rptReelWidth sub report.

There is a direct relationship between the detail information shown in the report and the information returned from the corresponding SQL Server stored procedure – the report merely formats the data and aggregates it for group and report totals:

Report Column	Stored Procedure Column
SPEC #	GradeSpec
Gross On-Reel	ActGross
Trim Loss	ActTrim
Culled Rolls	ActCull
Good	ActGood
Balance	ActBalance
Yield %	ActYield
Down Time	DownTime
Resale	ActResale

rptActTonsByMachine

This report shows the Actual Tons Summary By Machine. It contains the rptReelWidth sub report.

There is a direct relationship between the detail information shown in the report and the information returned from the corresponding SQL Server stored procedure – the report merely formats the data and aggregates it for group and report totals:

Report Column	Stored Procedure Column
ID	ProdMachId
Gross On Reel	ActGross
Trim Loss	ActTrim
Culled Rolls	ActCull
Good	ActGood
Balance	ActBalance
Yield %	ActYield
Machine Availability	UpTime
Down Time	DownTime
Machine Availability %	MachineAvailability
Resale	ActResale

rptReelWidth

This report shows the standard reel widths used in the report by machine. If the reel width is not the original reel width in the table, the effective date is also displayed. This report is used as a sub report for all the main reports.

rptStdTonsByMachineFactorSub

This report displays the ratio of standard good to actual good tons by machine. It is used as a sub report on the Standard Tons Summary by Machine report.

Queries

Most of the queries in the Access database are part of the data pipe for gathering report data. In general, they will consist of a prefix and the report name.

- A prefix of “sp” means the query is a pass-through query that represents a similarly named SQL Server stored procedure.
- A prefix of “qry” followed by the report name means it inserts data into the “tbl” + reportname table that it retrieves from the “sp” + reportname query.

spGetAllReelWidths, spInsertReelWidth and spDeleteReelWidth are used by the Reel Width administration form to retrieve and update reel widths in SQL Server.

Tables

The main production reports and rptReelWidths use tables as their record sources. These tables have the naming convention of “tbl” + reportname. The columns in the tables match the columns returned from the corresponding SQL Server stored procedures.

tblOption

This table contains user options for the Access side of the reporting system. tblOption currently holds the following options, which are settable from the Report Options form:

- DisplayReport – A Value of 1 indicates that when the report is finished running, show it on the monitor.
- PrintReport – A Value of 1 indicates that when the report is finished running, send it to the printer automatically.
- NotifyWhenReportPrinted – A Value of 1 indicates that when the report is sent to the printer, display a message box to that effect.

Forms

The Access database has three forms for running reports, modifying options and administering reel widths.

- Production Reporting – This is the startup form for the Access Database Refer to Figure 1.
 - The user may select the date range for a report and start it by clicking the desired button.
 - The start and end dates selected from the calendars are INCLUDED in the report. The label above the report buttons shows the current date range.
 - When this form is displayed, a Production Reporting menu is available to show the Reporting Options and Set Reel Width forms.
 - If any grade specs defined in the ss.grade_spec table do not have reel widths assigned for each production machine, a message will be displayed and the report will not run.

Figure 1. Start-Up Form for Production Reporting System.

The screenshot shows the 'Macon Production Reporting' application window. The interface includes a menu bar with options: File, Edit, View, Insert, Format, Records, Tools, Window, Help, and Production Reporting. Below the menu bar is a toolbar. The main content area is split into two vertical panels. The left panel contains two calendar controls. The top one is titled 'Select Start Date' and shows a calendar for October 2002 with the date 11th selected. The bottom one is titled 'Select End Date' and also shows a calendar for October 2002 with the date 11th selected. The right panel is titled 'Select Report' and shows 'Report of: 10/11/2002'. Below this, there are four buttons for running reports: 'Run Actual Tons Production Summary Report By Machine By Grade Spec', 'Run Actual Tons Production Summary Report By Machine', 'Run Standard Tons Production Summary Report By Machine By Grade Spec', and 'Run Standard Tons Production Summary Report By Machine'. At the bottom of the right panel is a 'QUIT' button.

Figure 2. Reporting Options Form.

The screenshot displays the 'Macon Production Reporting' application window. The main window has a menu bar (File, Edit, View, Insert, Format, Records, Tools, Window, Help) and a title bar. The main content area is divided into several sections:

- Select Start Date:** A calendar for October 2002. The start date is set to October 1, 2002.
- Select End Date:** A calendar for October 2002. The end date is set to October 11, 2002.
- Select Report:** A list of report options:
 - Run Actual Tons Production Summary Report By Machine By Grade Spec
 - Actual Tons Production Summary Report By Machine
 - Run Standard Tons Production Summary Report By Machine By Grade Spec
 - Run Standard Tons Production Summary Report By Machine
- QUIT:** A large button at the bottom.

