TABLE 13-1 Application Controls for Processing Integrity

Process Stage	Threats/Risks	Controls
Input	Data that is: • Invalid • Unauthorized • Incomplete • Inaccurate	Forms design, cancellation and storage of documents, authorization and segregation of duties controls, visual scanning, data entry controls
Processing	Errors in output and stored data	Data matching, file labels, batch totals, cross-footing and zero-balance tests, write-protection mechanisms, database processing integrity controls
Output	• Use of inaccurate or incomplete reports • Unauthorized disclosure of sensitive information • Loss, alteration, or disclosure of information in transit	Reviews and reconciliations, encryption and access controls, parity checks, message acknowledgment techniques, blockchain

field check - An edit check that tests whether the characters in a field are of the correct field type (e.g., numeric data in numeric fields).

sign check - An edit check that verifies that the data in a field have the appropriate arithmetic sign.

limit check - An edit check that tests a numerical amount against a fixed value.

range check - An edit check that tests whether a data item falls within predetermined upper and lower limits.

size check - An edit check that ensures the input data will fit into the assigned field.

completeness check (or test) - An edit check that verifies that all data required have been entered.

validity check - An edit test that compares the ID code or account number in transaction data with similar data in the master file to verify that the account exists.

reasonableness test - An edit check of the logical correctness of relationships among data items.

check digit - ID numbers (such as inventory item number) can contain a check digit computed from the other digits.

check digit verification - Recalculating a check digit to verify that a data entry error has not been made.

DATA ENTRY CONTROLS Source documents should be scanned for reasonableness and propriety before being entered into the system. However, this manual control must be supplemented with automated data entry controls, such as the following:

- A **field check** determines whether the characters in a field are of the proper type. For example, a check on a field that is supposed to contain only numeric values, such as a U.S. zip code, would indicate an error if it contained alphabetic characters.
- A **sign check** determines whether the data in a field have the appropriate arithmetic sign. For example, the quantity-ordered field should never be negative.
- A **limit check** tests a numerical amount against a fixed value. For example, the regular hours-worked field in weekly payroll input must be less than or equal to 40 hours. Similarly, the hourly wage field should be greater than or equal to the minimum wage.
- A **range check** tests whether a numerical amount falls between predetermined lower and upper limits. For example, a marketing promotion might be directed only to prospects with incomes between $50,000 and $99,999.
- A **size check** ensures that the input data will fit into the assigned field. For example, the value 458,976,253 will not fit in an eight-digit field. As discussed in Chapter 11, size checks are especially important for applications that directly accept end-user input, providing a way to prevent buffer overflow vulnerabilities.
- A **completeness check (or test)** verifies that all required data items have been entered. For example, sales transaction records should not be accepted for processing unless they include the customer's shipping and billing addresses.
- A **validity check** compares the ID code or account number in transaction data with similar data in the master file to verify that the account exists. For example, if product number 65432 is entered on a sales order, the computer must verify that there is indeed a product 65432 in the inventory database.
- A **reasonableness test** determines the correctness of the logical relationship between two data items. For example, overtime hours should be zero for someone who has not worked the maximum number of regular hours in a pay period.
- ID codes (such as part numbers) can contain a **check digit** that is computed from the other digits. For example, the system could assign each new inventory item a nine-digit number, then calculate a tenth digit from the original nine and append that calculated number to the original nine to form a 10-digit part number. Data entry devices can then be programmed to perform **check digit verification**, which involves recalculating the check digit to identify data entry errors. Continuing our example, check digit verification could be used to

verify accuracy of an inventory item number by using the first nine digits to calculate what the tenth digit should be. If an error is made in entering any of the ten digits, the calculation made on the first nine digits will not match the tenth, or check digit. Note that check digit verification only tests whether an ID code in a transaction record *could* exist and is designed to catch transposition errors (e.g., entering 35689 instead of 35869 as a part number). A validity check is the only way to verify that the ID code really does exist.

The preceding data entry tests are used for both batch processing and online real-time processing. Additional data input controls differ for the two processing methods.

ADDITIONAL BATCH PROCESSING DATA ENTRY CONTROLS

- Batch processing works more efficiently if the transactions are sorted so that the accounts affected are in the same sequence as records are stored in the master file. For example, accurate batch processing of sales transactions to update customer account balances requires that the sales transactions file first be sorted by customer account number. A **sequence check** tests whether a transaction file is in the proper numerical or alphabetical sequence.
- An error log that identifies data input errors (date, cause, problem) facilitates timely review and resubmission of transactions that cannot be processed.
- **Batch totals** calculate numeric values for a batch of input records. Batch totals are used to ensure that all records in a batch are processed correctly. The following are three commonly used batch totals:
  1. A **financial total** sums a field that contains monetary values, such as the total dollar amount of all sales for a batch of sales transactions.
  2. A **hash total** sums a nonfinancial numeric field, such as the total of the quantity-ordered field in a batch of sales transactions. Unlike financial totals, hash totals have no inherent meaning. For example, it is possible to sum up the invoice numbers in a batch of sales transactions but the result is meaningless; its only purpose is to serve as an input control.
  3. A **record count** is the number of records in a batch.

sequence check - An edit check that determines if a transaction file is in the proper numerical or alphabetical sequence.

batch totals - The sum of a numerical item for a batch of documents, calculated prior to processing the batch, when the data are entered, and subsequently compared with computer-generated totals after each processing step to verify that the data was processed correctly.

financial total - A type of batch total that equals the sum of a field that contains monetary values.

hash total - A type of batch total generated by summing values for a field that would not usually be totaled.

record count - A type of batch total that equals the number of records processed at a given time.

ADDITIONAL ONLINE DATA ENTRY CONTROLS

- **Prompting**, in which the system requests each input data item and waits for an acceptable response, ensures that all necessary data are entered (i.e., prompting is an online completeness check).
- **Closed-loop verification** checks the accuracy of input data by using it to retrieve and display other related information. For example, if a clerk enters an account number, the system could retrieve and display the account name so that the clerk could verify that the correct account number had been entered.
- A transaction log includes a detailed record of all transactions, including a unique transaction identifier, the date and time of entry, and who entered the transaction. If an online file is damaged, the transaction log can be used to reconstruct the file. If a malfunction temporarily shuts down the system, the transaction log can be used to ensure that transactions are not lost or entered twice.

prompting - An online data entry completeness check that requests each required item of input data and then waits for an acceptable response before requesting the next required item.

closed-loop verification - An input validation method that uses data entered into the system to retrieve and display other related information so that the data entry person can verify the accuracy of the input data.

PROCESSING CONTROLS

Controls are also needed to ensure data is processed correctly. Important processing controls include the following:

- ***Data matching.*** In certain cases, two or more items of data must be matched before an action can take place. For example, before paying a vendor, the system should verify that information on the vendor invoice matches information on both the purchase order and the receiving report.
- ***File labels.*** File labels need to be checked to ensure that the correct and most current files are being updated. Both external labels that are readable by humans and internal labels that are written in machine-readable form on the data recording media should be