Kinetic Posting Engine Technical Reference Guide

Version 2025.1

Disclaimer

This document is for informational purposes only and is subject to change without notice. This document and its contents, including the viewpoints, dates and functional content expressed herein are believed to be accurate as of its date of publication. However, Epicor Software Corporation makes no guarantee, representations or warranties with regard to the enclosed information and specifically disclaims any applicable implied warranties, such as fitness for a particular purpose, merchantability, satisfactory quality or reasonable skill and care. As each user of Epicor software is likely to be unique in their requirements in the use of such software and their business processes, users of this document are always advised to discuss the content of this document with their Epicor account manager. All information contained herein is subject to change without notice and changes to this document since printing and other important information about the software product are made or published in release notes, and you are urged to obtain the current release notes for the software product. We welcome user comments and reserve the right to revise this publication and/or make improvements or changes to the products or programs described in this publication at any time, without notice.

The usage of any Epicor software shall be pursuant to an Epicor end user license agreement and the performance of any consulting services by Epicor personnel shall be pursuant to Epicor's standard services terms and conditions. Usage of the solution(s) described in this document with other Epicor software or third party products may require the purchase of licenses for such other products. Where any software is expressed to be compliant with local laws or requirements in this document, such compliance is not a warranty and is based solely on Epicor's current understanding of such laws and requirements. All laws and requirements are subject to varying interpretations as well as to change and accordingly Epicor cannot guarantee that the software will be compliant and up to date with such changes. All statements of platform and product compatibility in this document shall be considered individually in relation to the products referred to in the relevant statement, i.e., where any Epicor software is stated to be compatible with one product and also stated to be compatible with another product, it should not be interpreted that such Epicor software is compatible with both of the products running at the same time on the same platform or environment. Additionally platform or product compatibility may require the application of Epicor or third-party updates, patches and/or service packs and Epicor has no responsibility for compatibility issues which may be caused by updates, patches and/or service packs released by third parties after the date of publication of this document.

Table of Contents

Introduction

Purpose of this Guide

The Posting Engine Technical Reference Guide explores how Kinetic processes financial transactions for display within financial reports and the general ledger. When Kinetic installs, it uses a default series of posting rules to run this process.

You can however, create multiple books for different financial needs. Each book, in turn, can have a different set of posting rules which affect how the financial transactions it contains display within the general ledger.

You can modify the posting process for a book through two methods. If you need to make minor changes to this process, you should use chart of accounts mapping to link, or map, one source COA to a target COA. The target COA can then be updated as needed to satisfy the changes you need in the target book. Most posting process changes are minor, so you typically will use the COA mapping functionality.

If you need to make more complex changes to the posting process, however, you can create, modify, and delete posting rules. Kinetic uses a series of general ledger transaction types, like AP Invoice and AR Invoice Adjustment, to process and record financial (business) transactions. Each GL transaction type contains a series of posting rules, called a rule set, that define how transactions for the specific type post. You can make precise changes to the posting rules by create a revision within the GL transaction type. When you finish changing the posting rules, you activate the revision. This active revision is then used for all business transactions for the GL transaction type.

This guide explains how you can customize the posting logic to match the needs of a specific book.

Intended Audience

This guide is for individuals within your company responsible or partially responsible for accounts receivable, accounts payable, cash management, and any other financial areas of responsibility.

Employees who may find this guide useful include:

- Controllers
- Business Managers
- Accountants
- Financial Planners

How it is Organized

This guide first explores the structure of the financial functionality within Kinetic. It then explains the stages of the posting process and the various components which impact the financial results.

The guide then continues by documenting the transaction type environment within GL Transaction Type Maintenance, the primary application you use to modify and create posting rules. The guide concludes with a series of sections which details the GL transaction types.

The main sections of this guide:

- **The Posting Engine**- This section contains an overview of both the financial functionality and how this functionality interacts with the posting process.
- **Primary Components**-- The main records used during the posting process are documented in this section. These records include GL control types, business entities, chart of accounts, fiscal calendars, books, and more.
- **The GL Transaction Type Environment**-- This section explores the items you can modify and create within each GL transaction type. The purpose of selection criteria, variables, posting rules, posting codes, and so on is explored.
- **Posting Rule Reference**-- Use this section to review the primary items which you can modify to create new posting rules or modify existing ones.
- **Posting Rules in Action**-- Use this section to review examples which show how the posting engine arrives at its results.
- **Posting Engine Troubleshooting** -- Use this section to help with the understanding and troubleshooting of errors you may receive when creating posting rules.

The Posting Engine

The posting process is the core financial functionality within Kinetic. It first collects financial data and then evaluates this information to create appropriate general ledger (GL) transactions.

Depending on the current transactions and your chart of accounts structure, the transactions can post through one or multiple books to the general ledger. The functionality which runs this process is the Posting Engine. It provides a unified process for all business transactions, and it is a flexible rule-based system you can modify to define which accounts you want to populate for which amounts.

The posting engine functionality runs independently for each company created within Kinetic. A company can also have multiple books defined for specific financial requirements, and each book, in turn, can be mapped to a source chart of accounts (COA) or have a unique set of posting rules which you define. These account maps and posting rules indicate how the business transactions are posted as GL transactions to various accounts. You have complete control over the financial processing needed for each company. You can define the specific data that populates dynamic segments, create automatically self-balancing segments, and convert and round currency amounts based on up-to-date exchange rates.

The posting process runs through a number of stages, pulling input data from a series of generated business transactions (sometimes called business documents). It then uses the account maps and posting rules (sometimes called booking rules) created specifically for each book to post the results. Kinetic provides default GL transaction types that generate all business transactions which then post as GL transactions to the general ledger; you can extend these GL transaction types to include more data or alternate posting methods.

Your organization will have unique reasons for leveraging the posting engine. Some typical uses for this functionality include:

- If your company requires multiple books, like one for taxing purposes and another for accounting purposes, you can set up posting rules to reflect the different purposes of these books.
- If your organization works in a multi-company environment, you can set up unique posting rules for an intermediate book which transforms the financial data into a format usable by the receiving company.
- During company acquisitions, posting rules can be configured so that the acquired company can maintain their old books, fiscal calendar, and so on during the transition year.
- Your company may require multiple books for payroll to track employee costs in separate locations. These locations, for example, may require different currencies.
- If your chart of accounts uses dynamic segments, you can directly assign this dynamic segment in a GL control or use some other posting rule combination. Typically however, you do not assign a dynamic segment value in a GL control, so you will modify a posting rule

instead.

- If you customize Kinetic, use the posting engine to handle the automatic posting needs of the additional data you enter through these custom applications.

By default, Kinetic uses the posting rules originally defined for the 8.03 application. These standard posting rules insures that a base set of rules is always initially available within each company. The standard posting rules contain the department and division account segments used within the 8.03 Classic application. Later you can delete or modify these posting rules as you need.

If this is a first time installation, however, you will most likely install the extended posting rules. This set contains a more open series of rules which do not contain department and division hierarchies, so you have a flexible framework to begin reflecting the specific posting needs of your organization.

If your organization is upgrading but still wants to leverage the flexibility of the extended rule set, Epicor recommends you first set up a book using the standard rules to conduct your normal business transactions. At the same time, create a second book and import the extended rule set into it. Use this second book for a soft conversion process where you gradually post specific business transactions into it when users are ready to make the change. Your consultant can help you set up a soft conversion process so that your company can gradually leverage the extended rule set.

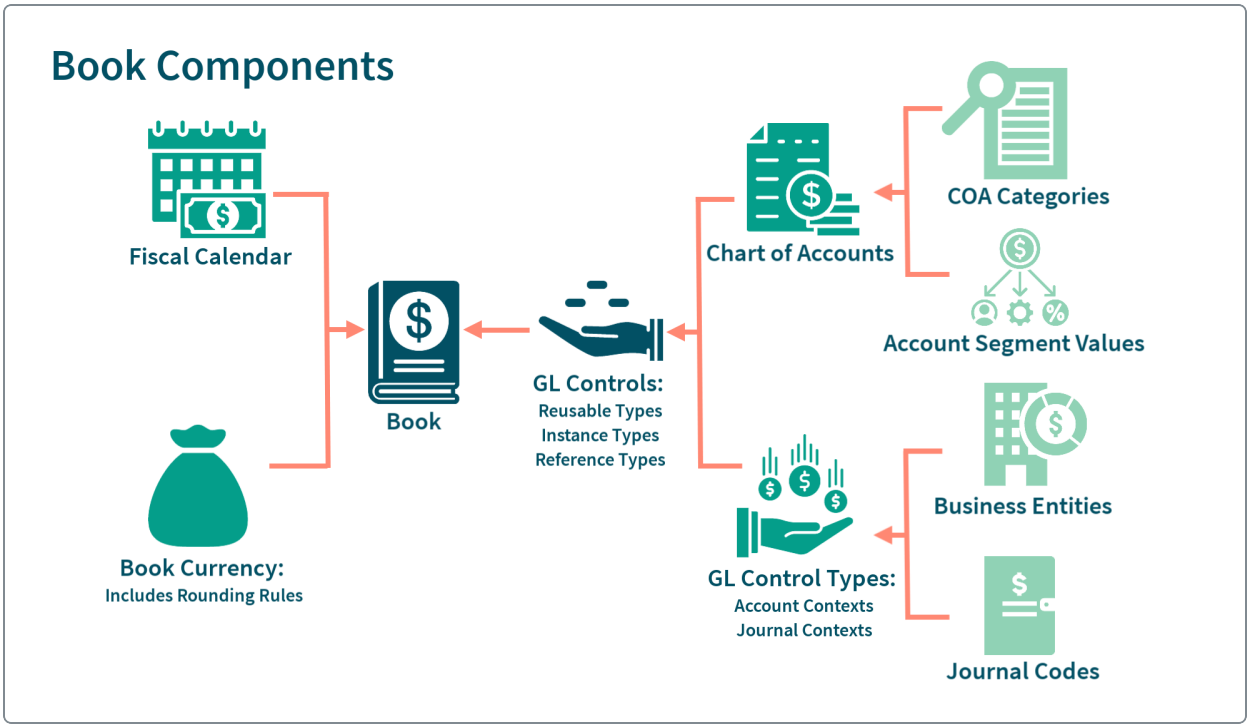
Regardless of the rule set installed, you can create new account maps, posting codes, posting rules, and other user extensions to the default GL transaction type structure. These modifications are called revisions, and they either are based on an original rule structure or on a unique rule structure you define. You have complete control over these posting rule revisions. To help you during the development of your posting rules, you assign status levels to them in order to control when the revisions are implemented on your financial records.

By leveraging this powerful toolset, you can set up the posting processes you need to create accurate financial records which address the internal realities of your organization and the legal requirements of the localities inside which you conduct business.

Book Components

This section describes how various components are used with each book.

Review this graphic and the brief descriptions which follow to become familiar with the book functionality:



The following list briefly explains how each item interacts with a book. Use this list to gain a beginning understanding of each item; all of these components are described in more detail later in this guide within the Primary Components section. Book components:

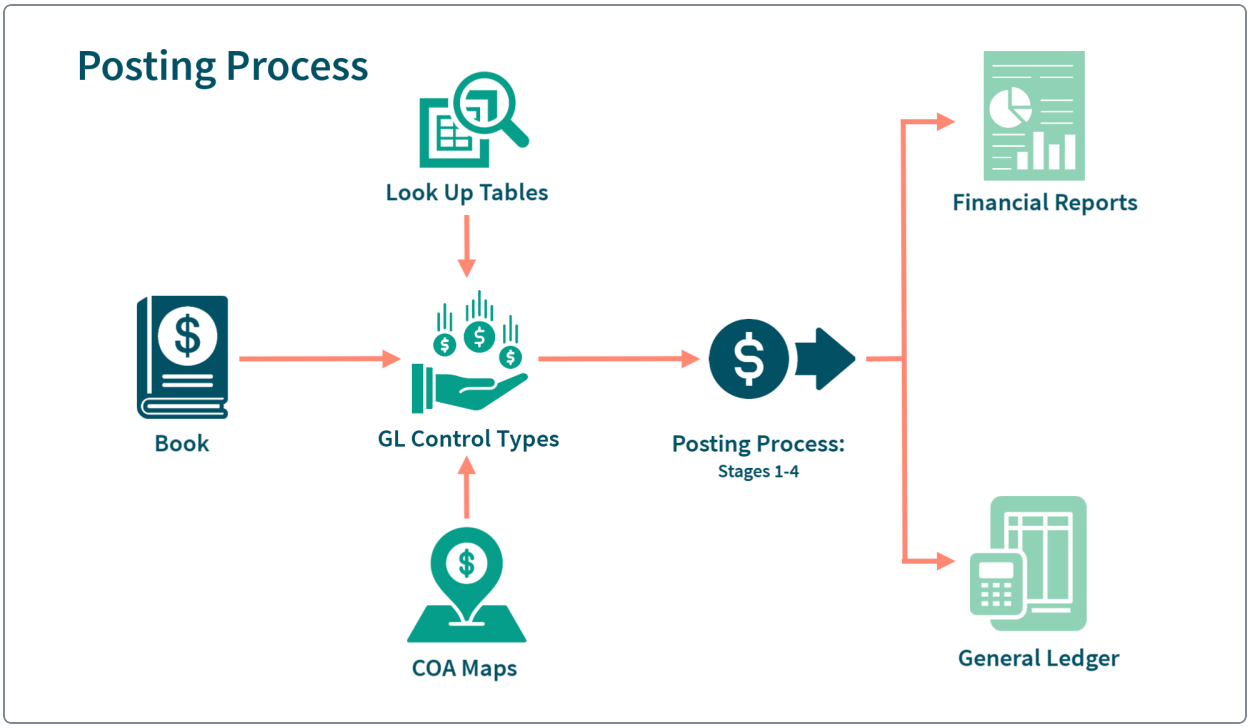
- **Account Segment Values**- These components define values for both dynamic and controlled segments. You can also use them to define options for natural account values and indicate the effective dates for these segment values. Account segments are applied to GL control types as account contexts.
- **Books**- A component which contains a unique set of financial records created for a specific purpose, like budgeting or alternate currency calculations. Each company can have as many books as it needs. A book defines the currency, chart of accounts, and fiscal calendar used to process its financial transactions; it can also map to a source COA and contain a unique set of posting rules.
- **Business Entities**- Business entities define various items within the company, like Customer, Supplier, and Part. These static tables contain the data related to these items. You can select business entities on GL control types.
- **Chart of Accounts**- These components define a set of accounts used to organize the financial results of business activity within a company. The posting engine places the GL transactions in the accounts specified for the COA you select for a specific book.
- **COA Categories**- Categories link natural accounts to specific values you define with Account Segment Values Maintenance. They group accounts which track similar functions. Use these

items to determine how balances are maintained for linked accounts and the structure and form of financial statements generated for a specific chart of accounts.

- **Currencies**- You must select a specific currency to use with the transactions generated through a specific book. Each currency record also contains the values it needs for rounding amounts on specific transactions.
- **Fiscal Calendars**- A component which determines the periods during which GL transactions post during a fiscal year. Each book can have a unique fiscal calendar assigned to it.
- **GL Control Types**- A component which groups account contexts, journal contexts, and business entities. GL control types are templates for GL controls. GL control types determine the overall account and journal contexts used for transactions with a selected GL control. They determine the business entities used with each GL control to generate their specific account information. You select the GL control type you need on each GL control. The GL controls use these contexts and business entities to define the specific account string and journal code to use with each GL transaction.
- **GL Controls**- A component which defines the account strings and journal codes available to a specific record (part, customer, marketing campaign, and so on) during the posting process. These items identify each account through a string value which then displays within financial reports and trackers. GL controls are used in three ways. Reusable GL controls are controls users link to setup (maintenance) records; only reusable GL controls use business entities. Instance GL controls are used with specific items within the database; the instance GL control links a unique account (like a purchase order account) to the specific item. Reference GL controls are hidden from users; they populate the TranGLC table with system records.
- **Journal Codes**- Journal codes define the codes used to group journals. These codes help you track and report on journals, and they are applied to GL control types as journal contexts.

Posting Process Components

This section describes how various components work together to post financial transactions. Review this posting process graphic and the brief descriptions which follow to better understand how the posting engine works with other components within Kinetic.



Posting process components:

- **Book**- A component which contains a unique set of financial records created for a specific purpose - like budgeting or alternate currency calculations. A book defines the currency, chart of accounts, and fiscal calendar used to process its financial transactions. It contains the transactions generated by financial activity for the book.
- **Chart of Account Maps**- You use COA maps (or account segment maps) to automatically transform one chart of accounts to another chart of accounts during the posting process. These items link, or map, accounts and journal codes across multiple books. The posting rules within GL transaction types use these maps to transform the COAs into the correct structure. You typically use COA maps when you only need to make minor posting changes between two books.
- **GL Transaction Types**- Transaction types define all the posting codes, functions, pre-posting rules, posting rules, reference rules, and other items required to pull in a business transaction and post its data as a GL transaction. Kinetic installs with a series of GL transaction types you can access through GL Transaction Type Maintenance. You create revisions for each GL transaction type and then create, modify, and delete posting rules and other items as you need. You typically use this functionality when you need to make more complex changes to the posting process for a specific book.
- **Lookup Tables**- These components link any number of source fields to any number of target values. These target values can be a segment, an account, or some other value. They pull in specific data required for the posting rules.

- **Posting Process**- This four stage process pulls in the financial transactions from a specific book and records it using the COA maps, default posting rules, and modified posting rules contained within GL transaction types. This complex process can be run manually whenever you need or automatically through a schedule you define for each entry application which posts to the general ledger.
- **Financial Reports**-- The resulting financial transactions can be viewed on various financial reports which display the posted transactions.
- **General Ledger**-- The company general ledger records the financial transactions. It uses the account structure and calculations defined for the book within its GL transaction types.

Posting Engine Elements

This section of the guide describes the primary elements which make up the posting engine process.

Available Modes

The posting engine process runs in the following modes:

- **Pre-Post Mode**- This mode is used when some accounts must be calculated before actual posting occurs. This mode calculates and outputs data used later during the posting process. Pre-posting rules pull in the data required for each GL transaction type.
- **Validation Mode**- Use this mode to test your posting rule modifications. This mode generates entries for the Review Journal; use this application to preview the GL transactions generated in books when you post source business transactions. The transactions are not yet posted to the general ledger, so you can make changes to the transactions, COA maps, and posting rules as you need. You can also print edit lists and use the Review Journal to view the results. When you are satisfied with the results in the Review Journal, you can then post them from within this application out to the general ledger.
- **Open Invoice Mode**- Run this mode when you use the Open Invoice Load application in either AP or AR. This mode populates reference GL controls without creating GL transactions. You use Open Invoice Load when you are first pulling in financial data from another application. After you post the invoices, you then use these reference GL controls with your current financial data.
- **Regular Posting Mode**- This mode posts a source business transaction (or document) to create a GL transaction. The posting engine process uses this mode when a user enters a record within an entry application. After the posting process is complete, the source business transaction is marked as posted. This mode can output calculated accounts for just GL transactions or both GL transactions and GL controls.

Posting Engine Process

The posting process can be run immediately against a group of transactions (for example, AR invoices) or run through a regular, automated schedule on the system agent. To leverage system

agent functionality, you first create the recurring schedule you need within System Agent Maintenance and then select this schedule on the specific posting process you need.

When the posting engine runs, it assigns the following status levels to business entities:

1. Kinetic first blocks adjustments to the corresponding business entity and designates the entity as in process for being posted.
2. Business transactions (or documents) generate for this business entity.
3. When the posting process completes successfully, each business transaction is defined as posted and the result GL transactions are available for display. All tracker applications display the posted status for the business transactions.
4. When the posting process fails, however, the Review Journal displays both the errors and warnings for the transactions. At this point, you can cancel the results, which indicates that the business transaction is unposted. You can then adjust the source business transaction and re-post them.

Depending on how your application is set up, the transactions post to a single book or multiple books. The journals uses the chart, currency, and fiscal calendar from the posting rules defined for the GL transaction type which generates the business transaction. When transactions post to multiple books, you must set up either COA maps or posting rules to indicate for which books the transactions post. Likewise, you can also set up posting rules for Red Storno and reversing journals.

GL Transactions

As described previously, when the posting engine runs, it creates GL transactions for one book or multiple books. A single input document, defined within a GL transaction type, creates a single GL transaction in every affected book.

You modify the COA maps and/or posting rules needed to process the transactions for the methods required in each book:

- **COA Maps** are user-defined items that link the source chart of accounts from one book to the target chart of accounts of another book. The posting engine then creates a GL transaction in one book by mapping the same GL transaction from another book. A GL transaction created in one book through a mapping table cannot be applied again later by a GL transaction type. You create COA maps to handle minor differences in the posting processes for separate books.
- **Posting Rules** are either default or user modified rules; they can a sequence of instructions on how to generate a GL transaction within a specific book using the data from the input business transaction. You can add, edit, and delete the posting codes, functions, posting rules, and other items needed to post business transactions for each GL transaction type. You modify posting rules to handle more complex changes required in the posting process for a single book or multiple books.

In some cases, the posting engine process also handles rounding differences that occur after converting currencies, auto-balancing segments, and validating transactions. You can configure these procedures to match your business environment, and Kinetic executes them as automatic procedures.

