

Introduction to COBOL Batch Processing

Synopsis:

- Designing Batch Processing is very important as it can speed up processing of a lot of the reports and maintenance of a lot of application.

Keywords:

- SSAD, Batch Processing

Written By:

- Mario A. Diaz
 - 21 4th Avenue, Cubao
 - Quezon City, National Capital Region
 - Philippines
 - Email Address:
 - ❖ mar10ad1az@yahoo.com
 - ❖ mad1az.1024@gmail.com
 - Yahoo group: http://tech.groups.yahoo.com/group/ssad_oad_depot/
 - Google group: <http://groups.google.com/group/ssad-oad-depot>
 - Last Updated : 06/23/2009 3:21 PM

1 Introduction to COBOL Batch Processing

1.1 Definitions

■ Batch

- A batch is a collection of input transactions, identified according to some logical or business rule.
- The purpose of batching is to provide a means of controlling the input and reducing the work involved in identifying and correcting errors.



Figure 1-1: A Batch may contain one or more input transactions.

■ Batch Processing

- Type of data processing in which a number of similar input items are grouped for processing serially with a minimum of operator intervention and no end-user interaction.
- Inputs are grouped based on the processing cycle requirements. i.e. daily, weekly, monthly, etc.
- Outputs are produced at the end of each processing cycle before the next one begins.
- Most effective way of processing large amounts of data regularly.

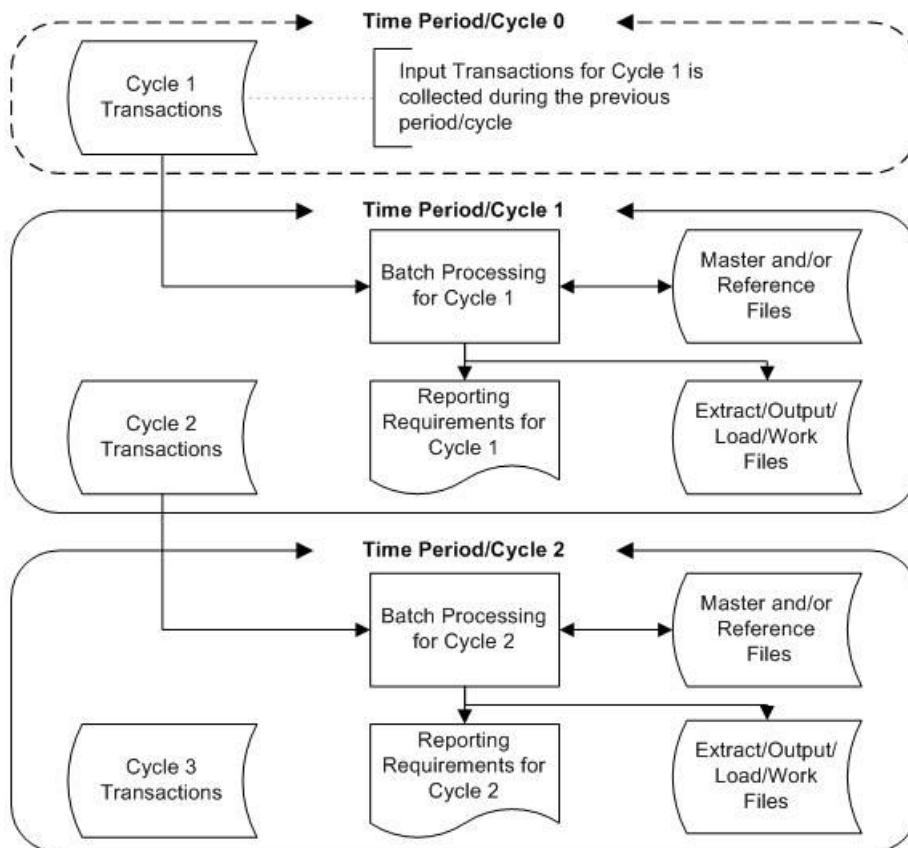


Figure 1-2: Relationships of Time Period/Cycles and Batch Processing.

■ Batch Job

- A batch job is a combination of programs or utilities that accomplishes a business functionality/sub-functionality without or minimal user intervention. Some are defined to run automatically at a certain time.

■ **Utility**

- A utility is an aid to achieve a generic task such as loading a file, converting a file, sorting a file, merging files into one file, splitting a file into many files, or report generation.
- The advantages of using utilities are that they are readily available and normally bug free. However, it may not be possible to implement certain complicated logic using utilities alone. So be careful in relying too much on utilities, it may not support future changes in business functionality.

■ **Program**

- Programs are meant to implement condition checking, looping logic, module invocation, and data manipulation.
- Programs are developed by programmers to meet a more specific need such as retrieving and/or saving specific data from/to tables or files.
- Significant development effort is involved in building the program modules in terms of analysis, design, coding, testing, and documentation.