A 'Report Options' dialog box is open in the center, titled 'When the report completes:'. It contains three checkboxes:

- Print the report
- Display the report
- Notify me

A 'Close' button is located at the bottom of the dialog box.

- Reporting Options – Use this form to indicate reporting preferences as described under tblOption. Refer to Figure 2.
- Reel Width –Use this form to view and set reel widths. Refer to Figure 3.
 - Click Add New to display the controls used to specify new reel width information.
 - Set the Reel Width, Grade Spec and Effective Date. Clicking the calendar will set the Effective Date value when “Immediately” is not selected.
 - Click Apply to save the information to the SQL Server.
 - To remove a future reel width change, click the correct row in the grid and click Remove.
 - To update a future reel width, click Add New and specify the same date as a future change. Click to apply to update the information on the SQL Server.

Figure 3. Reel Width Form.

The screenshot shows the 'Reel Width' form within the 'Macon Production Reporting' application. The form is divided into several sections:

- Select Date:** A calendar for October 2002. The date 15 (Tuesday) is selected.
- Select Production Machine:** A table with two columns: '01' and '02'. Below it is a table with columns 'Grade Spec', 'Effective Date', and 'Reel Width'. The data in this table is as follows:

Grade Spec	Effective Date	Reel Width
000013	N/A	198.900
353016	N/A	198.900
353017	N/A	198.900
353018	N/A	198.900
353020	N/A	198.900
971018	N/A	198.900
971021	N/A	198.900
993016	N/A	198.900
993020	N/A	198.900
- Form Fields:**
 - Production Machine: 01
 - Grade Spec: 000013
 - Reel Width: 199.5
 - Effective Date: Immediately, 10/15/2002 00:00:00
- Buttons:** Add New, Remove, Apply, Cancel, Close.

Modules

Configuration

SQL Server

MES

The MESAtlanta DSN used by the Nightly Report Data Mover – MES DTS package should be set up using these steps:

- Click Start->Programs->Administrative Tools->Data Sources
- Click System DSN
- Click Add
- Select the OpenLink Generic 32 Bit Driver and click Finish
- Name=MESAtlanta; Type=Informix 5; Protocol=TCP/IP; Hostname=10.101.16.132; Path=online_Client; UserID=openlink
- Click OK

To change the settings for MES (such as UserID), you must make any appropriate changes in the MESAtlanta DSN. In addition, the following steps must be done to modify the Nightly Report Data Mover – MES DTS package:

- Edit the MES DTS package
- Double-click any of the MES connections
- Change the User/System DSN if the name has changed
- Update the Username and Password fields

- Click OK
- Click OK on the Task References dialog
- Save the package

Peak

The Oracle alias used by the Nightly Report Data Mover – Peak DTS package should be set up using these steps:

- Click Start->Programs->Oracle For Windows NT->SQL Net Easy Configuration
- Click OK for Add database alias
- Database Alias=peak, click OK
- Protocol is TCP/IP, click OK
- TCP/IP Hostname=10.101.16.247; Database Instance = OFS1
- Click OK
- Click OK

To change the settings for peak (such as username), the following steps must be done to modify the Nightly Report Data Mover – Peak DTS package:

- Edit the Peak DTS package
- Double-click any of the peak connections
- Change the appropriate information
- Click OK
- Click OK on the Task References dialog
- Save the package

SQL Server

If the location of the SQL Server data store changes, the SQL Server connection in both the MES and Peak DTS packages must be updated.

- Edit the DTS package
- Double-click any of the SQL Server connections
- Change the appropriate information – Server, Database, login information, etc.
- Click OK
- Click OK on the Task References dialog
- Save the package

User Computer

The Access production report database is named ProductionReporting.mdb and may exist anywhere on a user's computer. The database is designed so that each user should have their own copy. **ERRORS CAN OCCUR IF MORE THAN ONE PERSON IS RUNNING THE REPORTS FROM THE SAME DATABASE AT THE SAME TIME.** The pass-through queries in Access use a DSN named Production Report Data Store.

NOTE: The following procedure creates a DSN using "Windows NT authentication using the network login ID" to access the SQL Server. The SQL Server administrator (Lamar Maddox) must have configured the user's id for access to SQL Server.

To create Production Report Data Store:

- Click Start->Programs->Administrative Tools->Data Sources
- Click System DSN
- Click Add
- Select SQL Server and click Finish
- Name=Production Report Data Store; Server=MACSRV3
- Click Next, Click Next

- Check “Change the default database to:” and select ClientDataStore, Click Next
- Click Finish
- Click Test Data Source to verify the connection to the database

If the location of the SQL Server data store changes, the appropriate changes must be made in the Production Report Data Store DSN.

Appendix A