The primary components which control the posting engine process are the GL transaction types. Each GL transaction type gathers the data input required for the posting engine process. Kinetic contains a series of GL transaction types which it uses as default templates for generating GL transactions. You can modify these transaction types as you need within GL Transaction Type Maintenance by creating revisions. You can modify nearly every aspect of a revision and then use it as the default for the GL transaction type.

Each business transaction belongs to a specific GL transaction type. Kinetic supports various types of business transactions, such as inventory issues and payments. Because of this, Invoice Payment is a GL transaction type, while a specific payment by check number XXX for the invoice number YYY is considered a single business transaction. The GL transaction type describes the entire business transaction and provides data that determines how GL transactions are constructed for this type.

The structure of a GL transaction type is unique. Each has a different number of rule levels, and posting codes and amounts are also unique at each level. This structure enables each GL transaction type to correctly reflect the business transaction. The standard structure also provides a common way for posting (also called booking) rules to access data.

Each GL transaction type has a rules structure which contains:

- A **Header** posting rules level

  Only one header rule is available for each GL book.

- An extensive number of **Detail** posting rules levels.
- **Functions**- These rules define various operations which are used across all of the posting rules for the specific GL transaction type. They include the rules required to populate account segments for divisions, departments, and charts.
- An optional **pre-posting rules** level which contains rules that populate the data needed before the GL transaction type runs.

Each level contains posting code and amount elements. During posting, posting rules use data pulled from these posting elements, depending on the rule level for which the posting element is defined.

- **Amounts**- These values are a collection of named amounts available at a specific level of the transaction type hierarchy. For example, an AR Invoice transaction type contains an Extended Price and Discount Amount which are available at the Invoice Line child level.
- **Posting Codes**- Posting codes are grouped in posting entities. Each posting entity is data used by the posting rules; each posting code within the entity usually corresponds to a field within that data table. For example, the Part business entity may contain a group of fields from the Part table; these fields are useful when Kinetic creates a GL transaction for the given business event.

You can define posting rules that use additional data by modifying the transaction type which generates the document. To do this, you extend the transaction type templates by first adding new

elements, like posting codes, functions, operations, logical conditions, and so on. You can then construct the posting rules which access these new elements.

By modifying the transaction types within GL Transaction Type Maintenance, you can customize the posting process to generate GL transactions that reflect the business needs of your company. These transactions display as GL journals.

You can quickly navigate from the current posting rule or function to the user-defined function definition by clicking the **Go to Function Definition** selection from the Context menu. A rule or function operation which calls a user-defined function is marked with a specific icon  . You can go back to the posting rule or function by clicking the related link displayed on the user-defined function's operation tab.

Multi-Company Considerations

The posting engine does not interact directly with the multi-company functionality. When the engine runs, it posts transactions to any accounts defined as a multi-company account.

Then at a later time, the multi-company functionality pulls this account information from the source company and updates it within the target company.

This process works the same if you require an intermediate book in order to transform the amounts from a currency used by the source company into the currency used by the target company. In this case, the posting engine runs and stores the transactions in the intermediate book. At a later time, the multi-company functionality pulls the account information from intermediate book at the source company into the corresponding intermediate book within the target company. These amounts then update the target book within the target company.

Output Data

The posting process creates GL transactions and calculated accounts; these transactions and accounts are used in subsequent processing.

During the posting process, business transactions (sometimes called source documents) are evaluated by the posting engine. The process then creates, or outputs, the results into the multiple locations you define through mapping and posting rules. A single business transaction can generate a number of successive GL transactions in all books configured to receive them. For example, one AR invoice detail line can post to multiple books, and the GL transaction can be placed into different accounts within each book.

The posting process can output:

- A GL transaction
- A number of calculated accounts
- Both a GL transaction and a number of calculated accounts

The calculated accounts are stored within the TranGLC table within the database. These accounts then display in the respective books which generated them during the posting process.

This process also outputs single or multiple calculated accounts and returns them as GL controls. A GL control is an item which stores a set of account strings and journal codes available to a specific business transaction during the posting process. You define all the specific account strings used during posting within each GL control. These primary components are discussed in more detail later in this guide.

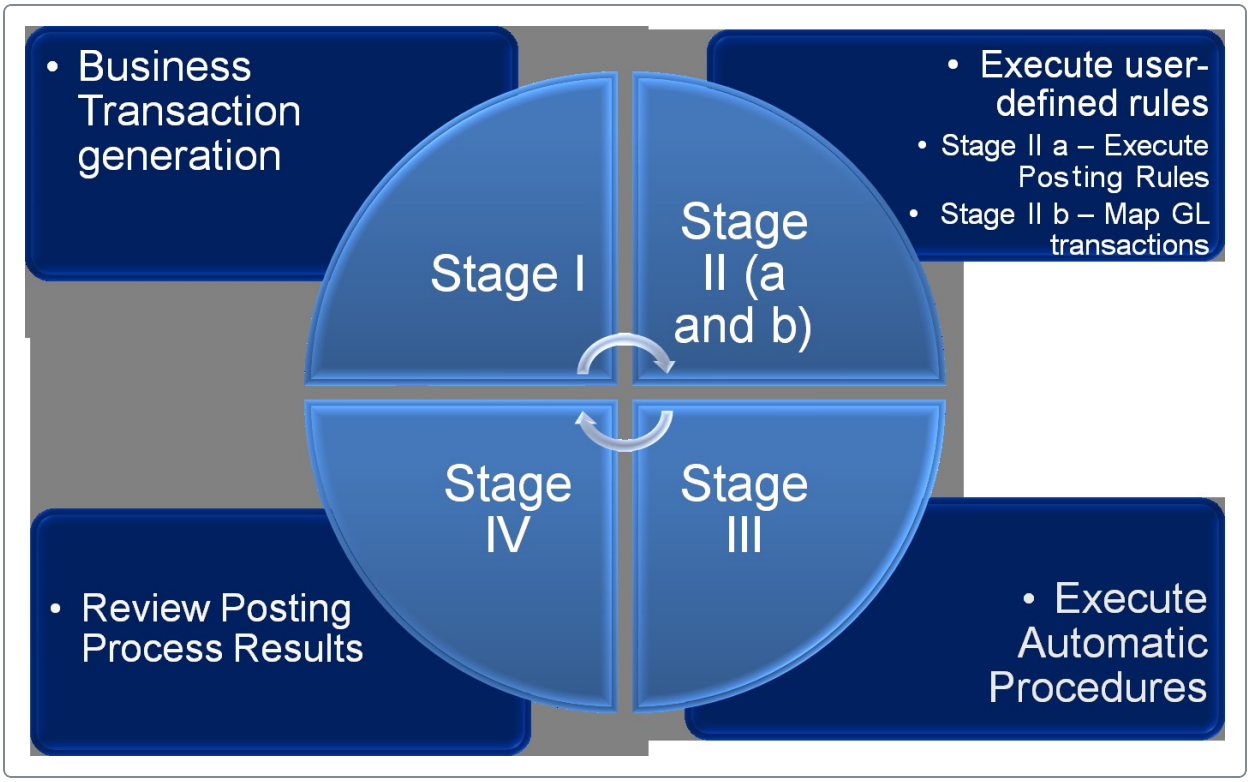
As an output data example, you create a payment on an invoice where the payment business transaction affects the same AR account calculated in the previous invoice business transaction. Another example is when you use a pre-posting rule, which is used by the posting process to calculate a default account overridden before the source business transaction posts to GL.

Posting Engine Process Stages

The Posting Engine runs its process through a series of stages. Each stage builds on the data generated during the previous stage.

The posting process stages:

- Stage 1: The business transactions generate.
- Stage 2: Both default and user modified posting rules execute. This two-part stage first runs posting rules and then maps the corresponding GL transactions.
- Stage 3: The automatic procedures execute.
- Stage 4: You can review the posting process results. An optional stage, you typically perform this stage when you need to validate modified maps/rules or when posting errors occur.



The next sections contain more detail on what occurs during each processing stage.

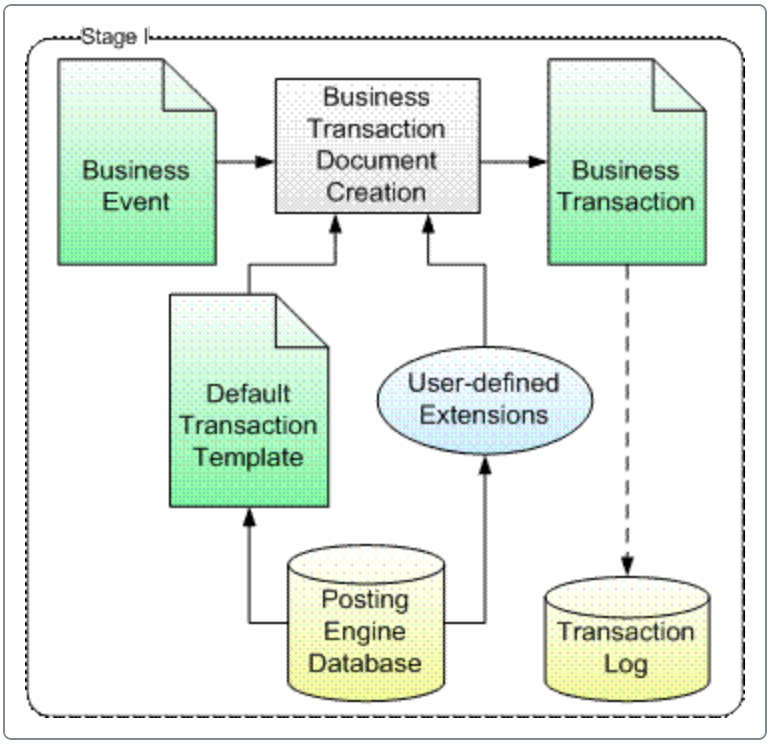
Stage 1 Business Transactions Generate

The posting process begins when Kinetic creates a business transaction for a specific business event. A business event type is first detected and then an incoming template loads for this event type.

The loading process first takes the default structure and then extends it with any modified posting elements (posting codes, functions, and so on) you may have created.

Next, the template populates with the business event data. It becomes a business transaction which will be processed during the subsequent posting stages.

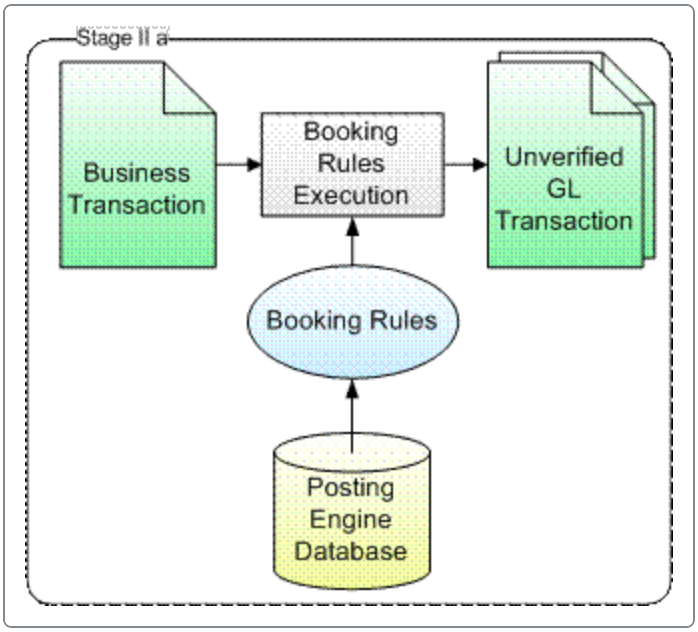
The following illustration shows the process flow for creating business transactions:



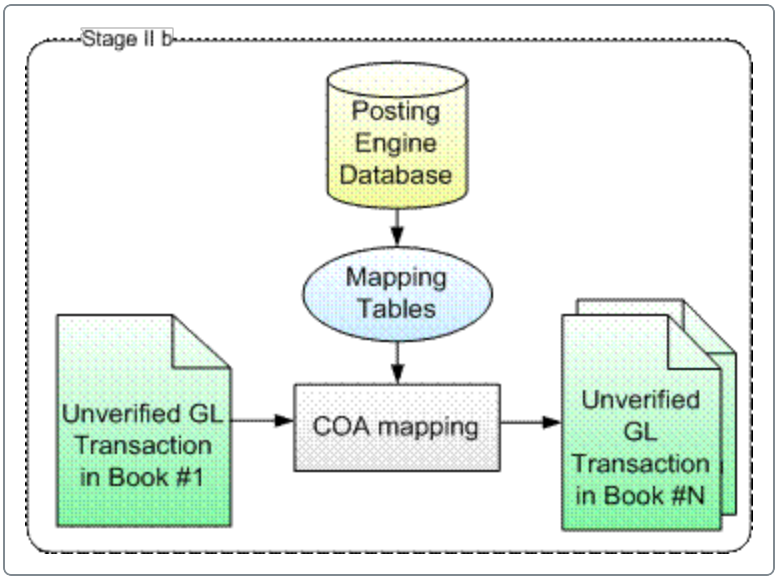
Stage 2 Run Posting Rules

During this stage, the business transaction is then used to generate GL transactions in the company books which record this business transaction. The GL transactions are generated through two methods.

The first method (Stage IIa) directly applies posting rules to the incoming business transaction. You define posting rules for books separately so that each book can have its own rule set for every GL transaction type. Each rule constructs a single GL transaction detail or a pair of transaction details which balance each other, creating an entire GL transaction. When this stage is complete, the GL transactions for the source business transaction generate in all books which have posting rules that handle the specific business transactions. You use this method when you need to make more complex changes to the posting process.



The second method (Stage 11b) maps GL transactions generated by the posting rules from one source book to one or more target books. When a GL transaction is created for a source book, it can also be mapped to other target books. You map a source book to target books when you can reuse the same set of posting rules to construct GL transactions in other books. Use this method for minor posting changes.

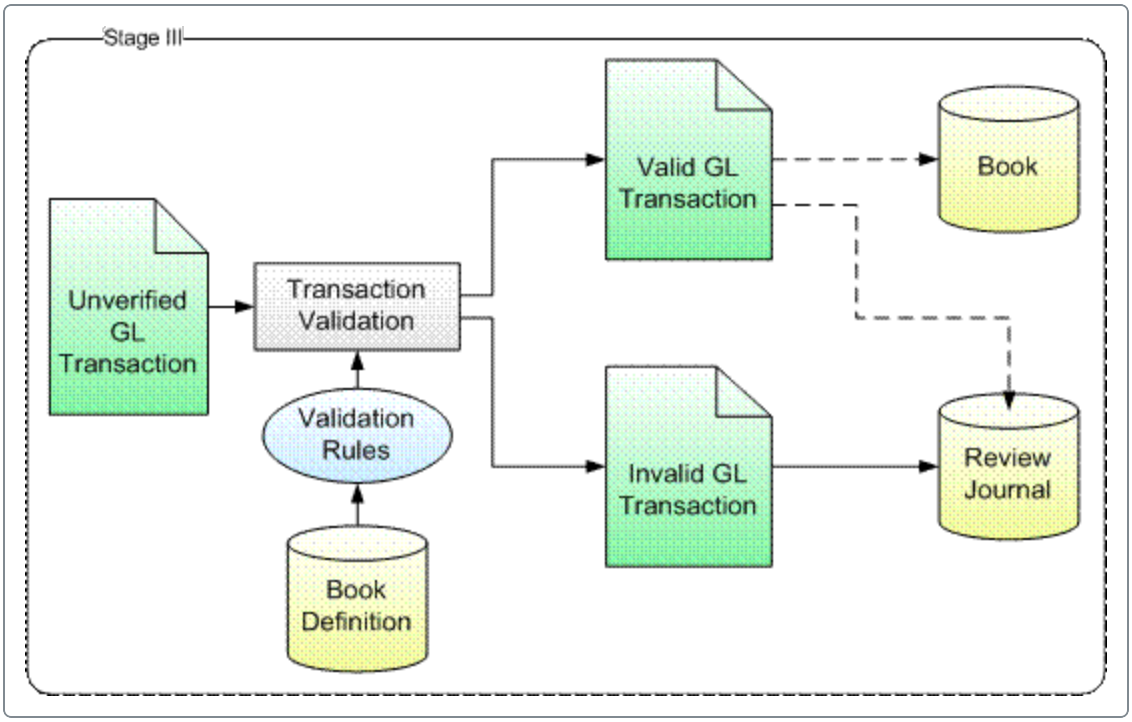


GL transactions are not validated (verified) during this stage. They may be unbalanced, may contain invalid accounting segments, or invalid combinations of segments in the account.

Stage 3 Run Automatic Procedures

During this stage, several procedures are applied against unverified transactions in all affected books.

These procedures validate the transactions generated through Stage 2. When Stage 3 completes, the GL transactions either update the book journals and account balances or are posted to the Review Journal - where you can adjust, validate, confirm, or cancel them.



The procedures run in the following order:

1. **Balance Validation**- Verifies that the transaction total debit equals total credit. If the process cannot complete the balance validation, the Posting Engine raises a logic error.

2. **Account Validation**- Verifies that all accounts are valid. This process is a complex review which includes verifying an account exists and is active, its combination of segments is valid, and that all mandatory segments are specified. If an account is invalid, the Posting Engine raises a logic error.

3. **Transaction Summarization**- An optional process which runs only if you summarize the GL transaction type. When you activate the summary process, all transaction details which affect the same account summarize using one of the following methods:

    - Kinetic replaces its details with a single detail. This detail amount is calculated by subtracting the total credit amount from the total debit amount for all the summarized lines. If a negative amount calculates, the resulting detail credits the account; if a positive amount is calculated, the resulting detail debits the account. Note that even if a

negative amount is calculated, the resulting detail contains it as a positive amount-- but in the Credit field.

- Debits and credits generate separately by Kinetic and will create one debit detail and one credit detail. It creates the debit details by summing up all the details that debit the account; it creates the credit detail by summing up all the details that credit the account. Note that the summarized credit or debit detail is created only if the sum is a non-zero amount. The original lines are kept separately as details of the summarized lines maintain the integrity of the references. This process is similar to what is done for summarized inventory transactions.

4. **Automatic Posting of Rounding Difference**- All rounding calculations, both the original amount and the rounding difference values, are captured by Kinetic before the posting process. Then when the process runs, the posting engine updates both the rounded amount and the difference amount per each detail line (for example, on a sales order). Then these values map to a total account which sums the amounts on each detail line.
5. **Self-balancing Segments** - This procedure makes sure that all accounting segments defined as self-balanced are actually balanced. If unbalanced segments are detected and the Posting Engine is configured to automatically balance such segments, it creates additional lines to balance these segments. If the automatic balancing of accounting segments is not enabled, the Posting Engine generates a logic error.
6. **Final Validations**- Several validation rules defined on each book are applied against the GL transactions in order to verify each transaction. If a validation rule finds that a transaction is invalid, it either generates a logic error or a warning, depending on the setup of the rule.

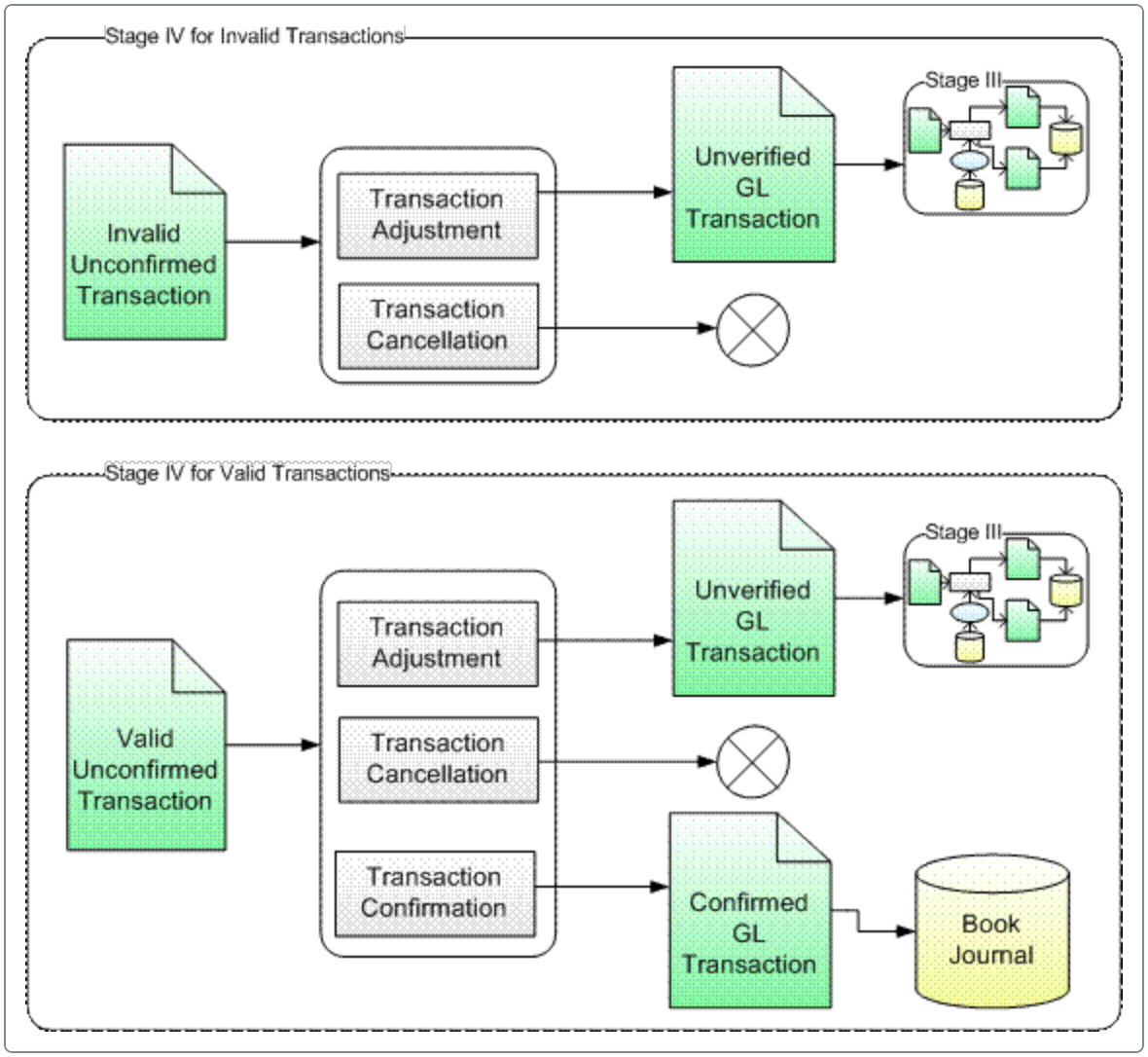
Each GL transaction in the affected book is eventually evaluated as either a valid or an invalid transaction. Valid transactions update the balances in books, while invalid transactions are sent to the Review Journal for manual processing. Note that if a transaction is evaluated in one book as invalid, the entire posting session is then considered invalid and the transactions go to the Review Journal.