1.2 Types of Information Systems Requiring Batch Processing

- **Transaction Processing Systems (TPS)**
 - Processing cycles can be: On-demand, daily, weekly, bi-weekly, monthly, bi-monthly and quarterly. Very seldom will TPS require semi-annual and annual processing.
- **Management Information Systems (MIS)**
 - Processing cycles can be: On-demand, daily, weekly, bi-weekly, monthly, bi-monthly, quarterly, semi-annual and annual.
- **Decision Support Systems (DSS)**
 - Typically on-demand but other processing cycles may be used.
- **Executive Information Systems (EIS)**
 - Processing cycles can be: Monthly, bi-monthly, quarterly, semi-annual, annual processing cycles. It is not unusual to use historical information - one, two, and up to 5 years in the past - for planning and forecasting purposes - one, two and up to 5 years to the future.

Characteristic	TPS	MIS	DSS	EIS
Inputs	Internal Operational and Accounting Data	Data from Internal Business Elements	Information from Internal Business Elements and MIS	Information from Business and Other Support Systems
Processing	Structured Process for Data Collection and Reporting	Data Processing for Organization and Reporting	Modeling and Calculation for Tailored Analyses and Reporting	Research, Analysis, and Summarization for Integration and Planning
Outputs	Rule Based Direction	Information through Structured Reports	Tailored Analyses and Reports for Recommended Courses of Action	Summary Reports with "Drill Down" Capability
Major Focus	Transaction Processing and Tracking	Status, Measures of Performance, and Control	Use of Data in Models	Oversight, Review, and Strategy
Major Components	Basic File I/O and/or Data Base with RPG - Report Program Generator or Utility	Data Base with DBMS plus RPG/SQL - Structured Query Language	Data Base Models Qualitative and Quantitative Analysis	Data Base Models Qualitative and Quantitative Analysis, Statistics
Development Process	System Analysts Using an SDLC - Systems Development Life Cycle	System Analysts and Steering Committee Using an SDLC	Systems/Business Analysts, Management Users, and Tool Specialists Using Prototypes/JAD	Systems Analysts, Executive Users, and Tool Specialists Using Prototypes/JAD - Joint-Application Design

Figure 1-3: Characteristics of Information Systems

1.3 Batch Processing Strategies

To help analyze, design and implement batch systems, the systems analyst and designer should be provided a framework on how to build the batch system. The framework is a Standard Batch Architecture consisting of:

- Basic batch application building blocks and;
- Programming design patterns with systems flowcharts.

Programmers should be provided with;

- Structure descriptions - process flows, use cases, and/or structure charts and;
- Module code shells.

When starting to design a batch job, the business logic should be decomposed into a series of steps which can be implemented using the following batch processing building blocks

- **Basic Components or Building Blocks:**
 - Conversion
 - For each type of file from an external system, a conversion component will need to be developed to convert the input records into a standard format output records required for the next processing steps.
 - This type of components can partly or entirely consist of conversion utilities.
 - Validation
 - Validation components ensure that all input and output records are correct and consistent.
 - It uses routines, often called "validation rules" or "check routines", that check for correctness, meaningfulness, and security of data that are used by the system.
 - The rules may be implemented through the automated facilities of a data base, or by the inclusion of explicit validation logic.
 - Update
 - It performs processing on input transactions from an extract or a validation component.
 - The update component involve reading a master/reference record to obtain data required for processing based on the input record, potentially updating the master/reference record.
 - Extract
 - An application component that reads a set of records from a database or input file, selects records based on predefined rules, and writes the records to a output file for transform, print and/or load processing.
 - Update and Extract
 - The update and extract functions are combined in a single component.
 - Print
 - Application component that reads an input file, restructures data from this record according to a standard reporting format, and produces reports direct to a printer or to a spool file. If a spool file is used, a spool print utility is used to produce printouts.
 - Extract, Transform, Load (ETL)
 - Extract, Transform, and Load (ETL) is a process in data warehousing that involves
 - ❖ Extracting data from outside sources;
 - ❖ Transforming it to fit business needs (which can include quality levels), and ultimately;
 - ❖ Loading it into the end target, i.e. the data warehouse.
 - ETL is important, as it is the way data actually gets loaded into the data warehouse.
 - ETL can also be used for the integration with legacy systems.
 - Usually ETL implementations store an audit trail on positive and negative process runs.
 - ❖ Note: This manual will not expound concepts related to the ETL components, but some design patterns, structure descriptions and code shells can be used to analyze, design and code ETL functionalities.

- Transform
 - The transform component applies a series of rules or functions to the Extract output file data to derive the data to be used by the Load and/or Print components.
 - Some data sources will require very little or even no manipulation of data. In other cases, one or more transformations routines may be required to meet the business and technical needs of the Load and Print components.
 - Conversion, Validation, Print and Load components are special types of Transform components.
- Load
 - The Load components load the data into the end target, usually the data warehouse (DW).
 - Depending on the requirements of the organization, this process ranges widely. Some data warehouses might weekly overwrite existing information with cumulative, updated data, while other DW (or even other parts of the same DW) might add new data in a detailed form, e.g. hourly.
 - The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs. More complex systems can maintain a history and audit trail of all changes to the data loaded in the DW.
 - As the load phase interacts with a database, the constraints defined in the database schema as well as in triggers activated upon data load apply (e.g. uniqueness, referential integrity, mandatory fields), which also contribute to the overall data quality performance of the Extract-Transform-Load process.
- **Utilities and Other Components:**
 - Sort
 - A component that reads an input file and produces an output file where records have been re-sequenced according to a sort key field in the records. Sorts are usually performed by standard system utilities. DFSORT or the ICETOOL utility is used for sorting or formatting data, including numeric format conversion, extracting required fields from a record, and eliminating records based on a condition.
 - Split
 - A component that reads a single input file, and writes each record to one of several output files based on a field value. Splits can be tailored or performed by parameter-driven standard system utilities.
 - Merge
 - A component that reads records from multiple input files and produces one output file with combined data from the input files. Merges can be tailored or performed by parameter-driven standard system utilities.
 - Data Base
 - The LOAD utility can be used for mass insertion or mass deletion of records to/from the database.
 - Report Generator
 - Crystal Reports is a business intelligence application used to design and generate reports from a wide range of data sources.
 - Data Capture
 - These are off-line and/or on-line applications that captures data either automatically or through manual data entry. Technologies typically considered as part of Automatic Identification and Data Capture (AIDC) include:
 - ❖ Bar codes, Radio Frequency Identification (RFID), biometrics, magnetic stripes;
 - ❖ Optical Character Recognition (OCR), Magnetic Ink Character Recognition (MICR);
 - ❖ Smart card, and voice recognition.

1.4 Standard Batch Application Architecture

When I was still a novice in COBOL programming and systems analysis and design, I have often wondered, how senior programmers and analyst could create on-line and batch systems run flowcharts from their mind and they would work. It was only later that I found out they were using standard batch and on-line application architecture principles. If one follows these principles, the design of components becomes easy.

However, I learned that there was no textbook that expounded these principles. It seems that the only way to learn them was through the traditional apprenticeship model. You learn as you emulate what your mentor is doing.

They have forgotten a very sound adage “It is better to learned from the knowledge and mistakes of others, so that you get the skills and avoid the pitfalls.” So I have decided to write articles and later a book on how to design and implement a COBOL batch processing standard architecture.

■ **Utilities and Other Components:**

□ **Sort-Split-Merge Components**

- These utilities are normally imbedded in the basic components or done before executing a basic component. They may also be grouped together to form a batch job. If this is the case, then the systems run flowchart below can represent the batch job.

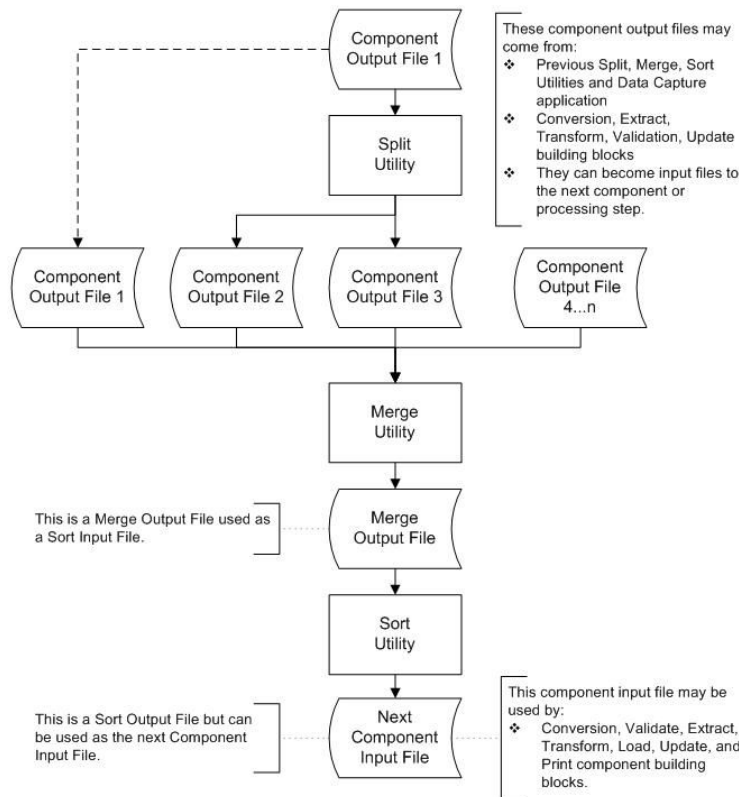


Figure 1-4: Split-Merge-Sort Components or Building Blocks

- The exact sequence will depend on the processing steps that a batch job requires. For example:
 - If you want to design a Print component input files that you want to send to many departments – so that they can print their own reports- and the Component output file contains all the printing requirements for all departments, then the sequence may be:
 - ❖ Sort Component output file by DEPARTMENT-CODE first, then from the sorted file, Split into many departmental Print input files.