Stage 4 Transaction Review

This optional stage occurs when a business transaction produces an invalid GL transaction or when you indicate that valid GL transactions can also post to the Review Journal. You activate the second procedure when you either want to manually confirm posted journals or when you need to verify that modified posting rules run correctly.

Note however, that the Review Journal displays the posting results; the GL transactions created in validation mode appear in the edit list for the specific entry application (for example, AR Invoice Entry) and not in the Review Journal.

During this phase, Kinetic stores the GL transaction in an intermediate journal, called a transaction review journal . You can review, adjust, cancel, and confirm transactions while they are in the Review Journal. Confirmed transactions become permanent journal transactions and later post to GL.



The only correction you can make directly in the Review Journal is to change the account code. All other changes must be made in GL Transaction Type Maintenance or within the entry record itself. When you have made the changes, run the posting process again (Stage 3) and evaluate the results in the Review Journal.

Note that when you are satisfied that the posting rules work correctly, you can indicate on the revision that you no longer want transactions to automatically display in the Review Journal. Then valid transactions will post straight to the book and only invalid transactions are captured by the Review Journal.

Primary Components

The components required for configuring the posting engine -- like books, business entities, and lookup tables -- are explored in the following section. Many of these components were explained briefly in the previous Book Components and Posting Process Components section, and are now described in more detail here.

Account Segment Values

Account Segment Values define the value you need for both dynamic and controlled account segments. You can also define options for natural account values and indicate the effective dates for these segment values.

To begin using this functionality, you first create the segments you need within Chart of Accounts Maintenance.

When you finish defining the segments you need for a specific COA, you then launch Account Segment Values Maintenance to define the values you want for each segment. Use this functionality to define:

- **Natural Accounts**-- Natural account values typically reflect the structure of the financial statements for the current company. You can associate COA categories with natural accounts. This association determines how Kinetic maintains account balances, as Kinetic zeros balances in income statement accounts and maintains balances for balance card accounts. You can also determine whether journals which use this account segment are summarized - reducing the number of records stored within the database. A natural account segment can also contain currency revaluations which occur when a transaction currency is different from the reporting currency. When reports are run, Kinetic captures exchange rate gains and losses for natural account segments which have currency accounts linked to them. Lastly, you can prevent manual journal entry for a specific natural account segment.
- **Controlled Segments**-- These segments define a specific business activity within the current company. The controlled segment values you define in this application are available when users enter accounts in applications such as AP Adjustment and Cash Receipts Entry.
- **Dynamic Segments and GL Reference Types**-- You can associate a dynamic segment with a GL reference type. You use reference types to define items within the current company that have a limited life span, like marketing campaigns and projects. When you associate a reference type with a chart of account (COA) segment, you can select a segment value for a posted transaction.
- **Effective Dates**-- You define a date for a segment to indicate when the segment value is no longer active. The COA can then reflect changes. For example, you can set a segment used for a product line to expire on the date when the product line is discontinued.

You link a segment to a business entity to use its values as dynamic segment values. Business entities define tables within Kinetic which hold primary data like customer and supplier records. You can select these entities for a dynamic segment.

Application Location and Modifiers

The following section details the location(s) where you can access the Account Segment Values functionality from the Main Menu. It also describes the values you can change for this item.

Use the Account Segment Values Maintenance to create the segment values you need for a specific chart of account. To launch this application from the Main Menu:

- Financial Management/General Ledger/Setup/Account Segment Values

Modifiers you define in this application:

- **Accrual Account**- A natural account option, this value defines the account to which reversing entries post. If no account is specified, reversing entries created by the exchange post to the currency account.
- **Allowed**-- A natural account option, this check box indicates that the currency account can keep the balance using the value in the specific currency.
- **Blocked**-- A restricted function option, this check box determines whether the natural account can appear in manual journal entries processed by the function. When you select this check box, it excludes the account from displaying within manual journals.
- **Category**-- Defines the COA category you want linked to the current account segment value. This value determines how Kinetic maintains both account balances and the accounts which provide financial statement values.
- **Credit**-- Defines the currency exchange rate used to revalue amounts posted to credit accounts. Typically, accounts payable and other credit accounts use a buy rate.
- **Currency Account**-- A natural account option, this designates the natural account as a currency account. When you select this check box, Kinetic converts transaction currencies to a selected reporting currency. Financial reports can then reflect exchange gains and losses resulting from revaluation at the time the report is run.
- **Debit**-- A natural account option, this value defines the currency exchange rate used to revalue amounts posted to debit accounts. Typically, accounts receivable and other debit accounts use a sell rate.
- **Effective From Date**-- Indicates the date from which the segment value can first be used for posting. By defining an Effective From Date, you cause the COA to reflect business changes in the current company. However, if you do not enter a date in this field, you can use this value to post the segment immediately.

- **Entry Control**-- Used for natural accounts, these values defines how the segment is used with the natural account. Available options:
    - **Mandatory (Controlled Segments)**-- When you select this check box for a controlled segment, users must enter and use this segment in any account codes which use the natural account.
    - **Mandatory (Dynamic Segments)**-- When you select this check box for a dynamic segment, users must enter and use this segment in any account codes which use the natural account. The GL control which uses the account code also requires that a business entity must be linked to the segment.
    - **Optional**-- Select this check box to indicate that this segment is available, but not required, for using the natural account.
    - **Not Used**-- Indicates that the current segment is not used with the natural account.
- **Function**-- A restriction function option, this value identifies the function itself. You create the restricted functions which display on this drop-down list by using Restricted Function Maintenance.
- **Gain Account**- A natural account option, this value defines the account used to capture gains occurring from currency exchanges.
- **Loss Account**- A natural account option, this value defines the account used to capture losses occurring from currency exchanges.
- **Normal Balance**-- Indicates whether the current account segment generates debit or credit values. You can only select this value for natural accounts.
- **Reference Type**-- Indicates the GL reference type used to group values for the segment. These account segments are then associated with the GLCOARefType business entity within Chart of Account Maintenance.
- **Revalue Option**-- A natural account option, this check box designates whether Kinetic captures gains, losses, or both when exchanges involve the currency. When you select this check box, revaluation occurs.
- **Summarization**-- A natural account modifier, this check box determines whether Kinetic summarizes the journal details when a transaction posts to an account which uses this segment. When you summarize an account segment, this reduces the number of records stored in the database. You have the following options:
    - **Summarize**-- Posts a single journal which summarizes the debits and credits for the account.
    - **Summarize Debits and Credits Separately**-- Posts one journal that summarizes the

account's debits and one journal that summarizes the account's credits.

- **Not Summarize**-- Posts journal details to the account.

- **To Date**- Defines the final date on which the segment value can no longer be used for posting. By defining a To Date, you cause the COA to reflect business changes in the current company. However, if you do not enter a date in this field, you can use this value to post indefinitely.

Logic/Algorithms

The Account Segment Values functionality uses this logic to calculate its results.

Account segment values define the account contexts used on GL control types.

Example(s)

The following example(s) illustrate how you use the Account Segment Values functionality.

You need to create a controlled segment that designates company divisions in the account codes. To do this, you define the values TOOLS, SERVICES, FINANCE, and OEM for the segment. The controlled segment values are then available when users enter accounts in applications such as AP Adjustment and Cash Receipts Entry.

Books

Each company can have one or multiple books, and each book can in turn have a different currency, chart of accounts, and fiscal calendar.

These unique values are then applied against journals which post to the specific book and these values are also used when you generate financial reports within the book.

Books can either record financial transactions or consolidate other books. A standard book records the financial activity of a company. Consolidation books, however, provide a preview which displays the financial results when multiple books are combined within the source company.

The fiscal calendar for the book determines valid apply dates. Kinetic maintains periodic balances for segments based on the fiscal calendar.

The currency used to post to the book displays on financial reports and in consolidations. All journal amounts post in the book currency. Books also store journals for currency accounts using transaction currencies instead of the book currency. You use these accounts as you need for reporting.

You can also use the same COA by mapping it between two or more books. Each book can also, however, use a unique set of chart of accounts. The COA defines valid general ledger accounts, available dynamic segments, and balance maintenance for any segments other than the natural

account (The natural account maintains continuous balances). The book defines the retained earnings account.

You can also implement validation rules for each book to correct posting errors which may occur. You can cause the validation rules to raise a warning, automatically correct an issue, or ignore the error condition. When you post a group (or batch) through an entry application, Kinetic blocks the posting of any invalid journals and continues to process the remaining journals in the group. You investigate error transactions in the Review Journal.

Typically, a company uses multiple books to display the same financial information for multiple purposes. Use multiple books to value items differently in financial and statutory reports. For example, an insurance company might use GAAP to value investments and other items for one report, but use National Association of Insurance Commissioners guidelines within another book. State, province, or other governing localities might also impact your financial reporting, requiring multiple books.

Before you implement multiple books, carefully consider the role each book plays in the financial management and reporting within the current company.

New Book Setup

Whenever you create a new book, you can either set up new posting rules for the book or use mappings instead of the posting rules. When the two books share the same COA, you can use COA Mapping to link a COA from a source book into the new target book. Because the new book uses the same posting rules, no changes need to be made to them. Typically when a COA is pretty much identical between books and the accounting methods are also the same but use different currencies, fiscal calendars, or level of details, consider using mappings instead of posting rules. The mappings are easier to maintain than user modified posting rules.

If you need to create more complex posting process changes, however, you can create these changes for a specific book by modifying GL transaction types. For example, if you conduct business in Russia, you could set up the posting rules to accommodate Red Storno accounting. The posting engine produces GL transactions within each book based on the unique GL transaction type revisions defined for each book.

You can change a book's settings until a journal is posted to its COA. After a journal posts to the COA selected on the book, only the book description can change.

Retained Earnings

You leverage the Retained Earnings functionality to designate the standard account used for retained earnings for each book. Kinetic continually updates the retained earnings account balance with postings to the revenue and expense accounts for the selected COA. Because of this, the balance card always reflects the current balance. At year-end closing, Kinetic transfers account balances to the next fiscal year.

Balance cards report the balance in the standard retained earnings account. You can further divide the reported retained earnings balance by substituting a segment in the retained earnings account for a corresponding segment in the income statement. You can then split retained earnings by division, department, or any other mandatory controlled account segment.

Account masks designate the relationship between the split segment and the retained earning accounts that displays the balances. Often, a book that uses segment substitution also uses self-balancing segments. When you designate a self-balancing segment in Self-Balancing Segment Maintenance, you cause the balancing journals within a segment to automatically balance. This makes sure the book to which the journals post maintains a balanced set of records within the account segment.

Application Location and Modifiers

The following section details the location(s) where you can access the Books functionality from the Main Menu. It also describes the values you can change for this item.

Use Book Maintenance to create new books or modify existing books.

Navigate to this application from the Main Menu:

Menu Path: Financial Management > General Ledger > Setup > Book

Modifiers you define in this application:

- **Account Mask**-- A Retained Earnings modifier, this value creates an account mask so the account's balances can transfer to multiple retained earnings accounts. Kinetic substitutes the segment in the income statement account for the corresponding segment in the retained earnings account.
- **Action**-- A validation rules modifier, this value defines how Kinetic handles invalid journals that post to the book. Available options:
    - **Error**-- Blocks posting of the journal. You can view the transaction in the Review Journal.
    - **Warning**-- Causes the journal to post, but a warning message appears in the error log.
    - **Ignore**-- Causes the journal to post, but no entry appears in the error log. This option is the default setting for most posting errors.
    - **Autocorrect**-- Select this option to cause Kinetic to use a pre-defined process to correct the error. For example, if a journal has an Apply Date which occurs in a closed period, Kinetic changes the Apply Date to the current period. A warning message displays in the Review Journal detailing the automatic change.
- **Book Currency**-- Select the currency you wish to use with the book from this drop-down list. The book currency can be used on financial reportsand in consolidations. All journal amounts

post in the currency you select on the book. For multi-currency reporting, books can also store journals using transaction currencies as needed for multi-currency reporting.

- **Calendar**-- Select the fiscal calendar you wish to use with the current book. These calendars define the fiscal periods used by Kinetic to record transactions. You create fiscal calendars within Fiscal Period Maintenance.
- **Chart of Account**-- Defines the chart of account (COA) you wish to use with the book. Each book can have a different COA or several books can share the same COA. The COA defines valid general ledger accounts, available dynamic segments, and balance maintenance for segments other than the natural account. You create chart of accounts within Chart of Account Maintenance.
- **Main Book**-- Indicates whether the book is the default book for the current company. This book automatically display on all reports and trackers for the current company.
- **Retained Earnings Account**-- A retained earnings option, this value creates an account mask for a retained earnings account so it can display balances from a split COA segment. Kinetic substitutes the segment in the retained earnings account for the corresponding segment in the income statement account.
- **Standard Account**-- Designates a balance card account as the standard account used for retained earnings for the book. Kinetic updates the retained earnings account balance with postings to the COA's revenue and expense accounts.
- **Type**-- Determines the primary function of the book.You can use books to either record financial transactions or consolidate other books.

Logic/Algorithms

The Books functionality uses this logic to calculate its results.

Books contain a fiscal calender, a chart of accounts, and a book currency. GL transaction types use these items to calculate the financial results for each book.

Example(s)

The following example(s) illustrate how you use the Books functionality.

A book uses a COA with a second mandatory segment that defines two divisions, LA and MP. The third segment is optional and defines cost centers. You use segment substitution to split the balance in the standard retained earnings account 3070. At year-end close, the following balances exist in COA revenue and expense accounts.

Account	Amount
4010-MP-100	240

Account	Amount
4010-MP-200	500
4020-MP-100	100
4020-LA-100	300
3010-LA-100	-500
3010-MP-100	-1200

Kinetic updates the retained earnings account when transferring opening balances to the new fiscal year.

Account	Retained Earnings Account	Amount
4010-MP	3070-MP	740
4020-MP	3070-MP	100
3010-MP	3070-MP	-1200
Total to 3070-MP		-360
4020-LA	3070-LA	300
3010-LA	3070-LA	-500
Total to 3070-LA		-200

Chart of Accounts

Kinetic uses a chart of accounts structure (COA) to control the entry of accounts. Each book must have a COA selected on it so that journals can post to the book.

The account structure and segment values used for the book is then defined by its selected chart of accounts. Each book can either have its own COA, or multiple books can also share the same COA through COA mapping.

A chart of accounts consists of controlled and dynamic segments. A controlled segment does not change as transactions are applied against it, so controlled segments record the primary financial history of the company. You use dynamic segments to record temporary, unique business activity.

You create and modify COAs within Chart of Accounts Structure Maintenance. Be sure before you modify or create a COA that you carefully consider the purpose each book and chart of accounts has

for the financial management within the current company. Once a COA is used for a transaction, you can no longer add a controlled segment to it. You can, however, always add dynamic segments to a COA, as these segments do not affect the validity of the account.

Segment Characteristics

Use Chart of Accounts Structure Maintenance to define the segments and their overall characteristics which will be used on the COA. The characteristics you define here indicate how a segment interacts with other segments. COA accounts can contain as many as 20 segments, each of which contains as many as 50 characters. Then each account code can contain up to 240 characters maximum, including their separator characters. Options you define for each segment:

- **Where they get their values**- Kinetic validates controlled segments posted to the COA against values defined as valid in general ledger applications. Values defined for controlled segments are available in fields used for entry of general ledger accounts. Dynamic segments set segment values based on values in posted transactions or on user input. Kinetic uses entity references, posting rules, and user input to set dynamic segment values.
- **Whether they are required in COA accounts**- When you indicate that a controlled segment is required or optional, this value applies to all accounts which use this controlled segment.
- **Account Length**-- You indicate how long each account will be within this application.
- **Alpha or Alphanumeric**-- You indicate whether the accounts can contain letters or both letters and numbers.
- **Balance Options**-- You indicate how each segment maintains balances on this card. These balances then appear on financial reports.

To define more specific characteristics for each segment, use Account Segment Values Maintenance, Self-Balancing Segment Maintenance, or General Ledger Account Maintenance. These applications are also described in this section of the guide. For example, Kinetic always designates the first segment in the COA as the natural account segment. You must define natural account values in Account Segment Values Maintenance. In that application, you define revaluation settings and segment requirements for each account segment.

Dynamic Segments - Setting Values

Kinetic sets values for dynamic segments through the following methods:

- **Accounting Rules**-- Dynamic segments without a business entity use lookup tables and posting rules to set the value of a segment based on values in a posted transaction. Account Segment Values define the segment values, which are selected applicationmatically when a transaction posts.

- **Record Identifiers**-- By using the identifier for a Customer, Supplier, Part, or other system business entity, the dynamic segment can define subsidiary ledgers for accounts receivable and accounts payable control accounts. Account segments can reference the same entities used with GL control types. A GL control based on a type can expose the dynamic segment to the posting process for the entity. Because of this, the posting rules for the process can set the value of the segment.

- **Reference-Type Identifiers** -- Use a reference-type segment to classify campaigns, projects, and other items with a fixed life span. You use GL COA Reference Types to define reference types and the accounts with which they are used. Account Segment Values define the type values. You determine the value of a reference-type segment when a transaction posts.

Business Entities

Business entities define the overall primary items within each company database, like customer, supplier, part, and part class. Kinetic installs with primary business entities; each business entity is essentially a static table which contains the overall database structure used to record transactions against the specific item.

When a transaction is placed against a record contained by a specific business entity (for example, customer), it populates the fields contained within the business entity table. You can configure how the existing entities can be used to match your business needs.

Business entities are at the top of the hierarchy which generates accounts during the posting process. The items directly below business entities in the hierarchy are GL control types. Each GL control type is linked to a specific business entity, and so the business entity defines what areas of the database the GL control type updates through its account and journal contexts. GL control types are templates used by the third and final level of this hierarchy -- GL controls. A GL control uses the account contexts specified through the GL control type to defines the specific accounts which update for each transaction.

To reflect any changes to the database, Kinetic will automatically update the primary business entities through each service pack and release. These business entities are required by Kinetic, so you cannot change them.

The **Use Business Entity** check box defines whether a business entity can be used for a dynamic segment. When this check box is selected, it indicates you can associate the business entity with a dynamic segment within Chart of Accounts Maintenance.

Master Chart of Accounts

The Master COA for a company defines the general ledger accounts available for manual entry of general ledger accounts. You define the Master COA for a company within Company Configuration. Kinetic uses the Master COA to do the following:

- Define available values in fields used for general ledger account entry. For example, Kinetic limits entries in the GL Account field in AP Adjustment to only Master COA accounts. Because

of this, the Master COA must define all accounts needed to post from these fields.

- Define default accounts available for manual journal entry in multi-book mode. This mode applies to a journal group defined in Journal Entry and results in the Master COA being used with all the journals in the group.
- Provide a means of specifying the COA to which other types of transactions post. Optionally, you can limit posting from AR Invoice, AP Voucher, and other applications to Master COA accounts. The general ledger control associated with Kinetic sets the limitation.
- Define available accounts when upgrading to Kinetic. The COA used prior to the upgrade supplies the general ledger accounts used by the Master COA. The upgrade creates a Main Book associated with the Master COA in order to maintain default posting processes.
- Determine which COA displays when general ledger maintenance applications open. These applications open with the Master COA displayed by default.

Typically, the Master COA maintains a complete set of accounts used to post journals from all applications to the Master Book for the company. Other books use amounts posted to the Master COA. You can map book segments and define posting rules to enable the transfer of amounts between books.

Alternatively, you can use the Master COA to capture journals for other books. For example, you define two books with different COAs, neither of which have enough details to derive accounts from each other. The master COA combines the COAs in the two books so all accounts are available to both books.

Balance Settings

Options in the Balance Options area of Chart of Accounts Maintenance affect balance processing for the selected segment. Available options:

- Store balances for segments for use in reports and trackers. The natural account segment always maintains balances, but Kinetic can store balances for other segments. Use the Include in Detail Balances and Include in Summary Balances check boxes to determine how Kinetic maintains segment balances for use on reports and trackers. In addition, consolidation processes create journals for source COA segments that maintain balances.
- Carry balances forward for revenue and expense accounts. By default, Kinetic zeros balances in temporary accounts at year end. When you select the Opening Balance on P/L option, the income statement balances independently from the fiscal year. For example, you apply the option to a segment used to post project revenue. As a result, Kinetic maintains period-to-date revenue data by project for use in multi-year analysis and reporting.

Application Location and Modifiers

The following section details the location(s) where you can access the Chart of Accounts functionality from the Main Menu. It also describes the values you can change for this item.

Use Chart of Accounts Structure Maintenance to create new COAs or modify existing COAs. To launch this application from the Main Menu:

Menu Path: Financial Management > General Ledger > Setup > Chart of Accounts

Modifiers you define in this application:

- **Master**-- Indicates whether this COA is the Master Chart of Accounts for the current company.
- **Chart Length**-- Defines the length of the chart.
- **Separator Character**-- Defines the character used to separate the various segments within the chart.
- **Display Order**-- Use this card to modify the order in which controlled segments and dynamic segments display.
- **Dynamic**-- Select this check box to indicate the segment is dynamic. Kinetic then defines the value of the dynamic segment based on entity references and posting rules. If this check box is clear, the segment is controlled and cannot be automatically changed by Kinetic.
- **Maximum Length**-- Defines the largest number of characters which can be used for the segment. Each segment can contain as many as 50 characters
- **Minimum Length**-- Defines the smallest number of characters which can be used for the segment. Each segment must have a minimum of 1 character.
- **Name**-- Use this field to enter the name you wish to use for the segment. Typically Segment 1 is often named Natural. Examples of other segment names: Department, Product, Customer, and so on.
- **Use Business Entity**-- Select this check box to indicate the current segment uses a business entity to define the segment value. If this check box is clear, it indicates Kinetic uses posting rules and lookup tables to define the segment value.
- **Business Entity** - Defines the business entity referenced by a dynamic segment. This segment type can define subsidiary ledgers for accounts receivable and accounts payable control accounts. All primary business entities are available for selection on this drop-down list when you select the **Use Business Entity** check box.

  When you select a customer, supplier, part or other business entity, you indicate the segment uses the values recorded within the business entity's table and fields to calculate the segment values. The posting rules which reference the business entity set the segment values during the posting process.
- **Segment Value Field** - Defines the database table used for the business entity. You can select tables which contain a single-value primary key.

- **Description Field Name** - Defines the database field for the business entity. All the fields available in the table you selected for the Segment Value Field display on this list.
- **Create Segment Values**-- Select this check box to indicate that dynamic segment values are created automatically. This causes segment values to be automatically added when a system business entity record is created. If this check box is clear, you must manually update the business entity values for this segment.
- **Balance Options**-- A number of options are available in this section:
    - **Include in Detail Balance**-- Indicates that Kinetic maintains segment balances for display on reports and trackers.
    - **Include in Summary Balance**-- Indicates that Kinetic summarizes segment balances for display on reports and trackers.
    - **Opening Balances on P/L**-- Indicates that Kinetic maintains year-end segment balances for expense and revenue accounts. By default, Kinetic zeros balances in temporary accounts at year end. If you select this option, you can maintain an income statement independent of the fiscal year.
- **Alphanumeric**-- Select this check box to indicate the account segment can include both letters and numbers.
- **Entry Control**-- Use this drop-down list to indicate whether the current segment must appear within general ledger account codes. Available options:
    - **Mandatory**-- This segment must be included in general ledger accounts.
    - **Natural Account**-- The segment can be either optional or mandatory for each natural account value. You define this option within Account Segment Values Maintenance.
    - **Optional**-- If you wish, you may post general ledger accounts using this segment.
    - **Reference Account Mask**-- The segment can be either optional or mandatory, depending on the account mask defined in GL COA Reference Type Maintenance.

Logic/Algorithms

The Chart of Accounts functionality uses this logic to calculate its results.

A chart of accounts contains the account structure used to generate accounts for a specific book.

Business entities define various items, like customers and projects, which are required for system GL control types.

Example(s)

The following example(s) illustrate how you use the Chart of Accounts functionality, including business entities.

Example One

Government regulations require reports that use an account structure different than the one used for corporate reporting. In this case, company D can define two books. One book uses a COA for government reporting; the other uses a COA for corporate reporting.

Example Two

COAs used by the subsidiaries within a company can supply values to a management COA that contains a limited number of accounts.

Example Three

You use a required controlled segment to define the departments within the current company. Values for each department are defined within Account Segment Values Maintenance. Then you define the valid combinations of the department segment and other controlled segments within General Ledger Account Maintenance.

Example Four

Customer is a business entity, and so the static table it contains is the CustID table. Its Description Field Name value is Name. When a GL control type is linked to the Customer business entity, the GL control type's account and journal contexts only evaluate data contained in the CustID table. The account and journal contexts defined on the GL control type then filter the accounts and journal codes generated for each of its child GL controls.

Chart of Account Categories

Categories link natural accounts to specific values you define with Account Segment Values Maintenance.

Use this application to determine how balances are maintained for associated accounts and the structure and form of financial statements generated for a specific chart of accounts. You modify and create these categories within COA Category Maintenance.

You set up these categories to determine how balances are maintained for associated accounts. Kinetic zeros balances in income statement accounts and maintains balances for balance card accounts.

You also use this application to define the structure and formatting of financial statements. The GL Report Wizard uses entries in this application to automatically create the company's balance card and income statement. Entries in this application define descriptions that appear in the reports and also indicates whether an entry displays as a debit or credit.

You can create sub-categories which are the children of parent categories, and you can create as many sub-categories as you need. Typically you will do this by running the Copy COA Categories command from the Overflow menu. Use this functionality to copy a set of categories from one COA to create another set of COA categories. The previous categories are then used as a template, and will often save you work. You can modify the categories copied from the template to create a set of categories appropriate to the COA.

This process is one method you can use to define financial statements. You can set up financial statements without the use of categories or the GL Report Wizard.

Through COA Category Maintenance, you also designate the balance card category used to report net income from the income statement. Corporate balance cards typically use Retained Earnings to describe this category. When you select the Net Income check box for a category, it defines it as the retained earnings entry on reports.

Predefined Categories

Predefined categories added to new COAs can help you create financial statements and define natural accounts. When you create a COA, Kinetic automatically creates the following default balance card and income statement categories:

Balance card Categories	Income Statement Categories
Assets	Sales/Revenue
Current Assets	Cost of Sales
Property and Equipment	Gross Profit
Liabilities	Expenses
Current Liabilities	Operating Income
Long Term Liabilities	Other Income/Expenses
Profit, Liabilities	Net Income before Taxes
Equity	Tax Provisions and Net Income

If you need, you can edit and delete all of the predefined categories.

Consolidation Types

Consolidation types and consolidation rate types determine the exchange rates and balance amounts used during consolidations. Typically, consolidations involve posting consolidation journals from a subsidiary book to an intermediate book or from a target book of the source book. These

settings apply when the currency of the source book differs from the currency of the book to which the journals post.

Consolidation rate types determine the exchange rate used during consolidations. You can use the period-end method to apply a specific exchange rate or the daily-average method to apply an average exchange rate for a consolidation period. Exchange rate groups supply values and effective dates used in the calculation of consolidation exchange rates.

Consolidation types determine the method used to calculate account balances and the consolidated rate type applied to them. Calculation methods are used to generate balances or period movements. The balance method uses year-to-date balances to determine period balances for an account. Period movements use period balances to calculate account amounts.

Consolidation rate types apply to consolidation types and to consolidation definitions. Consolidation types apply to account categories which you define in COA Category Maintenance. The consolidation definition supplies default exchange rates applied to consolidation journals posted from the source book.

The following table lists typical applications of consolidation rate types and consolidation types:

Typically applies to...	Consolidation Rate Type	Consolidation Type
COA categories for income statement accounts	Daily Average	Period Movement
The default consolidation rate type for income statement accounts. This value applies to the source book in a consolidation definition.	Daily Average	N/A
COA categories used with monetary balance card accounts	Period End	Balance
COA categories used with non-monetary balance card accounts. This includes fixed assets, inventory, and equity accounts, including retained earnings.	Either Period End or Daily Average	Period Movement
The default consolidation rate type for balance card accounts. This value applies to the source book in a consolidation definition.	Period End	N/A

Application Location and Modifiers

The following section details the location(s) where you can access the Chart of Account Categories functionality from the Main Menu. It also describes the values you can change for this item.

Use Chart of Accounts Structure Maintenance to create new COAs or modify existing COAs. To launch this application from the Main Menu:

- Financial Management/General Ledger/Setup/COA Category

Modifiers you define in this application:

- **Chart of Account**-- Use this drop-down list to select the COA for which you want to add or update a category.
- **Consolidation Type**-- Use this drop-down list to define the exchange rate settings used when consolidation journals generate from the selected COA.
- **Net Income**-- Indicates whether the category reports net income values. Typically corporate balance cards use Retained Earnings to describe this category.
- **Normal Balance**-- Use these options to define whether this category carries either a Debit or a Credit balance.
- **Parent Category**-- Use this drop-down list to define the main category under which this child category displays. The GL Report Writer uses this field to structure statements. You can define as many subcategory levels as you need.
- **Sequence**-- Determines the position of the current category in relation to the other child categories on financial statements.
- **Type**-- Determines whether accounts associated with the category are temporary or permanent. You have two options -- Balance card and Income Statement. At year end, Kinetic zeros balances in income statement accounts and maintains balances in balance card accounts. The GL Report Wizard uses the type to determine the financial statement where the category appears.

Logic/Algorithms

The Chart of Account Categories functionality uses this logic to calculate its results.

COA categories define the natural accounts used for account segment values.

Example(s)

The following example(s) illustrate how you use the Chart of Account Categories functionality.

You use this application and the GL Report Wizard to create a corporate balance card. First, you define Assets, Liabilities, and Equity categories. The Assets category has a normal balance of debit while Liabilities and Equity have credit balances. These settings ensure the correct placement of the categories on the report. These categories have no parent. On the balance card, they group other categories and display a total of their sub-accounts.

You define a category named Accounts Payable, select Liabilities as its parent, and set it as the first entry in the Liabilities category. As a result, the balance of the account associated with the Accounts

Payable category is used to calculate the total for the Liabilities category. Typically, you would define sub-categories for Accounts Payable.

Chart of Accounts Maps

You use COA maps (or account segment maps) to automatically transform one chart of accounts to another chart of accounts during the posting engine process. Leverage this functionality when the differences between books are minor.

For example, if two books share accounting methods but use different currencies, you can map the COA from one source book for use within the other target book. You create these maps within the Chart of Accounts Mapping application and then reference this map within the posting rules for the GL transaction type.

Epicor recommends you use COA maps as much as possible. Maps are easier to maintain than user-defined posting rules. If you have a situation where you can reflect the function of a new book by mapping an existing COA to it, you should always create the map. Mapping transforms a portion of one COA to a portion of another COA and eliminates the need to redefine posting rules that create journals. You can use COA maps to:

- Create consolidation journals. The consolidation process uses the maps to link accounts from a source book to a target book. Consolidation Definition Maintenance specifies the map used in a consolidation.
- Automatically transfer posted journals between books. You associate maps with a book and its rules in GL Transaction Type Maintenance.

Each logical map links two books by defining which COA is the original, or source, and which COA is the receiving, or target. This functionality transforms an account from the COA in one book to the COA in another book. You can then easily transfer accounting transactions between both books, as their COAs have similar structures.

You do this by associating each value of an account segment from the source COA with a single account segment from the target COA. This functionality automatically transfers an accounting transaction line from one book to another by replacing each segment in the accounting string with a corresponding segment from the COA in the target book. Each set of associations between segment values of two particular COA is treated as a single logical map.

During the posting process, the posting engine evaluates posting rules first, followed by any COA maps available within Kinetic.

Application Location and Modifiers

The following section details the location(s) where you can access the Chart of Accounts Maps functionality from the Main Menu. It also describes the values you can change for this item.

Use Chart of Accounts Mapping to create maps that transfer one COA to another COA. To launch this application from the Main Menu:

- Financial Management/General Ledger/Setup/Account Segment Mapping

Modifiers you define in this application:

- **Map Type**-- Defines how the map links the two COAs. Available options:
    - **Account Segment Map**-- Links the source and target COAs through their account segments.
    - **GL Account Map**-- Links the source and target COAs using entire GL accounts.
- **Source Chart of Account**-- Designates the source COA for the map. Transactions posted to accounts in this COA also posts to the target COA.
- **SourceGLAccount**-- Specifies a general ledger account in the source COA. Transactions posted to this account post to the corresponding target account.
- **Source Segment**-- Specifies a segment in the source COA. Values defined for this segment map to values in the target segment.
- **Source Segment Value**-- Specifies a value in the source segment. Transactions posted to this segment post to the corresponding target segment.
- **Target Chart of Account**-- Designates the receiving, or target, COA for the map. Transactions posted to accounts in the source COA also post to the target COA.
- **Target Company**-- Defines the company to which the target COA is used. A source COA can transfer transactions to target COAs in multiple companies.
- **TargetGLAccount**-- Specifies a general ledger account in the target COA. Transactions posted to the corresponding source account post to this account.
- **Target Segment**-- Specifies a segment in the target COA. Values defined for this segment map to values in the source segment.
- **Target Segment Value**-- Specifies a value in the target segment. Transactions booked to the corresponding source segment book to this segment.

Logic/Algorithms

The Chart of Accounts Maps functionality uses this logic to calculate its results.

COA maps are used by GL transaction types to move values from one COA in a source book to another COA in a target book.

Example(s)

The following example(s) illustrate how you use the Chart of Accounts Maps functionality.

The company has a financial and a legal book that use different COAs. You want to post the same accounts payable journals to both COAs. To do this, you create a map with the financial book's COA as the source COA. The map links natural account segment values for AP accounts in the two COAs.

You access the GL transaction type used to post accounts payable journals. This transaction type contains rules used to create the journals for the financial book. You associate the COA map with the financial book. As a result, journals booked to AP accounts in the financial book also post to the COA in the legal book.

Multi-Currency

You use the multi-currency functionality to enter and maintain the currencies through which each company in your database does business.

The multi-currency functionality affects transactions throughout the entire application. As exchange rates between currencies change (as often as daily), your transactions can be automatically revalued using the current, active exchange rates, ensuring that your monetary amounts on all your transactions are up to date.

You can also lock in the exchange rates on specific records. These records then ignore exchange rate fluctuations and maintain the original conversion amounts for all transactions linked to them.

Currency Types

Kinetic organizes currencies through different types. Available types:

- **Transactional**- A currency used on a specific document, or record, in order to conduct business on that record. Transactional currencies are used for records created for external transactions with customers and suppliers, and internal transactions with sites and companies within your organization. A transactional currency is also called a Document currency; you can have as many document currencies as you need for the localities in which you conduct business.
- **Reporting**- A currency your company uses to store, or report, amounts for financial reports. These currencies are used to calculate the final income and expense amounts generated by the business of a specific company. These amounts are officially expressed through a reporting currency. You can have up to three reporting currencies for each company.
- **Base**- The primary currency used by a company. Technically this currency is also considered a reporting currency by Kinetic as well (you may see documentation which refers to each company having four currencies). The base currency is unique, however, as it is the default currency used on all transactions generated within a specific company. When Kinetic cannot find another currency to use for a transaction, it calculates the amounts using the base currency defined for the company.

You can enter as many transactional currencies as you need. Each company, however, can only have four reporting currencies (including the base currency). Business transactions are recorded in all four reporting currencies. Use these currencies in published reports and financial documents. If your organization conducts business in other countries, you run quarterly and annual reports in the base and reporting currencies, displaying financial results with the correct currency values.

Reporting currency conversion is available in all modules except Fixed Assets, Payroll, and Inventory. For these modules, however, a currency conversion may happen if the book currency is different from the base currency for the company. In this case, a conversion does run using the current exchange rate for the currency and its defined rounding tolerances. As a result, a rounding difference may occur in this situation, and a balancing amount is generated within the rounding difference account.

Currency conversions do not happen in the posting engine if the business transaction is recorded in all of the reporting currencies.

Currency Selection

Kinetic locates the specific currency to use on a transaction through a hierarchical structure. It selects the currency to use depending on the position of the record within the hierarchy. Records lower within the hierarchy override records in a higher position. Think of the currency hierarchy as a tree, with a primary default currency at the top and overriding currencies defined on each large branch and then subsequently smaller branch on this tree.

The primary currency used as the default by Kinetic is the base currency. If another currency is selected on a maintenance record, like a book or country, this currency is used instead of the base currency. Likewise, if another currency is selected on a specific record, like a sales order, this currency overrides the currency defined on the maintenance record.

The currency hierarchy:

1. The base currency defined on the company record (Company Configuration).
2. The currency selected on higher level maintenance records overrides the base currency. If a currency is selected on a book or a country record, this currency is used for all records linked to the book or country.
3. If a currency is selected on lower level maintenance records, however, it overrides the currency defined on the higher level maintenance record. Any currencies selected on customer, supplier, or bank account records are used instead to calculate amounts.
4. Lastly, if a currency or rate type is selected on a specific transactional record, like a sales order or an AP invoice, this currency or rate type is used to calculate the amounts on the transactional record.

You can leverage this hierarchy to manage currencies through a method that best fits how your company conducts business. Use the currency hierarchy to select the currencies you need overall, but then modify these default currencies for any unique business situation that may occur.

Manage Exchange Rates

Exchange rates are used by Kinetic to calculate converted amounts from the source currency to the target currency. If you need, you can update exchange rates on a daily basis.

You set up the multi-currency functionality by creating currencies and rate types and then defining where these records are used on customers, books, and other maintenance records. From this point on, you can enter the active exchange rates for these currencies and Kinetic automatically converts your international transactions using these rates. You enter these rates through Exchange Rate Entry.

You can also temporarily revalue currencies through the Currency Revaluation Process application. Use this application to adjust, or revalue, the exchange rate a specific currency pair has within all your open transactions.

All open sales orders, purchase orders, AP invoices, AR invoices, and bank balances that use the selected currency update with the revised exchange rate. Any gains and losses that occur automatically post to the GL accounts assigned to the currency.

The rate changes you enter in the Currency Revaluation Process application do not update the active exchange rates defined within Exchange Rate Entry. This application only temporarily changes the exchange rate.

If you need, you can also temporarily revalue currencies within general ledger accounts through the GL Currency Revaluation process. This process can be run for each book set up within your company; it updates the Chart of Account values defined for the book. Any accounts which contain actual values are evaluated through this process, and any account which contains amounts in a different currency are selected by this process for revaluation.

Override Rates

You have control over the exchange rate used on specific records. You can override an exchange rate on an entry record and lock this rate for all other transactions created against this record. The primary financial entry applications, Sales Order Entry, Purchase Order Entry, AR Invoice Entry, and AP Invoice Entry, all contain fields you can use to override the multi-currency amounts calculated automatically by Kinetic. These fields are located on the header cards found within each entry application.

You do this by first selecting the Lock check box on the record header. The Exchange Rate field becomes active, and you can enter the exchange rate value you need for this specific record. The converted amounts are calculated using the conversion rule defined for the currency pair and the updated exchange rate.

Now this record is ignored by the multi-currency functionality. As long as the Lock check box is selected, its exchange rate never updates. Its exchange rate is frozen and all transactions created with this record use the locked exchange rate. You can then enter any exchange rate that you need on a specific record and maintain this rate throughout all the transactions calculated against the specific record.

Currency Accounts

Any amounts generated through currency conversion or revaluation are tracked by Kinetic. Each reporting currency has its own set of accounts that display these revalued amounts. These accounts are defined as Gain and Loss accounts. The amounts are further divided into Realized and Unrealized categories.

Usually gains and losses are recognized as Realized when the transaction is either fully or partially complete (in this case, the partial amount is considered settled). In some cases, however, a transaction is only considered Realized when a transaction is fully complete.

When gains and losses are considered unrealized, the transaction either is not started or is partially complete (not settled). These transactions typically need to be reversed at some point. When these amounts are reversed, they are changed immediately and both the direct and the reversal revaluation are created at the same time. The appropriate audit records are also created; balances, however, are not affected by the reversed amounts.

When the positive amount is booked as debit, a positive revaluation amount is described as gain, while a negative revaluation amount is described as loss (for example, an AR invoice). In the opposite case, when a positive amount is booked as credit, it is described as loss, while a negative revaluation is described as gain (for example, an AP invoice). Gain amounts are always recorded as credits, while loss amounts are always recorded as debits.

The accounts defined for each currency are:

- Realized Gain
- Realized Loss
- Unrealized Gain
- Unrealized Loss

Rounding

Each currency can additionally be set up to use a series of rounding rules. A rounding rule is available for a specific value type, like Unit Price, Extended Price, Total, Total Tax, and so on. The rounding rule you define for each value type indicates the value direction in which the amount will be rounded. Available rounding rules:

- **Round Up**
- **Round Down**
- **Round Nearest**
- **No Round**

The rounding rule you select for a value type is then used to calculate the final value of a transaction. For example, if you indicate the Unit Price for a currency will be rounded up, a Unit Price value will automatically go to the next highest decimal value. The number of decimals each currency uses is defined on the currency record. If three decimals are defined for the currency, this rounding rule calculates the next highest amount on the third decimal value.

You define rounding rules and decimals within **Currency Master Maintenance**. You enter the decimals allowed for the currency on the **Detail** card, and set up the rounding rules for each value type on the **Rounding** card.