- If you want to design a Validation input file from many BATCH-CODE sorted Data Capture output files, then sequence may be:
 - ❖ Data Capture output files must be merged first, then sorted based on the validation requirement – i.e. by ACCOUNT-NO, and by TRANSACTION-CODE – and the resultant sorted file becomes the Validation input file.

□ **Load Component**

- The LOAD utility can be used for mass insertion or mass deletion of records to/from the database.
- Some Load utility component can access indexed files and add, update or delete records. However, I have not yet seen a Load utility that can access an input sequential file and from a Load Utility Input File create an updated output sequential file.

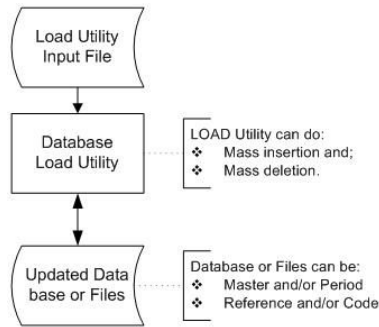


Figure 1-5: Load Component Systems Flowchart

□ **Report Generator Component**

- Crystal Reports is a business intelligence application used to design and generate reports from a wide range of data sources.
- For simple reports, RPG programming language may be used.

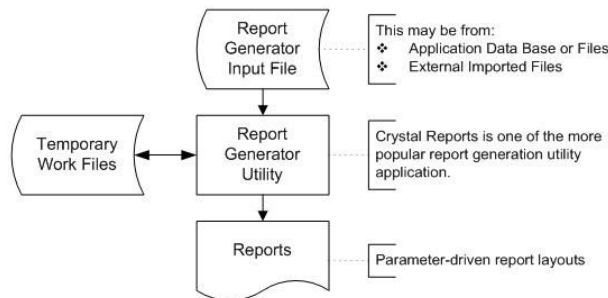


Figure 1-6: Report Generator Component Run Chart

□ **Data Capture Components**

- These are off-line and/or on-line applications that captures data either automatically or through manual data entry. Refer to Figure 1-7: Data Capture Components.
 - Samples of manual data entry systems:
 - ❖ Key-to-Tape/Disk – all encoded records are stored in magnetic tape, diskettes, optical, or memory disks.
 - ❖ On-line data entry components or applications.
- WEB-based files may also be used. i.e. they must be processed with care, as there may be some formatting or code scheme problems (ASCII to EBCID, etc).
- XML based files are the best format to use. Most data base systems have an embedded XML converter as one of its utilities.
- The most important thing to remember in Data Capture is the resultant next Component Input File is compatible with the batch processing environment.

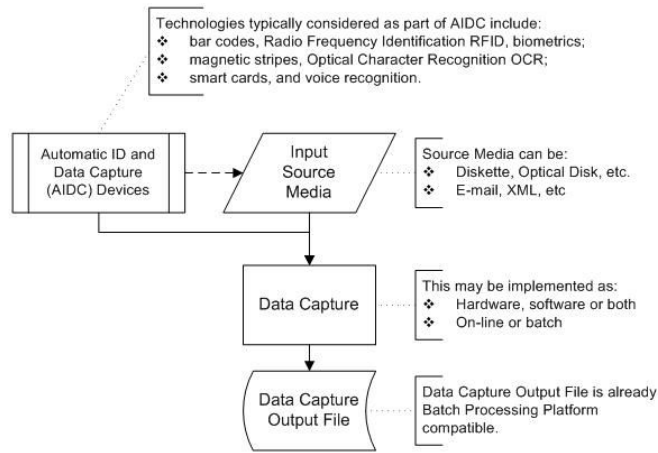


Figure 1-7: Data Capture Components Systems Flowchart

■ **Basic Components or Building Blocks**

□ **Conversion Component**

- For each type of file from an external system, a conversion component will need to be developed to convert the input records into a standard format output records required for the next processing steps.
- The conversion component can perform one or all of the following:
 - Selecting only certain columns to convert and/or certain records
 - Generating or changing sort key values
 - ❖ If you want to design a Conversion output file containing only transaction data from one BATCH-CODE sorted Data Capture output file containing transactions and batch control records, then the processing sequence may be:
 - Data Capture output file data must be reformatted first, the sort key – ACCOUNT-NO and TRANSACTION-CODE – must be created, selected columns or fields moved into the Conversion data fields and batch control records must be removed.