Rounding rules are mainly used with sales documents, such as invoices where specific requirements exist for determining how to round values. The multiplier and rules for the rounding total can also be used in **AP Invoice Entry**.

For more information on leveraging multiple currencies, review the Currency Management module documentation.

Application Location and Modifiers

The following section details the location(s) where you can access the Multi-Currency functionality from the Main Menu. It also describes the values you can change for this item.

You launch the following applications to create and manage multiple currencies within Kinetic:

Currency Master Maintenance

Use Currency Master Maintenance to enter and update the currencies available within a company. You define the currency symbols, number of decimal places, and the rounding rules for each currency. To launch this application from the Main Menu:

- Financial Management/Currency Management/Setup/Currency Master

Modifiers you define in this application:

- **Base Currency**-- Indicates that the currency is the primary currency used by the company. This currency is the default used for all transactions within the company. Only one currency can be defined as the base currency for a company.

- **Currency Symbol**-- Indicates the special character used for this currency. This value appears on reports and applications near the currency amounts.
- **Global Currency**-- Indicates whether this currency can be used by other companies within your organization. Select this check box if you want to make this currency available across your entire organization.
- **Inactive**-- Select this check box to indicate the current company can no longer use this currency. Users cannot create new transactions against this currency, but any previous records that use this currency are maintained by Kinetic.
- **Maintain Rate**-- Select this check box to indicate the exchange rate values for this currency can be updated. Selected by default, this check box activates the exchange rate fields for this currency within Exchange Rate Entry.
- **Number of Decimals**-- Defines how many decimals this currency can display for unit cost amounts. You can have between 0-5 decimals for Cost and Price amounts and 0-3 decimals for General amounts like extended prices, tax amounts, and any amounts posted through inventory, general ledger, assets, and banks.
- **Reporting Currency**-- Select this check box to indicate the currency can be used to display amounts on invoices, purchase orders, sales orders, and other records. You must select at least one currency as a reporting currency for each company. This currency then becomes available on Currency lists throughout Kinetic.
- **Scale Factor**-- Defines the value used to modify the actual exchange rate in order to display amounts for a currency in a more readable format. The actual rates within the database are not changed by this value, but all display rates are entered by users and are shown using this factor value.

Rate Type Maintenance

Use Rate Type Maintenance to apply conversion rules to currency pairs. A currency pair consists of an original, or source currency and a resulting, or target currency. Each currency you add through Currency Maintenance is automatically paired with another currency. This relationship displays in Rate Type Maintenance as undefined; use this application to then define the conversion rule which applies to the currency pair. You can have multiple rate types available to reflect different business needs.

To launch this application from the Main Menu:

- Financial Management/Currency Management/Setup/Rate Type

Modifiers you define in this application:

- **Alternative Cross Rate Currency**-- Use this section to define the interim currency for the Alternate Cross-Rate conversion rule or the Secondary Interim currency for the Double-Cross Rate and Reverse Double-Cross Rate conversion rules. If you can round values for the

Alternative Cross Rate Currency, select the Round check box. You can then indicate up to how many decimals can be used to round within the Decimals field.

- **Conversion Rule**-- Use this drop-down list to define the rule used for a selected currency pair. Available rules:
    - **Direct**-- Multiples an amount in the source currency against the exchange rate for the target currency.
    - **Inverse**-- Divides an amount in the source currency against the exchange rate for the target currency.
    - **Cross-Rate** -- Uses an intermediate, or Interim, currency to convert amounts between the source and target currencies.
    - **Double Cross-Rate** -- Uses two Interim currencies to convert the source amount to the target amount. This conversion rule runs the Direct calculation to convert the amounts generated between each currency.
    - **Alternate Cross-Rate** -- Users an alternate Interim currency to convert amounts between the source and target currencies.
    - **Reverse Double Cross-Rate** -- Uses two Interim currencies to convert the source amount to the target amount. This conversion rule runs the Inverse calculation to convert the amounts generated between each currency.
- **Cross Rate Currency**-- Use this section to define the interim currency used for the Cross-Rate conversion rule or the primary interim currency for the Double-Cross Rate and Reverse Double-Cross Rate conversion rules. If you can round values for the Cross Rate Currency, select the Round check box. You can then indicate up to how many decimals can be used to round within the Decimals field.
- **Decimals**-- Determines how many decimals calculate and display for amounts converted through the current rate type. You can enter between 0-6 decimal places in this field.
- **Fixed Rate**-- Indicates whether the rate defined for this currency pair can update the Effective Rate value. If you select this check box, the rate is locked, or fixed, and the Effective Rate cannot be updated by this conversion rule. If the check box is clear, however, Kinetic considers that the conversion rule uses a variable, or float, exchange rate and it can be updated daily.
- **Global**-- Indicates whether this rate type can be used by other companies within your organization.
- **User Base Rate**-- Select this check box to indicate if the conversion rules from the parent rate type (if one is selected) should be used for the currency pair. If you select this check box, all other fields on the Conversion Rules - Detail card become inactive.

Exchange Rate Maintenance

Use Exchange Rate Maintenance to define exchange rates for currency pairs and an effective date for the rate type to which they belong. Kinetic uses these rates to convert amounts on multi-currency transactions.

To launch this application from the Main Menu:

- Financial Management/Currency Management/General Operations/Exchange Rate Entry

Modifiers you define in this application:

- **Effective Date**-- Defines the date on which the exchange rates begin.
- **Rate Type**-- Indicates the Rate Type against which these exchange rates apply. You create rate types within Rate Type Maintenance.
- **Rate Value**-- Defines the specific exchange rate value you need for each currency pair.

Logic/Algorithms

The Multi-Currency functionality uses this logic to calculate its results.

- **Direct Conversion** = Source Currency Amount x Target Currency Exchange Rate
- **Inverse Conversion** = Source Currency Amount/Target Currency Exchange Rate
- **Cross Rate Conversion** = (Source Currency Amount x Interim Currency Exchange Rate) x Target Currency Exchange Rate
- **Alternate Cross-Rate Conversion** = (Source Currency Amount x Alternate Interim Currency Exchange Rate) x Target Currency Exchange Rate
- **Double Cross-Rate Conversion** = ((Source Currency Amount x Interim Currency Exchange Rate) x Alternate Interim Currency Exchange Rate) x Target Currency Exchange Rate
- **Reverse Double Cross-Rate Conversion** = ((Source Currency Amount/Interim Currency Exchange Rate)/Alternate Interim Currency Exchange Rate)/Target Currency Exchange Rate

Example(s)

The following example(s) illustrate how you use the Multi-Currency functionality.

You are converting a transaction from Singapore Dollars (SGD) to Japanese Yen (JPY). If you do not use a scale factor, the conversion values are the following:

- Source Currency: SGD
- Target Currency: JPY
- Actual Rate: 0.007716 (1 JPY = 0.007716 SGD

- Actual Rate: 129.59 (1 SGD = 129.59 JPY)
- Direct Conversion: JPY Amount = SGD Amount * 0.007716
- Inverse Conversion: SGD Amount = JPY Amount * 129.59

If you use scale factors, however, the conversion values are the following:

- Source Currency: SGD
- Source Scale Factor: 1
- Target Currency: JPY
- Target Scale Factor: 100
- Display Rate: 0.7716 (100JPY = 0.7716 SGD)
- Display Rate: 1.2959 (1SGD = 129.59 JPY)
- Actual Rate: 0.007716 (1JPY = 0.007716)
- Actual Rate: 129.59 (1SGD = 129.59 JPY)
- Direct Conversion: JPY Amount = SGD Amount * 0.007716
- Inverted Conversion: SGD Amount=JPY Amount * 129.59

Fiscal Calendars

Fiscal Calendars define the company calendar used for financial reporting. You can create multiple calendars and designate one calendar as the primary company fiscal calendar.

If you use the Asset Management module, a fiscal calendar must be assigned to this functionality. You must also assign a fiscal calendar to each book. Calendars can start on any date, have any number of periods, and be whatever length in days is required.

Each fiscal calendar contains a sequence of fiscal years. The fiscal years in one Fiscal Calendar record cannot overlap; likewise, no gaps between years can occur.

Because periods close for posting independently in different locations, business transactions may occur in one period in one book and a different period in another book. Each closing period lasts for one day, and if that day is the final date of the fiscal year, the posting engine can post the transaction to this closing period date.

Application Location and Modifiers

The following section details the location(s) where you can access the Fiscal Calendars functionality from the Main Menu. It also describes the values you can change for this item.

Fiscal Calendar Maintenance

Use Fiscal Calendar Maintenance to determine the periods during which journals post.

Launch this application from the Main Menu:

Menu Path: Financial Management > General Ledger > Setup > Fiscal Calendar Modifiers you define in this application:

- **Description**-- Displays the concise explanation for the fiscal calendar. Use this text field to help explain the purpose of the fiscal calendar.
- **End Date**-- Indicates the last date on which the fiscal calendar is active.
- **Start Date**-- Indicates the first date on which the fiscal calendar is active.

Logic/Algorithms

The Fiscal Calendars functionality uses this logic to calculate its results.

EachThe book must have one fiscal calendar selected for it.

Example(s)

The following example(s) illustrate how you use the Fiscal Calendars functionality.

A business may want to have a calendar that represents their fiscal year according to their operations which runs from October 1st through September 30th, while their taxes may need to be calculated based on a calendar year.

General Ledger Accounts

Each COA must contain a valid series of general ledger accounts so that financial information posts correctly. You create and modify these accounts within General Ledger Account Maintenance.

General ledger accounts contain the natural account segment and other controlled segments. You cannot, however, include dynamic segments in general ledger accounts, as the posting rules set the value of most dynamic segments.

You must set up these accounts so that they generate valid entries for a specific COA. Kinetic validates the general ledger transactions that post to the COA against the accounts you define within General Ledger Account Maintenance.

You also define which accounts display on GL Account drop-down lists throughout Kinetic. For example, you can select a GL account within AP Adjustment Entry, but this drop-down list only displays Master COA accounts. Any GL account you define as a master account within General Ledger Account Maintenance displays on this list.

Optionally, each account can be limited for use during a time period you specify. Use the Effective From Date and To Date values to indicate when a specific GL account is active; this causes the COA to reflect changes in the company's account structure or business needs.

For each account, you also must determine if the account description and active status are preserved during automatic account updates. Lastly, you also define whether the account is used for inter-company processing. Inter-company processing updates multi-company accounts defined in the COAs of a parent company and its subsidiaries.

Chart of Accounts Structure Maintenance designates whether a segment is dynamic or controlled. You use the Account Segment Values application to define the values used for the account segments in General Ledger Account Maintenance.

Application Location and Modifiers

The following section details the location(s) where you can access the General Ledger Accounts functionality from the Main Menu. It also describes the values you can change for this item.

General Ledger Account Maintenance

Use General Ledger Account Maintenance to define valid general ledger accounts for a selected COA.

Launch this application from the Main Menu:

Menu Path: Financial Management > General Ledger > Setup > General Ledger Account

Modifiers you define in this application:

- **Active**-- Indicates whether this GL account is currently used by Kinetic.
- **Chart of Accounts**-- Use this drop-down list to select the COA for which you want to create general ledger accounts.
- **Description**-- Defines the text used for the GL account. This value prints on reports and displays in applications which use this account. Enter the text you need in this field. By default, this value includes the name of the natural account and the abbreviated descriptions of the other segments.
- **Effective From Date and Effective To Date**-- This date range defines the time period during which this GL account can be used for posting. The From Date defines the first day on which this GL account can be used. If the To Date field is blank, the current GL account can be used

for posting indefinitely (or end when its Active check box is clear).

- **GL Account**-- Use this field to either create a new GL account or select an existing GL account. To create a new account, click the drop-down list and select the chart, division, and department options as you need. To search for an existing GL account, click the GL Account button to find and select it.
- **Multi-Company** -- Indicates whether this account is used for inter-company processing. During this process, all multi-company accounts update within the book COAs of a parent company and its subsidiaries. If you select this check box, the current GL account is included in this process.
- **Preserve Activation**-- Indicates whether the current GL account should remain active during automatic account updates. These updates occur while accounts generate within Chart of Accounts Structure Maintenance.
- **Preserve Description**-- Indicates whether the Description text cannot be deleted during automatic account updates. These updates occur while accounts are generated within Chart of Accounts Structure Maintenance.

Logic/Algorithms

The General Ledger Accounts functionality uses this logic to calculate its results.

General ledger accounts define the accounts used for posting to the general ledger.

Example(s)

The following example(s) illustrate how you use the General Ledger Accounts functionality.

Effective Dates Example

An account is defined to record financial transactions for a product line. This product line will be discontinued at the end of the year. When you create the account, you set the To Date field to the last day of the year. As a result, an error occurs if a journal that uses the account is posted to the COA after the end of the year.

Segment Properties Example

The following example illustrates how segment properties affect general ledger account codes. You define the segments and enter the settings shown in the following table.

Segment	COA Settings	Valid Values
1	Segment type: Controlled Entry control: Natural Account	1000, 2000, 3000 (Segment values defined in the Account Segment Values application)

Segment	COA Settings	Valid Values
2	Segment type: Controlled Entry control: Optional	10, 20 (Segment values defined in the Account Segment Values application for these Departments)
3	Segment type: Dynamic Entry control: Optional	Dalton, Addison (Segment values created for these Customer IDs)

You designate 1000-10 and 2000-20 as valid general ledger accounts in General Ledger Account Maintenance. The following table shows how Kinetic validates journal account codes when posted to the general ledger.

Account Code	Valid or invalid?
1000-10	Valid combination of defined segment values.
1000-20	Invalid combination of defined segment values.
1000-10-Dalton	Valid combination of defined segment values and customer ID.
1000-Dalton	Valid combination of a defined segment value and customer ID. The second segment is optional.

GL Control Types

GL control types and GL controls work together to define the accounts used for all business transactions entered in Kinetic.

First, GL Control Types work as a template of default values common to all GL controls assigned to each GL control type. Each type contains a list of account contexts, journal contexts, and business entities available for all of its child GL controls. Each GL control defines the account strings and journal codes available to a specific record during the posting process.

Business entities are at the top of the hierarchy which generates the accounts during the posting process. The items directly below business entities in the hierarchy are GL control types. Each GL control type is linked to a specific business entity, and so the business entity defines from which database tables the GL control type pulls information through its account and journal contexts. The items in the third and final level of this hierarchy are GL controls. A GL control uses the account contexts specified through the GL control type to defines the specific accounts and journal codes to update for each business transaction.

GL control types define:

- **Account Contexts**-- GL controls use the account contexts defined for the type to specify the books and accounts to which journals post. Posting rules also use the account information to define journal details.
- **Account Entry Defaults for GL Controls**-- The defaults can limit how an account maps to a book; they can also require using a specific account for all GL controls linked to the GL control type.
- **Journal Contexts**-- Use journal contexts to associate journal codes with GL controls. During the posting process, journal codes can group journals to which a GL control applies.
- **Entities**-- The business entities limit the records accessible by the GL controls. Specific applications can only use a GL control if its type references the business entity managed by the entry application. For example, if a GL control is linked to a type that includes the Customer business entity, this GL control can only be selected within Customer Maintenance.

You can modify GL control types to extend the posting rules functionality. Before you do this, however, make sure the new contexts are appropriate for the business entities to which the GL control type applies. Once you create new GL control types, you need to modify and create posting rules which then reference the GL control type and its child GL controls.

Kinetic installs with a default set of GL control types; these GL control types correspond to the applications that post business transactions. You might, however, need to create a GL control type which supports either a new accounting process or integration with another financial application.

Application Location and Modifiers

The following section details the location(s) where you can access the GL Control Types functionality from the Main Menu. It also describes the values you can change for this item.

GL Control Type Maintenance

Use GL Control Type Maintenance to create and modify GL control types to reflect the account contexts, journal contexts, and business entities you need for all GL controls. To launch this application from the Main Menu:

Menu Path: Financial Management > General Ledger > Setup > GL Control Type

Modifiers you define in this application:

- **All Books**-- Indicates whether the account context can map to multiple books.
- **Book**-- Optionally, use this drop-down list to select a specific book. This account context can then only be mapped within the selected book.
- **Business Entity**-- An Entity modifier, use this drop-down list to define the business entity for which the GL control type uses. All the available business entities display on this list.

- **Context**-- Describes the purpose for the account context. Enter the context you need in this field; for example, bank fee.
- **Journal**-- A Journal Context modifier, use this drop-down list to define the journal to which the GL control type uses. All the journal codes available within your application display on this list; you create these codes within Journal Code Maintenance.
- **Required**-- Indicates whether an account must be defined for this account context. If you select this check box, you must define the account for GL codes which use this GL control type.
- **User Master Chart**-- Indicates whether the current account context requires a master COA. If you select this check box, you can only use this account context for accounts defined as part of the Master COA. When you select this check box, the Master COA displays in the accompanying field.

Logic/Algorithms

The GL Control Types functionality uses this logic to calculate its results.

GL control types determine whether a GL control is a system GL control or a reference GL control.

Example(s)

The following example(s) illustrate how you use the GL Control Types functionality.

The AR Account and AP Account control types reference the Company business entity. You define GL controls which use both types and apply them to Company A within Company Configuration .

Now a transaction which belongs to Company A is posted. The posting rules use the GL control type and GL control's account references to generate the accounts for the company journals.

GL Controls

GL controls define all the account strings and journal codes used during the posting process. Each GL control contains the accounts and journal codes available to a specific record; as part of the posting process, these accounts generate as needed based on the business transaction which occurred.

Every account or journal code within the GL control has a string identifier that defines the business transaction, likes Sales or Payable.

To generate accounts and journal entries, each GL control uses the account contexts, journal contexts, and business entities (for a reusable GL control, instance and reference GL controls do not use business entities) defined on the GL control type to which it is assigned. The GL controls reflect the COA structure required for a specific book or multiple books. When a business transaction

generates, Kinetic reviews the group of accounts specified in the GL control and selects the specific account which applies to the business transaction.

You create GL controls within GL Control Maintenance. All posting accounts used in Kinetic are defined within this application. You can select as many accounts as you need for each GL control. Based on the GL control type, the journals each business transaction creates is automatically included with the GL control. You can use GL controls to define an account hierarchy through the posting rules. For example, you can assign a GL control to a customer and a part class record. You can then set up the posting rules so that if a business transaction is not valid against a customer record, it will then use the GL control linked to the part class record. Instead of the GL controls on the customer, the GL controls defined on the part class are used by Kinetic to determine the accounts.

After you set up the GL controls, you can then select one or more GL controls to a specific record within a maintenance (setup) application. Although you can select multiple GL controls for each record, each GL control you select must be linked to a different GL control type. Within Customer Maintenance, for example, you can select multiple GL controls, each with different GL control types, for a specific customer record. The accounts and journal codes defined for each GL control then calculate when a business transaction is placed against this customer record.

You do not use GL controls on applications where you select posting accounts while entering transactions. Examples of this type of application include AP Adjustment and Cash Receipts . You instead select a GL account defined as part of the Master COA; these accounts display on the GL Account drop-down lists within these applications.

Instance, Reference, and Reusable GL Controls

As briefly described previously, Kinetic uses GL controls in three ways:

- **Reusable GL Controls**-- These GL controls can be modified by users and generate unique results for display on financial reports. They provide multiple contexts for accounts and journal codes, and they can be assigned to various maintenance records (for example, customer, supplier, and part records). Only reusable GL controls use business entities, so the posting results for these GL controls always populate the business entity tables linked to their parent GL control types.

- **Instance GL Controls**-- These GL controls define a specific account used for a particular item. Any transactions which cause accounts to generate master data, operational data, or general settings require instance GL controls. Typically, instance GL controls generate pre-posting data. They are also used when a posting rule provides the specific account for a transaction. An example is the instance GL control which generates account values for a PORel (purchase order release) record.

- **Reference GL Controls**-- These GL controls generate system level transactions required for Kinetic. They create automatically by the posting engine during the posting process; they create a link between the GL transaction and the source business transaction. Hidden from users, reference GL controls populate the TranGLC table with GL transactions. For example,

the source business transaction can be a PartTran (part transaction) record and reference GL control generates when the Capture COS/WIP GL transaction type posts.

Posting Process

During posting, various posting rules, defined in GL Transaction Type Maintenance , use control values to create journal details. The rules generate account values through their account contexts. Often, control accounts lack values for one or more segments. The posting rules use data in posted transactions to define these segment values. Likewise, the journal contexts associated with the GL control type generate the specific journals codes defined on the child GL control during the posting process. As a result, you can track and report on journals by code.

You can create and modify GL controls to extend posting functionality. For example, you add a Landed Cost account context to the AP Account GL control type so that you can then add a landed cost account to the AP GL control. You select this GL control to a company record within Company Configuration . The Landed Cost accounts now automatically generate for this company when landed cost financial transactions calculate.

Be sure the new account contexts are appropriate for the business entities to which a reusable GL control applies. If you modify or delete an account you need within the GL control, it can cause invalid journals to generate. Before you go live with your GL control changes, verify that the posting processes which use the modified reusable GL control create valid accounts after you make account context changes on a GL control type.

Application Location and Modifiers

The following section details the location(s) where you can access the GL Controls functionality from the Main Menu. It also describes the values you can change for this item.

GL Control Maintenance

Use GL Control Maintenance to create and modify GL controls. You then assign these controls to specific records within maintenance applications. To launch this application from the Main Menu:

Menu Path: Financial Management > General Ledger > Setup > GL Control Code

Modifiers you define in this application:

- **Account**-- Specifies an account referenced by the posting processes to which the control applies. Posting process rules use this reference to define journal accounts. You can link as many accounts as you need to the GL control.
- **Type**-- Use this drop-down list to define the GL control type to which you need to assign the current GL control.

Logic/Algorithms

The GL Controls functionality uses this logic to calculate its results.

GL controls determine the account strings and journal codes used to record financial transactions for a specific record.

Example(s)

The following example(s) illustrate how you use the GL Controls functionality.

The AR Account and AP Account control types reference the Company business entity. You define GL controls which use both types. You then select them as GL controls for the Company A record within Company Configuration. A transaction for Company A posts to the general ledger. The posting rules use the account strings selected on the GL controls to record both the AP and AR transactions and generate the appropriate journal codes.

GL Reference Type

GL reference types contain values which classify specific campaigns, projects, and other temporary items within the current company that have a fixed life span. Reference types define the specific temporary item you need in an account segment.

Reference type masks determine the general ledger accounts which include the reference type segment. They also indicate if the specific segment is either required, optional, or blocked from the account generation.

When the posting process runs, the GL reference types automatically generate GL controls which contain the accounts and corresponding journals required to record the transactions. As a result, posting rules are also not needed to generate the values for these accounts.

You use GL COA Reference Type Maintenance to first define the reference types and reference type masks you want to apply to a COA segment. You can then select a specific reference type within the Account Segment Values application, and, in turn, you can select this account segment within Chart of Accounts Structure Maintenance.

You can create a reference-type account segment which contains multiple reference types or create a reference type account segment linked to a single GL reference type which contains multiple masks. Account masks determine the general ledger accounts that include the reference-type segment and whether inclusion of the segment is required or optional.

In some cases, two references types can have masks that match the same account. In this case, Kinetic uses the type with the mask that most closely matches the account for the transaction. For example, a type that includes a mask for the account 2008-22-738 takes priority over a type with a mask of 2008-2_-738.

You can define a different set of reference types and masks for each COA, but if you use a standard set of reference types across multiple COAs, you will have a broader analysis of company data.

When a transaction posts, users determine the value of a reference-type account segment. Because these values are entered manually, you can use classifications which cannot be set by values within posted journals. For example, you define a reference type to classify businesses as corporations or sole proprietorships, because this information is unavailable from customer or supplier records.

Application Location and Modifiers

The following section details the location(s) where you can access the GL Reference Type functionality from the Main Menu. It also describes the values you can change for this item.

GL COA Reference Type Maintenance

Use GL Control Maintenance to create and modify GL COA reference types. To launch this application from the Main Menu:

Financial Management/General Ledger/Setup/GL COA Reference Type

Modifiers you define in this application:

- **Chart of Account**-- Use this drop-down list to select the COA you want to contain with the reference type segment.
- **Account Mask**-- Determines the general ledger accounts that use the current reference type. You can select the specific accounts or use the underscore (_) as a wildcard to reference multiple accounts.
- **Reference Status**-- Use this drop-down list to define whether accounts that match the mask include values defined for the selected type. Available options:
    - **Exclude**-- Blocks using reference type values with accounts that match the mask.
    - **Optional**-- Can use reference type values with accounts that match the mask.
    - **Required**-- Must user reference type values with accounts that match the mask.

Logic/Algorithms

The GL Reference Type functionality uses this logic to calculate its results.

GL reference types are used by account segment values to define short term items like marketing campaigns and projects.

Example(s)

The following example(s) illustrate how you use the GL Reference Type functionality.

You want to create a reference type to classify travel expenses incurred by employees. In Chart of Accounts Maintenance, you define a reference-type segment for the Master COA.

In this application, you select the COA and segment and create a Travel type. You define the mask 2008-___-___. This mask includes the natural segment account you use for employee vouchers, and the division and department segments are masked. You indicate the mask status is required.

In Account Segment Values, you select the segment mask and define segment values for the options Lodging, Air Fare, and Per Diem. Since the mask is required, users must designate their expenses as either Lodging, Air Fare, or Per Diem when they post them.

GL Transaction Type

GL Transaction Types define the posting rules which run during the posting process for a specific business activity. Examples of GL transaction types include AP Invoice, AR Invoice, Bank Reconciliation, PO Release, and so on.

Each GL transaction type has a set of posting codes and amounts specific to the posting process; these values pull in the data needed by the posting rules to build journal details. You modify and create GL transaction types within GL Transaction Type Maintenance. You can also import and export GL transaction types as .xml files; use this feature to use modified GL transaction types in different companies within your organization.

The GL multi-book transaction type (MultiGLJrn) includes the standard default set of posting rules for a single book. The single book is required to be linked to the master chart of accounts (COA) in order for the rules to work correctly. You can then use this default set of posting rules as a starting point to build on the rules for your book or for the COA mapping of other books.

You modify a GL transaction type by creating revisions. You can add, edit, and delete the posting elements (posting codes, amounts) and posting rules (functions, pre-posting rules, posting rules, and reference rules) on each revision. By modifying posting codes and rules, a revision can extend how a specific transaction type posts its results to financial reports and the general ledger. Posting rules can create a single detail or a pair of balancing entries; these rules, except for the reference rule, determine the line amount, the account to which it posts, and whether the amount debits or credits the account. The reference rule is used to set up and populate a reference GL control with no direct relation to posting any amounts or selecting debit and credit accounts. In some cases, rules create a GL control used to store accounts and contexts for subsequent processing. Pre-posting rules define defaults used during manual entry of general ledger accounts.

Each GL transaction type revision can be modified for each book, so a book's chart of accounts, fiscal calendar, and currency all affect the posting process. The COA mapping you select for the book in the GL transaction type determines how the accounts are structured. The GL controls used by each book also affect posting results, as they determine the specific account strings and journal codes used to record each transaction. You can modify the posting rules as you need within each book and these specific rules are processed independently. And if you add a book, you will either need to select a COA map or copy posting rules to it before the new book can post transactions to the general ledger through the GL transaction type.

You can create a revision which summarizes the journals posted by the transaction type to the book. This limits the number of financial transactions saved to the general ledger tables, as posting invoices, vouchers, and other transactions can result in too many detail records being added to the database. Revisions can also define the posting codes and rules used to post journals. The posting codes determine the attributes used with the journals, while the posting rules define the journal details.

You can also create revisions which integrate Kinetic with other financial applications. The posting rules can prepare the financial data for import into these other applications.

Creating, modifying, or deleting posting rules can create invalid journals. Be sure to verify you are creating valid accounts after you make changes. You should always review the financial results of a new or modified transaction type on a test server to avoid posting invalid journals to your live data.

Posting Rules Versioning

The virtual business document (VBD) and posting rules are upgraded only if there is an actual change, improving performance of the conversion application when a new service pack is released.

The incoming data structure of a transaction type, was previously referred to as an abstract business transaction (ABT) and is now identified as a virtual business document (VBD).

Customized revisions also receive updates from the service packs for out-of-the-box rule sets and individual rules/functions. Users can import individual rule sets into existing revisions, making a manual upgrade. This greatly simplifies setting up multi-book configurations.

Business Transactions and GL Transactions

When the posting engine runs, it uses the GL transaction types to generate business transactions (sometimes called business documents). These business transactions hold the actual financial data which will post to the general ledger.

Each business transaction reflects a specific financial activity within your company. GL Transaction Type Maintenance provides a single interface where you can modify the settings for posting each business transaction. The posting rules process a business transaction for each book in the company, and the validation rules defined on each book verify the results of the posting process. When posting completes, the business transactions are recorded as one or more GL transactions within the general ledger.

When you are modify a GL transaction type you should review all of the financial information it generates before actively using the revision. To do this, you select the Manually review all transactions check box on the Revisions > Revisions Detail card; you can then review all the valid and invalid GL transactions in the Review Journal.

GL Transaction Types are only discussed briefly in this section. For details on how to create and modify transaction types, review the GL Transaction Type Environment section later in this guide.

Application Location and Modifiers

The following section details the location(s) where you can access the GL Transaction Type functionality from the Main Menu. It also describes the values you can change for this item.

GL Transaction Type Maintenance

Use GL Transaction Type Maintenance to create and update the posting rules for transaction types. To launch this application from the Main Menu:

Menu Path: Financial Management > General Ledger > Setup > GL Transaction Type

This application is not available in Classic Web Access.

- Financial Management/General Ledger/Setup/GL Transaction Type Maintenance

Modifiers you define in this application:

- **Always a single Active Revision**-- Select this radio option to indicate that only one active revision can be used with this transaction type at a time. You cannot activate other revisions while you are in this mode.
- **Detailed Description**-- Displays the purpose of the transaction type. For example, Posting AP Logged Invoice.
- **Manually review all transactions**-- Indicates that all of the GL transactions generated by the current transaction type will display within the Review Journal. Activate this mode after you have modified or created a transaction type; you can then use the Review Journal to verify you are generating the results you want.
- **Select an Active Revision by Date**-- Select this radio option to indicate you can use different active revisions for the current transaction type. Kinetic uses the Apply Date value on a journal to calculate which active transaction type to use.
- **Transaction Type**-- Defines the specific transaction type you are creating or modifying.

Logic/Algorithms

The GL Transaction Type functionality uses this logic to calculate its results.

GL transaction types contain the posting rules logic for a specific business activity within your database.

Example(s)

The following example(s) illustrate how you use the GL Transaction Type functionality.

You define a rule for a journal generated when an AR invoice posts. The rule pulls in the warehouse ID for the inventory amount which was sold; this value is puled from the sales order linked to AR invoice. The posting rule sets the value of a dynamic segment based on the warehouse ID.

Journal Codes

Journal codes define the codes used to group journals. These codes help you track and report on journals.

The journal codes are applied to journals by GL controls. The journals each GL control generates is defined by the GL control type linked to the GL control; the GL control type defines the journal contexts which each GL control then uses to define a specific journal code for each transaction. In some cases, posting rules automatically use default journal codes when a GL control does not contain the needed code.

Consolidation definitions also apply codes to journals posted from source books to intermediate books and target books. You can then use journal groups to validate and track consolidation journals.

You create journal codes to group:

- **Month-End Journals** - These journal codes record month end activity.
- **Consolidation Journals**- These journal codes group journals used with the Consolidate to Parent and Import Consolidation from Subsidiary applications.
- **New Transactions**- These transactions are created by new GL transaction types which you create within GL Transaction Type Maintenance.

Pre-Defined Codes

When you create a new company within Company Configuration, a series of journal codes are automatically created by default. The following table details these codes:

Code	Description	Process
AJ	Adjustments Journal	Adjustment entry in AP or AR
CD	Cash Disbursements Journal	Check entry in AP
CR	Cash Receipts Journal	Cash receipt entry in AR

Code	Description	Process
FA	Fixed Assets Journal	Asset posting process in fixed assets
GJ	General Journal	Journals entered in Journal Entry
IJ	Inventory Journal	Capture COS/WIP activity in Job Management
IN	Internal Transfer Journal	Multi-site consolidation
PJ	Purchase Journal	Invoice entry in AP
PR	Payroll Journal	Payroll check entry in Payroll
SJ	Sales Journal	Invoice entry in AR

Default Journal Codes

Some posting processes use default codes when there is no journal code is supplied by a GL control. The following table lists applications where default journal codes are defined:

Journal Code	Entry of default settings occur in...
Adjustments Journal	• The Accounts Receivable- General card in Company Configuration • The AR Account Detail card in AR Account Maintenance • The Detail card in Accounts Payable GL Accounts Maintenance
General Journal	The General Ledger card in Company Configuration
Inventory Journal	The Inventory - General card in Company Configuration
Purchase Journal	The Detail card in Accounts Payable GL Accounts Maintenance
Sales Journal	• The Accounts Receivable- General card in Company Configuration • The AR Account Detail card in AR Account Maintenance

Application Location and Modifiers

The following section details the location(s) where you can access the Journal Codes functionality from the Main Menu. It also describes the values you can change for this item.

Journal Code Maintenance

Use GL Transaction Type Maintenance to create and update the posting rules for GL transaction types. To launch this application from the Main Menu:

- Financial Management/General Ledger/Setup/Journal Code Maintenance

Modifiers you define in this application:

- **Description**-- Indicates the purpose for the journal code. This text value displays on various reports and applications.
- **Journal Code**-- Defines the code you use to identify this journal. When you use a journal code with a GL control, it applies the journal code to all the journals created through the GL control.
- **System Journal**-- Select this check box to indicate that this code is applied to default journals generated by Kinetic.

Logic/Algorithms

The Journal Codes functionality uses this logic to calculate its results.

Journal codes determine the journal contexts on GL control types.

Example(s)

The following example(s) illustrate how you use the Journal Codes functionality.

The pre-defined journal code SJ (Sales Journal) applies to journals that post AR invoices. The code SJ is linked to the GL control which posts journals from AR Invoice Entry. Kinetic applies the code to journals posted from the AR Invoice Entry application. Reports and trackers then group sales transactions by the SJ (Sales Journal) code.

Lookup Tables

Lookup tables map database field values to both segment and account values. You can then modify or create posting rules which use lookup tables.

The tables define segment and account values for journal details based on values which generate in a business transaction, which provides the transformation of the source business transaction data into target GL transaction data.

Posting rules use lookup tables to obtain values for any COA segment. Typically, the tables set values which define dynamic segments. You can leverage lookup tables to define multiple target values from a source value. The parameters you define on a posting rule indicate the lookup table columns it uses during the posting process.

Use this functionality to link any source (sometimes called key) value to one or multiple target values. This functionality:

- Creates and maintains groups of fields with the names and descriptions you assign to them.
- Accesses data from posting rules as a separate logical named lookup table.

If you modify or delete a lookup table, invalid journals may generate. After you change a pre-posting rule, verify that functions which use a lookup table still create valid accounts.

Using lookup tables reduce how many changes you need to make to a posting rule for it to define segment and account values. You create look up tables within the Lookup Tables Setup application.

Application Location and Modifiers

The following section details the location(s) where you can access the Lookup Tables functionality from the Main Menu. It also describes the values you can change for this item.

Use Lookup Tables Setup to create and update lookup tables used during the posting process. To launch this application from the Main Menu:

- Financial Management/General Ledger/Setup/Lookup Tables Setup

Modifiers you define in this application:

- **COA Code**-- Designates the COA used in the lookup table map.
- **DB Field**-- Designates the database field that supplies data for the lookup. Entries include fields in the selected database table.
- **DB Table**-- Designates the database table that supplies data for the lookup.
- **Name**-- Defines the lookup table type. Available options:
  - **Segment**-- Create a lookup table which defines the selected segment values.
  - **Use Full Account**-- Creates a lookup table that sets account values.
- **Search Name**-- Defines the table. Posting rules you modify and create in GL Transaction Types Maintenance use this name as a parameter in lookup functions.
- **Source Field Name**-- Defines a field used to supply the values in the lookup table. The name appears in the source column header on the Detail card.
- **Source Value Fields**-- Designates the database table and fields that provide the data which populates a dynamic segment.

- **Target Field Name**-- Defines the target lookup field. The name appears in the target column header on the Detail card.
- **Validate**-- Indicates whether Kinetic validates field details entered in this application. If you select this check box, a warning displays if target value fields on the Detail card contain invalid values.
- **Value Type**-- Describes the type of information contained in the field. This field provides information to users of this application.

Logic/Algorithms

The Lookup Tables functionality uses this logic to calculate its results.

Lookup tables pull in database values required by a GL transaction type.

Example(s)

The following example(s) illustrate how you use the Lookup Tables functionality.

Example One

You define a rule for a GL transaction which generates when a sales order posts. You want to modify the rule that obtains the ID of the warehouse from which the inventory came and sets the value of a dynamic segment based on the ID.

You use the Source Value Fields card in Lookup Tables Setup to define the database table and field that store the ID. You use the Target Value Fields card to select the COA. In the Details card, you define rows that link warehouse IDs with the values defined for the dynamic segment.

In GL Transaction Type Maintenance , you add a posting code to return the ID from the posted transaction. The rule uses the ID and the lookup table to set the segment value.

Example Two

A user assigns a user-defined property color to parts and needs to account for parts differently depending on their color. A lookup table can associate color codes with accounting segments or accounts. A GL transaction type revision (and an underlying posting rule) receives a color code as a posting code and uses it to retrieve a corresponding accounting segment directly from the lookup table. This saves the work of defining the association in the posting rules.

The table links each particular field of the source value to a field in one of the existing database tables. Posting rules populate the field with the values from the linked database field. In a similar way, the table links each particular field of the target value to an accounting segment in a chart of accounts. If such link is created, the rules can use the values of a linked segment or a whole accounting string as a target value.

Review Journal

The Review Journal is a key tool you leverage while you develop new or modify existing posting rules. When you first create a new revision within your test environment, you can indicate that all transactions generated by the revision display within the Review Journal.

To do this, you select the Manually review all transactions check box on a GL transaction type revision; this check box is located within GL Transaction Type Maintenance on the Revision-Detail card. You can then evaluate the GL transactions generated by the revision and correct any errors the revision may cause. After you finish your corrections, you can then re-post them from the Review Journal and review the data results.

During normal operation of Kinetic, the Review Journal captures any posting errors which occur. So after you go live with your new GL transaction type revision, the Review Journal is still an important tool for troubleshooting issues and monitoring results. You can display error transactions by module, date, and user. You can then identify the source of the GL transaction, make a change to the record, and re-post. The Review Journal shows you the results of your change.

You can also directly adjust the account numbers for an entry line. After you complete the adjustments, you can re-validate the journal and directly post it from the Review Journal. Typically you adjust and post new account numbers when invalid journals originate from an external application. You can also both cancel and confirm GL transaction entries. Both actions remove the journal entries from Kinetic. The confirmation process updates the GL transactions within the books to which they post. If the updated transactions are invalid, the Review Journal displays the posting errors.

The Review Journal divides the posting results by book. You can then print edit lists to review the results within each book, so you can generate lists for source business transactions which have generated both internally through Kinetic and externally through third party applications.

When you post a single transaction, it can create different journal details in multiple books. A single journal entry groups all journals posted to all the books by a single GL transaction. The validation settings defined on each book determine the errors and warnings which display in the Review Journal, so the validation process can produce different results when you post to different books. One book may ignore a GL transaction that generates an error or a warning in another book.

If a revision within a specific book is designed to summarize transactions, it affects the adjustments available within the Review Journal. Summary lines post to both natural accounts and accounting processes. The Review Journal displays source details for summarized lines.

Application Location and Modifiers

The following section details the location(s) where you can access the Review Journal functionality from the Main Menu. It also describes the values you can change for this item.

Use the Review Journal to verify the results of a posting process. You also use this application to review and correct posting errors. To launch this application from the Main Menu:

- Financial Management/General Ledger/General Operations/Review Journal

Modifiers you define in this application:

- **GL Account**-- Defines the account to which the amount posted. If you need, you can change this account and re-post the transaction through the Review Journal.

Logic/Algorithms

The Review Journal functionality uses this logic to calculate its results. The Review Journal has two modes:

- Capture all GL transactions for a modified revision for later review and validation.
- Capture GL transactions for which a warning or an error occurred during the posting process.

Example(s)

The following example(s) illustrate how you use the Review Journal functionality.

Posting an AR invoice can post a journal with one set of details to the company's financial book and a journal with completely different lines to the company's legal book. Journal entries displayed in Review Journal group these separate GL transactions. You can then review these GL transactions to make sure they process correctly.

Rounding Engine

Use the Rounding Engine to define location-specific rounding requirements and rounding-related settings for specific business areas. This functionality includes currency tolerances, company tolerances, and rounding accounts.

The Rounding Engine controls rounding for currencies, sales documents, customers, payments, and cash receipts.

All rounding calculations, both the original amount and the rounding difference values, are captured by Kinetic before the posting process runs. Then during the posting process, the functionality updates both the rounded amount and the difference amount per each detail line (for example, on a sales order). The difference amount, for example, may generate by rounding the maximum difference between the debits and the credits. These values then map to a total account which sums the amounts on each detail line -- causing the rounding values to balance.

You activate rounding by defining rounding tolerances for both AR and AP within the Company Configuration application. You can also define tax amount tolerances on customer records within Customer Maintenance. This tolerance criteria defines when Kinetic recognizes when amounts are considered a currency difference and then posts these amounts to the specified account.

Application Location and Modifiers

The following section details the location(s) where you can access the Rounding Engine functionality from the Main Menu. It also describes the values you can change for this item.

Currency Master Maintenance

Use Currency Master Maintenance to define rounding settings for a currency. You define the rounding tolerances for a specific currency on the Rounding card. To launch this application from the Main Menu:

- Financial Management/Currency Management/Setup/Currency Master

Modifiers you define in this application:

- **Multiplier**-- Defines the value used when rounding amounts for the value type. This value must be a positive number. A zero value indicates no rounding calculation occurs. The number of decimals used when entering multipliers is the same number of decimals defined for the currency on the Currency-Detail card.
- **Rule**-- Use this drop-down list to select the rule you need for each value type. Available options:
    - No rounding
    - Round up to multiplier
    - Round down to multiplier
    - **Round to nearest multiplier**-- For example, round up if the remainder value of dividing the number by the multiplier is greater or equal to half the value of the multiplier; otherwise, round down.
- **Value Type**-- Indicates the field values affected by the rounding tolerance. You can define rounding tolerances for Unit Prices, Unit Taxes, Extended Price, Extended Tax, Total, and Total Tax.