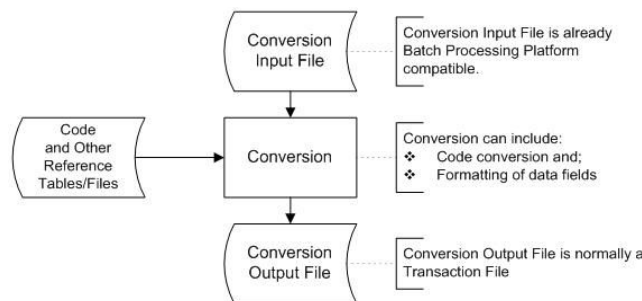


Figure 1-8: Conversion Component Systems Flowchart

- You can also perform the following functions, but do not over-do it. Some functions are better handled in the next processing steps.
 - ❖ Translating coded values (*e.g.*, if the input file stores 1 for male and 2 for female, but the batch system uses 'M' for male and 'F' for female)
 - ❖ Encoding free-form values (*e.g.*, mapping 'Male' to 'M', 'Female' to 'F')
 - ❖ Deriving a new calculated value (*e.g.*, SALES-AMOUNT = QTY * UNIT-PRICE)
 - ❖ Appending data from master or reference files

□ **Validation Component**

- Validation components ensure that all input and output records are correct and consistent.
- It uses routines, often called "validation rules" or "check routines", that check for correctness, meaningfulness, and security of data that are used by the system.

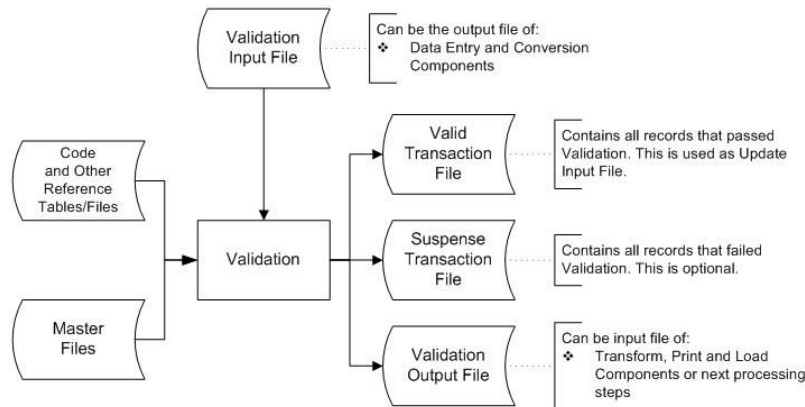


Figure 1-9: Validation Component Systems Flowchart

- Design issues:
 - Reject every error?
 - Reject defined error types only?
 - Do you notify user upon detection of an error?
 - Do you put erroneous data in a Suspense file?
- Validation methods:
 - Format or picture check
 - ❖ Checks that the data is in a specified format (template), e.g., dates have to be in the format MM/DD/YYYY.
 - ❖ Telephone numbers should include the digits and possibly the characters +, -, (, and) (plus, minus, and parentheses)
 - Data type checks
 - ❖ Checks the data type of the input and give an error message if the input data does not match with the chosen data type, e.g., In an input box accepting numeric data, if the letter 'O' was typed instead of the number zero, an error message would appear.
 - Range check
 - ❖ Checks that the data is within a specified range of values, e.g., the month of a person's date of birth should lie between 1 and 12.
 - Limit check
 - ❖ Unlike range checks, data is checked for one limit only, upper OR lower, e.g., data should not be greater than 2 (>2).
 - Presence check
 - ❖ Checks that important data are actually present and have not been missed out, e.g., customers may be required to have their telephone numbers listed.
 - Check digits
 - ❖ Used for numerical data. An extra digit is added to a number which is calculated from the digits. The computer checks this calculation when data are entered, e.g., The ISBN for a book. The last digit is a check digit calculated using a modulus 11 method.
 - Batch totals
 - ❖ Checks for missing records. Numerical fields may be added together for all records in a batch. The batch total is entered and the computer checks that the total is correct, e.g., add the 'Total Cost' field of a number of transactions together.

- Hash totals
 - ❖ This is just a batch total done on one or more numeric fields which appears in every record, e.g., add the ACCOUNT-NO, QUANTITY, etc together for a number of or all Customer transaction or master records.
- Consistency Checks
 - ❖ Checks fields to ensure data in these fields corresponds, e.g., IF TITLE = "MR." THEN GENDER = "M".
- Cross-system or Cross-files Consistency Checks
 - ❖ Compares data in different systems and/or files to ensure it is consistent.
 - The address and/or name for the customer with the same id are the same in both systems and/or files.
 - ❖ The data may be represented differently in different systems and may need to be transformed to a common format to be compared.
 - One system and/or file may store customer name in a single Name field as 'DOE, JOHN Q', while another in three different fields: FIRST-NAME (JOHN), LAST-NAME (DOE) AND MIDDLE-INITIAL (Q).
- Spelling check
 - ❖ Looks for spelling and grammar errors. This may be very difficult to program in a mainframe environment.

□ **Update Components**

- It performs processing on input transactions from an extract or a validation component.
- The update component involve reading a master/reference record to obtain data required for processing based on the input record, potentially updating the master/period record.
- Master File Update
 - These are the domain or actor artifacts that must be updated for every processing cycle.
 - For Example:
 - ❖ Member, Account, Video, etc.

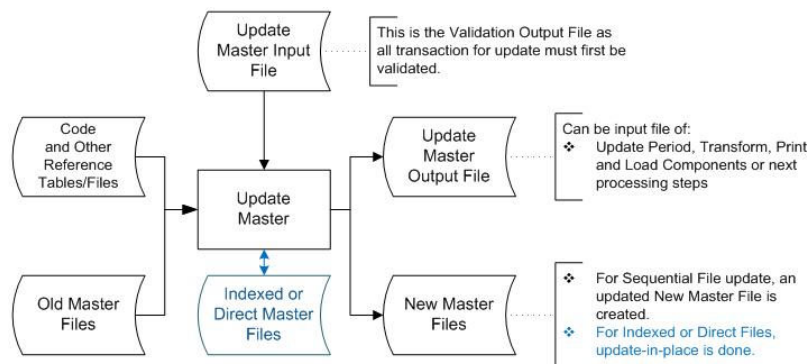


Figure 1-10: Update Master Component Systems Flowchart

- Period File Update
 - These are the files that you store year-to-date, quarter-to-date, month-to-date or any period-to-date information that you would like to monitor, or required by reports and inquiries.
 - For Example:
 - ❖ Loans Statistics – how many times a title or video was loaned or rented
 - ❖ To-Date Information – From Start, Year-to-Date, Quarter-to-Date, etc.
 - ❖ Archival or Historical Information - This Quarter, Last Quarter, Same Quarter Last Year, etc.

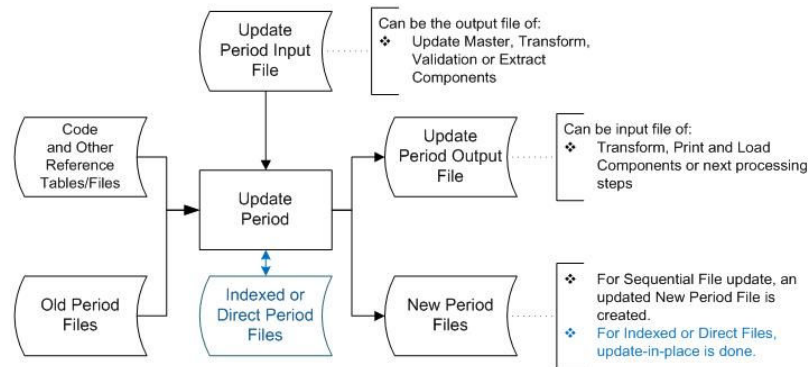


Figure 1-11: Update Period Component Systems Flowchart

- Update Approaches
 - Transaction-Driven Update
 - ❖ Transaction direct update logic
 - File-Driven Update
 - ❖ Master file or data base structure directs the update logic
- Transaction-Driven Updates
 - It is best to sort Update Input File – transaction - before updating.
 - Only selected records from the data base are read and rewritten which means the Master/Period file must be indexed or direct, and Update-in-place is used.
 - Multiple transactions may update the same record. Certain types of transaction must be processed ahead of others. The following is the recommended sequence of processing transactions to insure accuracy:
 - ❖ Changes to Static data - change in name and address
 - ❖ Changes to Dynamic Data - change in credit limit and change in account no. due to lost card
 - ❖ Changes based on Current Transactions
 - Debits (Debit Memo, Sales Invoice)
 - before Credits (Credit memo, Return Slip, Official Receipts)
- File-Driven Updates
 - It is mandatory to sort Update Input File – transaction by Account, Date and Transaction Type- before updating.
 - Read all Old Master/Period File records and write all New Master/Period File records.
 - Multiple transactions may update the same record paragraph in Transaction-Driven Updates above also applies.
 - Other File-Driven Update Approach functionalities:
 - ❖ Posting of Internal Events
 - End of year initialization of period-to-date values
 - Adding one year to the age of all employees
 - ❖ Data Base control and integrity check
 - The only way to check if a file is still intact is to read and rewrite all records
 - ❖ Reduce Runtime
 - Sequential File Processing is the fastest and safest technique for Update.
 - ❖ Recovery and Restart
 - For very large data bases or files, Sequential File Processing allows for Update with breakpoints. It allows the system to restart if problems occur during Update. Restore can also be designed and programmed the same way.