Company Maintenance

You can define rounding tolerances in both AR and AP configuration. Company settings determine whether customer-specific rounding applies. Rounding rules apply to the net unit price when the company calculates discounts on unit price in sales applications.

To launch this application from the Main Menu:

- System Management / Company Maintenance / Company

Customer Maintenance

You can also define rounding parameters on specific customer records. Use Customer Maintenance to define a tax rounding rule and indicate whether the rule applies to invoice lines or totals. Customer settings override the rounding defaults defined for the company.

To launch this application from the Main Menu:

- Financial Management/Accounts Receivable/Setup/Customer
- Financial Management/Multi-Site/Setup/Customer
- Production Management/Field Service/Setup/Customer
- Production Management/Material Requirements Planning/Setup/Customer
- Sales Management/Customer Relationship Management/Setup/Customer
- Sales Management/Demand Management/Setup/Customer
- Sales Management/EDI/Setup/Customer
- Sales Management/Order Management/Setup/Customer
- Sales Management/Quote Management/Setup/Customer

Logic/Algorithms

The Rounding Engine functionality uses this logic to calculate its results.

Provides the logic used to round amounts on specific transactions for a currency.

Example(s)

The following example(s) illustrate how you use the Rounding Engine functionality.

You map all GL transactions, line by line, so they move from Book A to Book B. Book B is in a different currency, so this book needs to round the incoming amounts and record the change in a difference account. When the posting engine updates Book A and Book B, the credit and debit amounts will not balance, and Book B records the difference.

Self-Balancing Segments

A self-balancing segment is one which automatically posts a balancing journal when a GL transaction occurs between two values defined for the segment; the sum of these two values must equal a zero balance. This makes sure that the book linked with the COA maintains a balanced set of records within the segment. Self-balancing segments use balancing and offset accounts to create

the balancing journal. You define which existing segments are self-balancing within Self-Balancing Segment Maintenance.

All self-balancing segments must have a balancing account. You can also define an offset account for optional segments in the chart of accounts (COA). When you define an offset account for an optional segment, you can balance the accounts for mandatory segments.

Segments that define divisions, organizations, departments, and projects are often used as self-balancing segments. You must first create the segment within Chart of Accounts Structure Maintenance and then select the segment within this application.

Application Location and Modifiers

The following section details the location(s) where you can access the Self-Balancing Segments functionality from the Main Menu. It also describes the values you can change for this item.

Use the Self-Balancing Segment Maintenance application to define segments used for self-balancing journals. To launch this application from the Main Menu:

- Financial Management/General Ledger/Setup/Self-Balancing Segment

Modifiers you define in this application:

- **Balancing Account**-- Defines the natural account used to process balancing journal amounts. Either enter this account directly or click the Balancing Account button to find and select it.
- **Level**-- Determines the processing order when a COA has multiple self-balancing segments. The process begins with a segment which has a "1" value and continues in numeric order.
- **Offset Natural Account**-- Defines the account which can receive offsetting balances when an optional segment is self-balancing.
- **Self-Balancing** -- Indicates whether you want the current segment to self-balance.

Logic/Algorithms

The Self-Balancing Segments functionality uses this logic to calculate its results.

- When a balancing account is present, record the balancing amounts in this account.
- When an offsetting account is present, place the starting value within the balancing account and the opposite value in the offsetting account.

Example(s)

The following example(s) illustrate how you use the Self-Balancing Segments functionality.

The COA has two controlled segments: a mandatory division segment and an optional project segment. You define both as self-balancing and use the following accounts.

- A balancing account of 8000 for the division segment.
- A balancing account of 7000 and an offset account of 7001 for the project segment.

A journal credits the revenue accounts for a project owned by Division A and shared with Division B. The journal contains the following lines:

Account	Division	Project	Amount
2100	A		-2500
4010	A	10	1200
4010	B	10	500
4010	A	11	500
4010	A		500

The 500 amount applies to Project 10, which is the project shared by the two divisions, and must transfer from Division B to Division A to balance the division segment. The balancing process first uses the project segment accounts to offset balancing account entries.

Account	Division	Project	Amount
7000	A	10	-1200
7000	A	11	-500
7001	A		1700
7000	B	10	-500
7001	B		500

The process can then create the balancing entries for the division segment.

Account	Division	Project	Amount
8000	A		500
8000	B		-500

The GL Transaction Type Environment

This section describes GL transaction types in more detail. It explores the various items like posting codes, functions, pre-posting rules, posting rules, and reference rules which make up a transaction type.

The section concludes with a section on posting rule tips.

GL Transaction Types in Detail

Kinetic contains a unique set of GL transaction types which post financial results to reports, trackers, and the general ledger. GL transaction types control the posting processes used by each book, and a posting process uses one or more GL transaction types to generate its financial results.

A GL transaction type uses an incoming document template to pull in a specific business transaction, and then generates one or multiple GL transactions for it using a set of posting rules to determine how journals and accounts record the business transaction.

Each book also has settings that affect the posting process. These settings include the chart of accounts, fiscal calendar, and currency for the book. The GL controls used by the posting process define the accounts used by the posting rules.

The posting rules can contain default and user modified algorithms to generate GL transactions. You can select an amount to post and determine which accounts record the selected amount. You can then determine whether the account can be a debit or a credit and you can also include any transaction text you may need to identify the results. In the case of reference rules, they are detached from storing account codes for future reference from the posting rules themselves. This type of rule creates a GL control and links it to a posting entity independently from creating a actual GL journal detail. In other words, the reference rule sets up and populates a reference GL control with no direct relation to posting any amounts or selecting debit and credit accounts.

Kinetic installs with a series of GL transaction types which record the business transactions created by financial activity in the database. For example, GL transaction types define the posting process for inventory movements, invoices, and bank reconciliation. You can display and review GL transaction types within GL Transaction Type Maintenance, a application located on the Main Menu within the General Ledger > Setup folder.

In order to update a GL transaction type to match your posting needs, you create a revision for it. Within the revision, you can add, modify, and delete posting elements (posting codes, amounts, BAQs, and so on) and posting rules (functions, pre-posting rules, posting rules, and reference rules) as you need. You typically create revisions for the following purposes:

- Use COA maps from one source book for use in a new target book.
- Copy posting rules from one source book into a new target book.

- Define the revision used to post GL transactions. This revision applies the posting elements and posting rules you define for the revision.
- Summarize the journals posted to a book. Posting invoices, vouchers, and other transactions can cause a large number of detail records to generate in the database. Summary journals reduce the number of financial transactions saved to general ledger tables.
- Define the posting codes and rules used to post journals. Posting codes determine the attributes used with the journals, and posting rules define journal details.
- Change and add posting codes to extend the functionality available when journals post.
- Define the posting rules you use to post journals. Pre-posting rules define defaults used in the manual entry of general ledger accounts. These rules pull in the data needed for the business transaction to post to the general ledger. Posting rules can create a single detail or a pair of balancing entries; these rules determine the line amount, the account to which it posts, and whether the amount debits or credits the account. In some cases, rules create a GL control used to store journal and account contexts for subsequent processing.
- Create revisions which integrate posting processes from Kinetic to posting processes within a third-party financial application.

If you change a posting rule, it can create invalid journals. Before you use a modified rule on your live data, make sure that the posting rule creates valid accounts. Always modify and verify rules on a test server to avoid the risk of posting invalid journals to the live general ledger.

Virtual Business Document Version

The virtual business document (VBD) and posting rules are upgraded only if there is an actual change, improving performance of the conversion application when a new service pack is released.

The incoming data structure of a transaction type, was previously referred to as an abstract business transaction (ABT) and is now identified as a virtual business document (VBD).

The VBD Version field specifies the version of the virtual business document (VBD) and consists of a major and minor sequence (for example 504.1).

The major sequence increments every time a change is made in the server code that generates the document and is used to verify the structure of the document is compatible with the server code.

The minor sequence increments every time a change is made to the document structure and is reset to zero each time the major version increments. It is used to track changes.

The Patch VBD Version specifies the patch version of the virtual business document (VBD) and has the same structure as the main version. The patch version increments each time a change is

released in a patch to a client between service packs and resets to zero each time a service pack is released.

Rules Set Version

In GL Transaction Type Maintenance, use the **Book > Detail** card to determine how transactions post to a book.

When you select a book on this card, you can define its rules. All posting rules, functions, and variables defined for a single book are a rule set which generates a complete journal for the book.

The Ruleset Version field increments each time a change is made in any underlying rule or function. The Package field specifies the identifier of a rules package (for example, Standard or Extended).

Kinetic comes with two packages of posting rules, in addition to several Country Specific Functionality (CSF) packages. The Standard package is available for users upgrading from Vantage 8.03. The Standard package contains all of the account segments used in this previous version, ensuring that previous GL transactions update into Kinetic. The Extended package is available for new users and existing users who wish to modify their posting processes. The Extended rules package is derived from the Standard rules package, but all segments are removed so that they do not contain any division or department specific logic. Because of these changes, this rules package is a better base for modifying the posting process.

You should only import CSF posting rules if you have activated the corresponding CSF license for your company. Refer to the CSF Guides for details on using CSF Posting Rules.

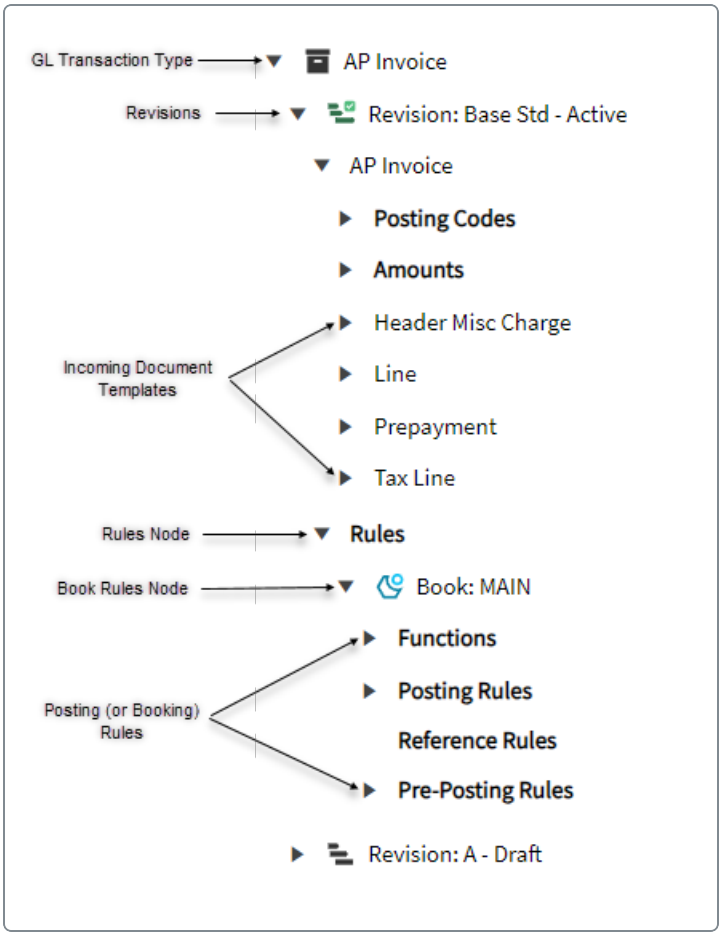
Individual Rule and Function Version

The version control on the rule and function level is used when comparing two rule sets, one of which is about to be imported and the other already in the system. This provides a way to distinguish between posting rules and functions that have changed and therefore need to be pulled in. Those that have not changed do not need to be imported, saving time on the import and conversion.

The Version field specifies the individual rule or function version. The Version field uses a single numeric sequence and increments every time a rule or function is changed in a corresponding rule.

The Patch Version field specifies the rule or function patch version. A single numeric sequence is used and it increments each time a change is released in a patch to a client between service packs. The patch version resets to zero every time a service pack is released.

GL Transaction Type Components



- **Transaction Type**-- Displays the transaction type as registered in Kinetic -- for example, AP Invoice and AR invoice.
- **Revisions**-- Displays the current revisions for the transaction type. Each node also indicates the current status of the revision - active, draft, or blocked.
- **Incoming Document Templates**-- Displays the various business transaction structures available within the GL transaction type. For example, the AP Invoice transaction type contains the Header Misc. Charge, Line, and Tax Line document templates.
- **Posting Codes**-- Displays the specific data values the revision pulls in to process a business transaction. For example, on the AP Invoice GL transaction type, the default revision pulls in the Central Payment Company, Company Parameters, Currency Master, Details, EuroFinance, and Supplier database information.

- **Amounts**-- This node displays the specific amount values pulled in by the revision. For example, on the AP Invoice GL transaction type, the default revision tracks the Balance, Expense, Invoice, Rounding Difference, and Tax amounts.
- **Rules**-- Indicates where all the rules for the GL transaction type revision are contained.
- **Book Rules Node**-- Displays the rules node for a specific book. The functions, posting rules, and pre-posting rules for the book display under this node. You can add multiple books to each revision.
- **Functions**-- Expand this node to display the default functions available within the book. Functions are global rules which generate default data across all the posting rules within the current book.
- **Posting Rules**-- Expand this node to display the rules used to post GL transactions for the current book.
- **Reference Rules** - Expand this node to display the rules used only to post GL transactions with no direct relation to populating any amounts or selecting debit and credit accounts. This rule type does not create any GL details.
- **Pre-Posting Rules** -- Expand this node to display all of the pre-posting rules used to gather data in order for the posting rules to run correctly within the current book. Pre-posting rules use lookup tables to pull in required data.

Revisions

To modify transaction types, you create revisions. Only one revision can be active at a time within the book. You can either create a new revision by copying the existing posting rules, or you can create an entirely new revision which contains no posting rules and create the posting rules you need.

You can add or update many items within each revision; you can modify posting codes, pre-posting rules, posting rules, reference rules, functions, book variables, rules variables, operations, and so on.

You can assign one of three **status levels** to each revision. Each status defines the current state of the revision. Assign the status on the **Revisions > Revision Detail** card. The **Status** field has the following options:

- **Active** - The revision Kinetic currently uses to post GL transactions for the GL transaction type. The revision is in use, processing GL transactions for all books assigned to it. Only one revision can be active at a time for each GL transaction type and you cannot modify an active revision.
- **Draft** - A revision you are currently modifying; this status is the default when you create a new revision. For as long as the revision stays in the draft status, you can edit it. Once you change its status to Active or Blocked, it can no longer be updated.

- **Blocked** - A previous or alternate revision which is no longer in use. Kinetic does not use this revision to generate GL transactions for the current GL transaction type. Although you cannot modify a blocked revision, you can change its status back to active and use it to post again. Blocked revisions are also used as a reference to compare against the active and draft revision(s).

Use different revisions to model and test changes in the accounting logic for each transaction type. For example, your company uses posting rules to set values for a dynamic segment that defines a product line. The product line expires at the end of the year. You create a revision that includes new posting rules which record this change at the same time.

You can delete blocked revisions by archiving them into a specified folder from the **Actions > Delete Revisions** menu.

Option to Manually Review Transactions

In addition to creating a new revision and assigning it a status, you can activate the manual review option when you select the **Manually review all transactions** check box. This causes Kinetic to capture all GL transactions generated by the process and log them in the Review Journal. Epicor recommends you select this option while you are creating and testing a revision; after you are satisfied with the results, clear this check box option so the Review Journal only captures errors and warnings.

Upgrades and Revisions

Transaction Type Maintenance has a separate and independent version control for different parts of the transaction type. These include:

- VBD level
- Version and Package ID on the rule set level
- Version on the posting rule/function level

This extended versioning allows limiting imports and updates to the parts of transaction types where actual changes are present during manual imports and patch upgrades. You can choose to import new revisions in replace mode or apply updates to already existing revisions.

When you upgrade Kinetic through a service pack or patch, you can configure the upgrade in **Update** or **Replace** mode. Refer to the installation guide for detailed instructions.

In the **Replace mode**, the current active Revision remains **Active** if there were no changes in the Business Document Structure. The revision becomes **Blocked** and a new Active Revision is generated if changes in VBD were successfully merged. Custom rules of the current revisions are not upgraded. The new out-of-the-box Revision is installed as blocked. You can then review the blocked version to see if any updates are needed to your modified revision. If you wish, you can then add your modifications to the new default revision and set its status to Active.

In the **Update mode**, the current **Active** Revision is updated to reflect changes in the Incoming Document Template. Only out-of-the-box Posting Rules and Functions which have changes are updated. Customized posting rules/functions are not updated.

Updates are always automatically installed for both the Standard and Expanded packages of rules.

Compare Revisions

Before you make changes to a revision, you can compare revisions to see the different posting elements and rules between the two versions. To do this, click on the Overflow menu and select Compare Revisions. Use this window to review the differences between the revisions.

Incoming Document Templates

An incoming document template specifies the elements used to gather business transaction data for the posting rules. Each GL transaction type uses one or more templates, and each template has a set of elements unique to the posting process for that GL transaction type.

All templates gather both general and specific data about posted business transactions. General information includes the date the business transaction posts, the business entity involved in the transaction, and the journal code for the transaction. Each template also contains a set of elements specific to a posting process. For example, the template used to post invoices defines elements that contain data from the invoice header, invoice lines, and tax lines.

Templates can combine predefined and custom elements. Predefined elements ensure support for the transaction type's posting processes. You cannot change or delete predefined elements. You can add custom elements to provide attributes needed by any new or modified posting rules. You can also modify and delete your custom elements.

The posting rules for a revision use information from the template elements to define journal details. These rules, except for the reference rule, determine the detail amount, the account to which the detail posts, and whether the amount debits or credits the account. The reference rule is used to set up and populate a reference GL control with no direct relation to posting any amounts or selecting debit and credit accounts.

Each incoming document template contains the following elements:

- **Document Lines** - Document lines group elements obtained from a portion of the posted transaction. For example, an accounts receivable invoice template contains document lines for the invoice header, invoice line, tax line, and miscellaneous charge line.
- **Posting Entities** - Entities group related posting codes together. An entity holds a posting code collection and provides default settings used to populate fields.
- **Posting Codes** - Posting codes define the parameters used with posting rule operations and functions. Posting rules use posting code values to create journal details. You can add posting codes and modify and delete the codes you create. Posting codes are grouped under a parent

posting entity.

- **Amounts** - Amount elements hold the transaction amounts used by calculated fields.

Selection Criteria

In GL Transaction Type Maintenance, selection criteria are statements which return specific data from the posting codes on the revision. The posting rules then use the data to create accounts used in journal details.

The incoming document template contains posting codes unique to its process. The posting codes are grouped by posting entities and the entities are then grouped by the document lines from the business transaction (incoming document template).

You can create or update selection criteria to return the posting code data required by a rule. For example, you want to define a rule which posts an invoice amount to an account segment based on the Warehouse ID. The incoming document template contains an invoice line element which has the warehouse posting entity which contains the posting code required for each warehouse. You select Invoice Line in the For Each field and use the other field in the card to create a statement that selects the correct posting code. The result resembles a WHERE clause in a SQL statement. You then click Add to include the statement in the Selection Criteria field. You use the And, Or, and Not buttons to insert logical operators into the statement.

Selection criteria can only pull in data from parent tables. Information stored in child tables cannot be pulled in by using selection criteria from the parent. You must create or modify a posting rule which pulls in data directly from the child table.

Posting Codes

To add posting codes, you define settings that control how the revision retrieves data used for its posting rules. These rules includes defining the data source for the posting codes. Kinetic obtains data for posting code fields by using:

- **Entity Fields**-- Business entities (do not confuse these items with posting entities) are static database tables like Part, Customer, and Supplier which contain data a business transaction requires. Because of this, you can specify a field in the same table to access data in the business entity field.

- **Business Activity Queries (BAQs)**-- You can pull in more detail into the revision by linking a business activity query (BAQ) to an incoming document template. The revision then uses selection criteria defined for a posting rule to obtain data pulled in by the BAQ. The interface can return a value from a single field or a calculated sum of fields.

Functions

Functions are standard packages of posting rules which contain functionality common across the posting rules within a revision. They include the rules required to populate account segments for

natural accounts, divisions and departments.

When you install or update Kinetic, functions are included as part of each GL transaction type revision. Typically each function contains a series of account and journal contexts. When the posting engine processes the function, it pulls in a complex piece of data and makes this data available to the pre-posting, posting, and reference rules for the revision. For example, when the AR Invoice transaction type is run, its functions pull AR account information out of the database to make this information available to the AR invoice posting rules.

Each function is an algorithm that gathers the specific data and the function code which processes the information into a format usable by the pre-posting, posting, and reference rules. The other rules and functions share the gathered data. You can create, modify, and delete functions as necessary. When you add or modify a function, you extend the operations list available to the posting rules.

Functions are very useful when you create or update posting rules. You can use functions to avoid writing complex rules to pull in common financial data. They also help you avoid constantly repeating the GL hierarchy while you develop posting rules.

Posting Booking Rules

Use the Book Detail card to determine how transactions post to a book. You first select a book from either the tree view or the Booking Rules - List card; you can then create, edit, or delete posting rules for the current book.

Use the Book Detail card to determine how transactions post to a book. You first select a book from either the tree view or the Booking Rules - List card; you can then create, edit, or delete posting rules for the current book. All the rules defined for the book create a rule set. When the posting engine runs, it generates a complete journal for all books contained within the revision -- using the default or modified rule set defined for each book.