□ **Extract Component**

- An application component that reads a set of records from a database or input file, selects records based on predefined rules and, reformats and writes the records to an output file for update, transform, print and/or load processing.

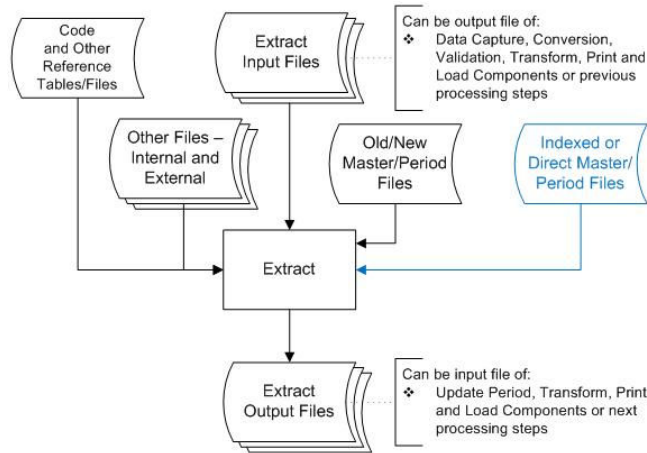


Figure 1-12: Extract Component Systems Flowchart or Run Chart

- **Approaches:**

- Embedded Extract
 - ❖ The update and extract functions are combined in a single component - refer to Update-Extract Component.
 - ❖ Embedded extract is used where report volume and complexity are low
- Stand-alone Extract
 - ❖ The extract function is run independent after the update processing.
 - ❖ Each report for each reporting cycle.

□ **Update and Extract Components**

- Update and Extract
 - The update and extract functions are combined in a single component.
 - This applies to both on-line and batch processing.

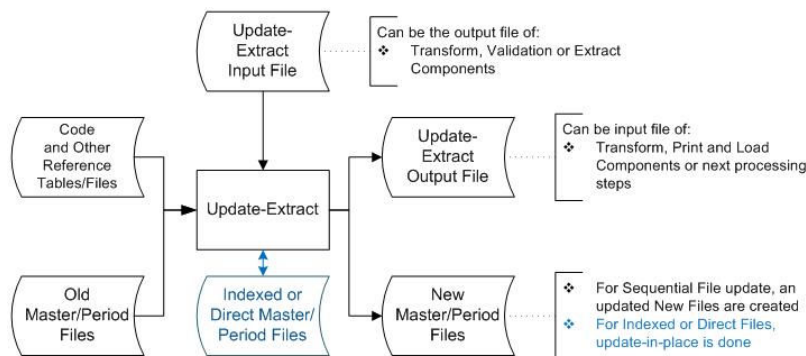


Figure 1-13: Extract and Update Components Run Chart

- **Update and Extract Approaches:**

- For on-line systems
 - ❖ Use stand-alone extract
- For batch Transaction-Driven Update Approach
 - ❖ Produce validation and error reports and audit trails using an embedded extract
 - ❖ Produce all other reports using a Stand-alone extract

- For batch File-Driven Update Approach
 - ❖ Produce all reports using an embedded multi-report extract
- For Commercial Packages or Off-the-Shelf Applications
 - ❖ Use stand-alone extract for all on-line and batch updates

□ **Transform Component**

- The transform component applies a series of rules or functions to previous Component output file data to derive the data to be used by the Load and/or Print components.
- Data Capture, Conversion, Validation, Print and Load components are special types of Transform components.

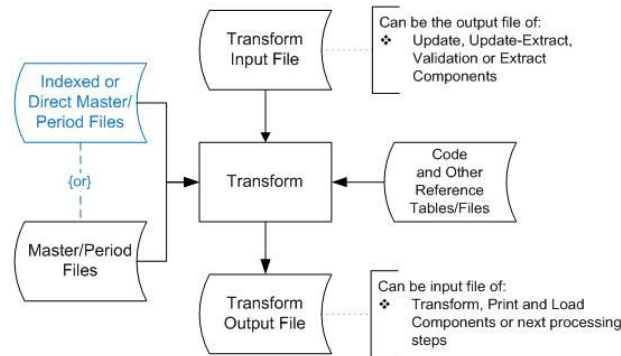


Figure 1-14: Transform Component Run Chart

- **Transform Functionalities:**
 - Selecting only certain columns or fields to load (or selecting null columns not to load)
 - Translating coded values and encoding free-form values
 - Deriving a new calculated value
 - Joining together data from multiple sources (e.g., lookup, merge, etc.)
 - Summarizing multiple rows of data (e.g., total sales for each store, and for each region)
 - Generating surrogate key – sort key, concatenated key, etc - values
 - Transposing or pivoting (turning multiple columns into multiple rows or vice versa)
 - Splitting a column into multiple columns (e.g., putting a comma-separated list specified as a string in one column as individual values in different columns)
 - Applying any form of simple or complex data validation; if failed, a full, partial or no rejection of the data, and thus no, partial or all the data is handed over to the next step, depending on the rule design and exception handling.

□ **Load Component**

- The Load components load the data into the end target, usually the data warehouse (DW).
- The Load component can also be used to update code tables, reference files and master and period files. It can be referred to as a Simple Update component.

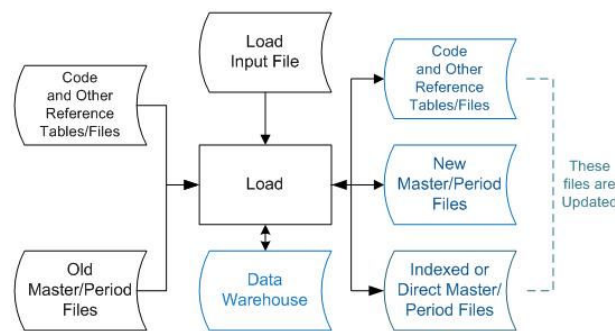


Figure 1-15: Load Component Run Chart or Systems Flowchart

□ **Print Component**

- Application component that reads an input file, restructures data from this record according to a standard reporting format, and produces reports direct to a printer or to a spool file. If a spool file is used, a spool print utility is used to produce printouts.
- The printouts or spool file may be for one or more reports.

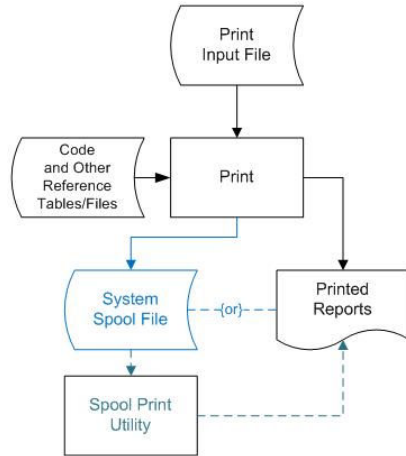


Figure 1-16: Print Component Systems Flowchart

□ **Extract-Transform-Load (ETL) Components**

- Extract, Transform, and Load (ETL) is a process in data warehousing that involves
 - Extracting data from outside sources;
 - Transforming it to fit business needs (which can include quality levels), and ultimately;
 - Loading it into the end target, i.e. the data warehouse.

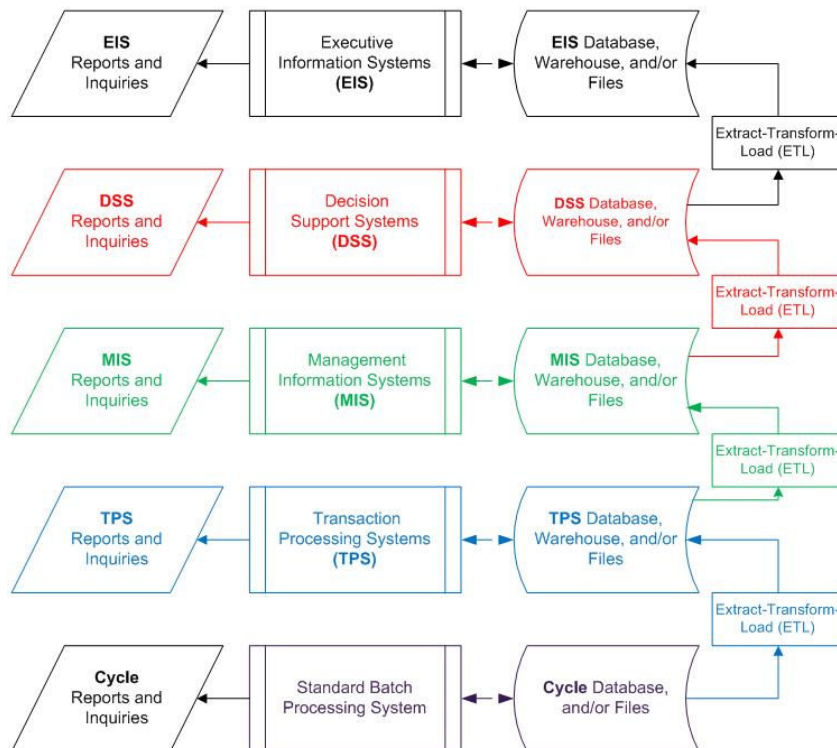


Figure 1-17: ETL Relation with Information Systems

Prepared by: Mario A. Diaz, President – Sistema-Pinas Inc. Philippines

Email address: mar10ad1az@yahoo.com