Most posting rules create journal details from business transactions. These rules can create one transaction line, a single journal detail, or a pair of balancing entries. Rules determine the detail amount, the account to which it posts, and whether the amount debits or credits the account.

As described previously, rules use functions to generate global data across all of its pre-posting and posting rules. These algorithms provide the data required to populate pre-posting and posting rules with the data required to generate GL transactions.

Another type of rule, called a pre-posting rule, creates an instance GL control and calculates the default account which displays and the GL transaction. For example, a pre-posting rule creates the default expense account for the miscellaneous line on a purchase invoice. Pre-posting rules do not process data from posted transactions, instead they process the incoming data from the business transaction and places it in the appropriate accounts required for the posting process.

To generate a GL transaction, a posting rule populates detail fields with appropriate values. The posting rule builds the detail field-by-field. Posting rules can also create output GL controls. These optional GL controls store an association between source data, account contexts, and specific

accounts. By storing this data through the output GL control, Kinetic can then use it during later processing.

You can do the following actions with posting rules:

- Determine whether journals are summarized by account when posted. When you summarize the posted details, it reduces the number of records stored in the database.
- Designate whether mapped segments populate book accounts. The Chart of Accounts Mapping application defines the maps between COA segments. A map transfers journal details in another book to this book. Segment maps reduce the need to define the same posting rule in multiple maps. You typically use this functionality when you only need to make minor changes in the posting process in different books.
- Define the functions referenced by the posting rules. Reusable functions reduce the time needed to application the rules. For example, the book might use a COA with a mandatory division segment. You define a reusable function that pulls a posting code value by using a lookup table; the function then returns the segment value.
- Define variables that provide values to the posting rules. The rules use variables to populate fields in a GL transaction.

Creating, modifying, or deleting posting rules can create invalid journals. Be sure to verify that valid accounts generate after you make your changes. You should always first generate the results on a test server to avoid posting invalid journals to live data.

Operations

Operations define the processing sequence used by a posting rule to move the data gathered by pre-posting rules and functions into financial data which displays on financial reports and the general ledger. The posting engine uses the values generated by operations to create GL transaction detail lines.

Operations define the processing sequence used by a posting rule to move the data gathered by pre-posting rules and functions into financial data which displays on financial reports and the general ledger. The posting engine uses the values generated by operations to create GL transaction detail lines.

You create and update operations within the Booking Rules - Operations cards. This interface displays all of the operations which make up the current rule and provides you with a way to edit an existing operation or add a new one. There are two cards under the Operations card that feature the identical interface for configuring posting rules. This logic allows you to divide your operations-related activities. You can use either of these cards to create and update operations considering the following:

- Use the Operations - Base card to configure the base part of the posting rule. During Kinetic updates, only the base part of the posting rule which was not modified is replaced with the one that comes in an update regardless of whatever changes you made in the customized part. The possibility to configure rules on this card is kept mostly for the purposes of backward compatibility. Usually posting rules are customized on the Customization card.

- Use the Operations - Customization card to configure the customized part of the posting rule. It is recommended to configure the customization, not base, part of the rule as this is the only way to keep receiving changes from updates. The customization part of the rules is executed right after the base part.

The order in which the operation nodes display represents the order through which these operations run. Each operation sets a value in a container for the rule. Containers define the line amount, the account to which it posts, and whether the amount debits or credits the account.

Typically the operation logic runs sequentially, although in some cases they can use conditional branching. Each operation in the posting rule returns a single value or tests a condition. You can add, update, or delete various items on operations:

- Variables used by the rules. You might use a variable to store an intermediate value used in calculation of a field in a rule.

- Functions referenced by the rules. For example, the book might use a COA with a mandatory division segment. A function pulls in a posting code value used to look up and return the segment value.

- Containers that make fields available to the rules. Use containers to access the GL transaction's journal code, apply date, and closing period. Containers for a detail include credit account, debit account, book amount, transaction amount, text, and other transaction-specific containers.

- Selection Criteria that select the posting code to which the rules apply. Use the Selection Criteria card to define the process used to select a posting code from one of the doc lines in the incoming document template.

- Operations that include functions used to set container values. Operations include lookup functions that set segment values based on attributes in a posted transaction.

Each specific operation can be either a formula, a procedure, or a conditional operator (If/Then/Else). A formula populates a field of a GL journal detail (Header and Posting Rule) or an account only (Pre-Posting Rule). The syntax typically used with each operation is the following:

- <Field/Variable> = Formula <Operands>

- Call<system function>

- Comment< Text>

- Conditions
- If <logical expression>
- Else <logical expression>
- Else If<logical expression>

To define accounts, the operations set segment values based on values within an incoming business transaction. For example, a posting rule could set the value of a dynamic segment which classifies customers by country. The operation uses the customer's country, data included in the incoming business transaction, to pull in the country value needed for the segment.

Operations use the following functions to pull in segment values:

- **Lookup Tables**-- These tables define relationships between database fields and account segments. These relationships define the key-value pairs used by the posting rules.
- **BAQs**-- A business activity query (BAQ) obtains data related to the posted GL transaction to use in defining account codes. To do this, you must define the BAQ within the incoming document template.

Modifying and creating operations is powerful functionality. Although you can create the rules you need to post a GL transaction type, using this functionality can often be complex and unnecessary. You should always try COA mapping or lookup tables first to see if you can receive the financial results you want. The COA mapping defines the maps between segments; the map transfers journal details from one book to another book. Lookup tables locate a segment value based on a value in a posted transaction. If using this functionality does not provide you with the results you need, you should next modify or create posting rules.

Color Key

The editing environment on the Operations card of GL Transaction Type Maintenance uses colors to designate various elements. Colors highlight the various logical elements in the text of an operation. The color legend is the following.

Element	Color	Example
Comment	Green	Comment: Define Transaction and Book amounts
Standard Container	Bold Violet	Debit Account.Chart
User-def. Container	Violet	TestAccount.Chart

Element	Color	Example
Named Constant	Maroon	Today; Journal Code
Literals & expressions	Dark Red	"Expense", Today + 3
Posting Element reference	Bold Blue	Amount Record, Supplier, AccountingCode
Function Keywords, Control Flow Operators	Bold Black	IF Debit Account Is Valid Then
Function Parameter (not initialized)	Blue Underlined	Convert Amount To Currency Using Rate Type
Function Parameter (initialized)	Same color as the actual argument	Convert Debit Amount To "USD" Using Amount Record Invoice Rate Type

Variables

Variables are temporary fields intended to store the intermediate data results from operations. Several pre-defined variables are available for each GL transaction type revision, and they can be defined at different locations within the revision structure.

Variables are temporary fields intended to store the intermediate data results from operations. Several pre-defined variables are available for each GL transaction type revision, and they can be defined at different locations within the revision structure. To extend posting functionality, you define variables. Available variable types:

- **Book Variable**-- A variable usable by all the rules within the current book. These variables are set by the book when it is initialized and they cannot be modified within a posting rule.
- **Function Variable**-- A variable only available within the function which contains it. You cannot use a function variable outside of its parent function.
- **Rule Variable**-- A variable only available within the posting rule which contains it. You cannot use a rule variable outside of its parent rule. Rule variables are available for you to use on all operations in the rule which contains them.

Custom variables can use any of the data types available during the posting process. For example, you might use a custom variable to store an intermediate value used for calculating a field in a custom posting rule.

The incoming document template for the GL transaction type revision can hold both predefined and custom variables. The posting process controlled by the GL transaction type uses the pre-defined variables. You can neither delete nor modify these variables. You can, however, add, modify, and delete custom variables on any revision you create.

Viewing Posting Engine Log Information

The **Posting Engine (PE) Log Viewer** tracks the results of your posting rules. You can evaluate how effective the posting rules are processing transactions.

When a posting engine process runs, the data which the posting log generates writes to this log. Each time you run the posting engine process, it adds processing information to the log results. This feature helps you review the history of the posting process. Be aware, however, that this log contains a lot of business call detail. Only use this log if you understand how to trace the posting results.

You can access the **Posting Engine (PE) Log Viewer** from the main menu, GL Transaction Type Maintenance and Review Journal apps.

To access the PE Log Viewer from the GL Transaction Type Maintenance or the Review Journal app, select the **PE Log Viewer** option from the context menu in the **Display Name** field or the **Journal Entry** field correspondingly.

Use the landing page of Kinetic to view existing PE logs.

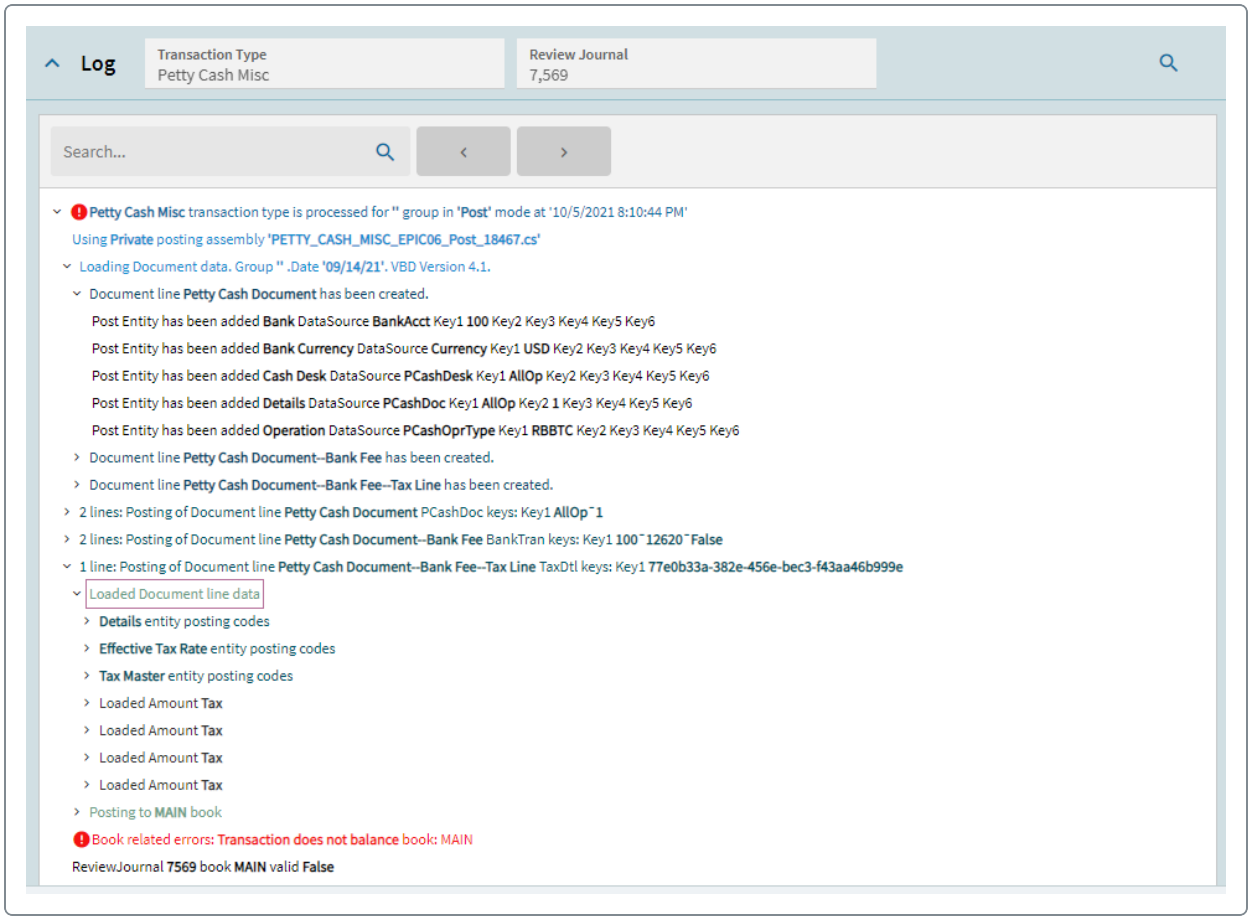
Viewing Log Information

1. From the main menu, go to **Financial Management > General Ledger > General Operations > PE Log Viewer**.
2. Select the record for which you want to review log information on a transaction type or a journal entry.
3. Review the posting log information and any warning or error messages that display.

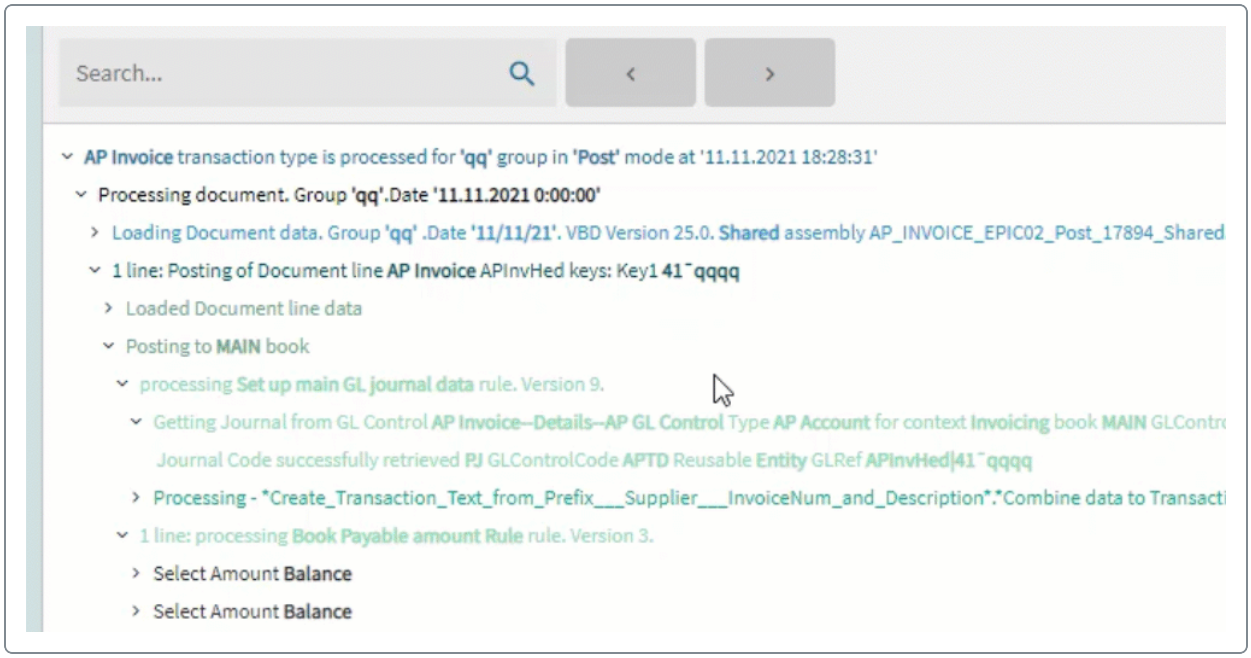
   Each posting log has 2 main levels:

   Incoming data - The document data and the related information that is used in booking rules.

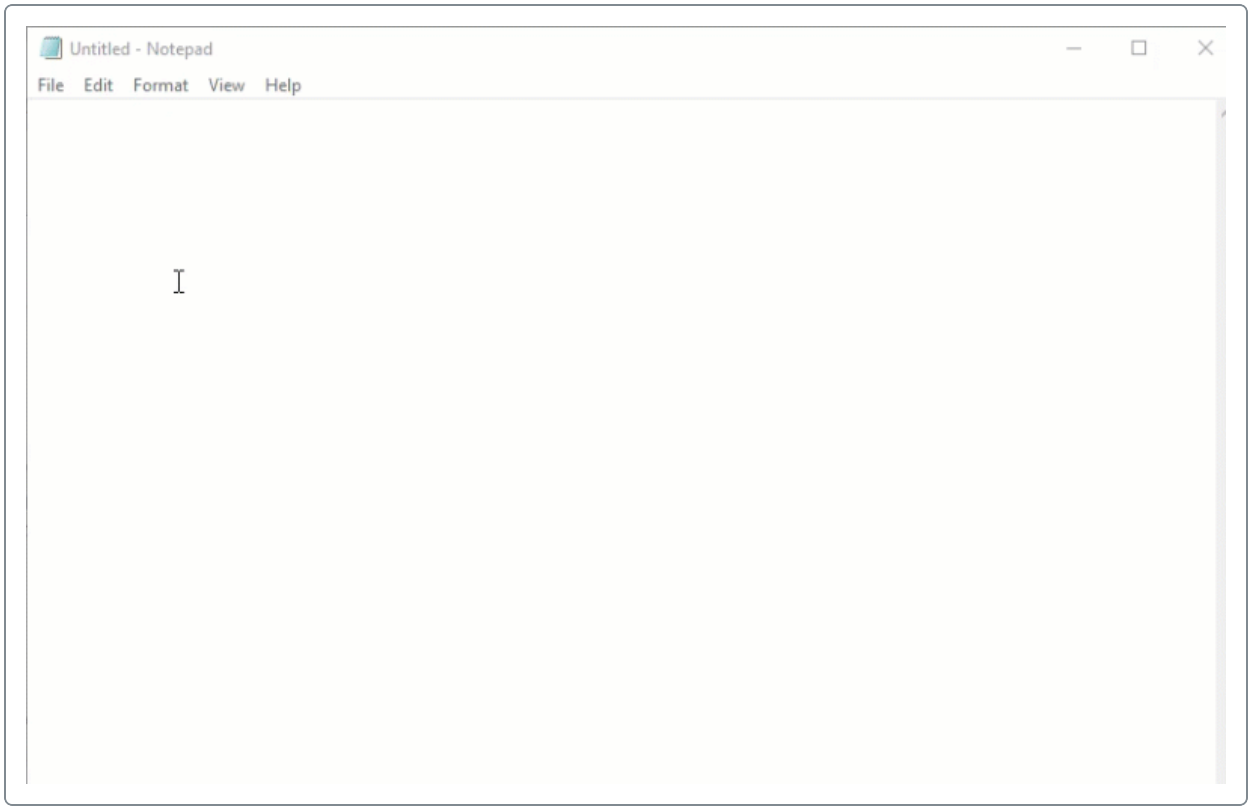
   Produced data - The data the posting engine created from the booking rules.



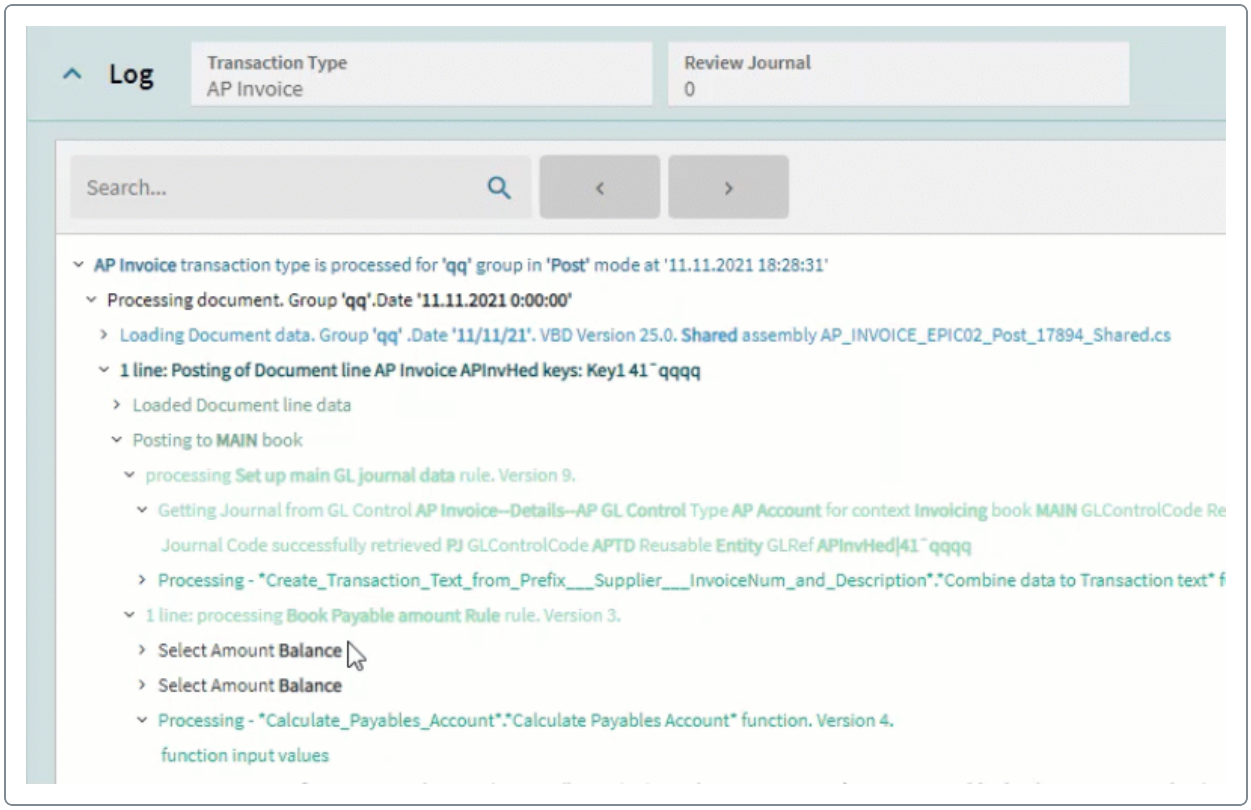
4. Copy the text of the current line or the current line and its sub items. To do this, right-click the line and select a copy option from the context menu.



5. Now paste the text into a text editor.

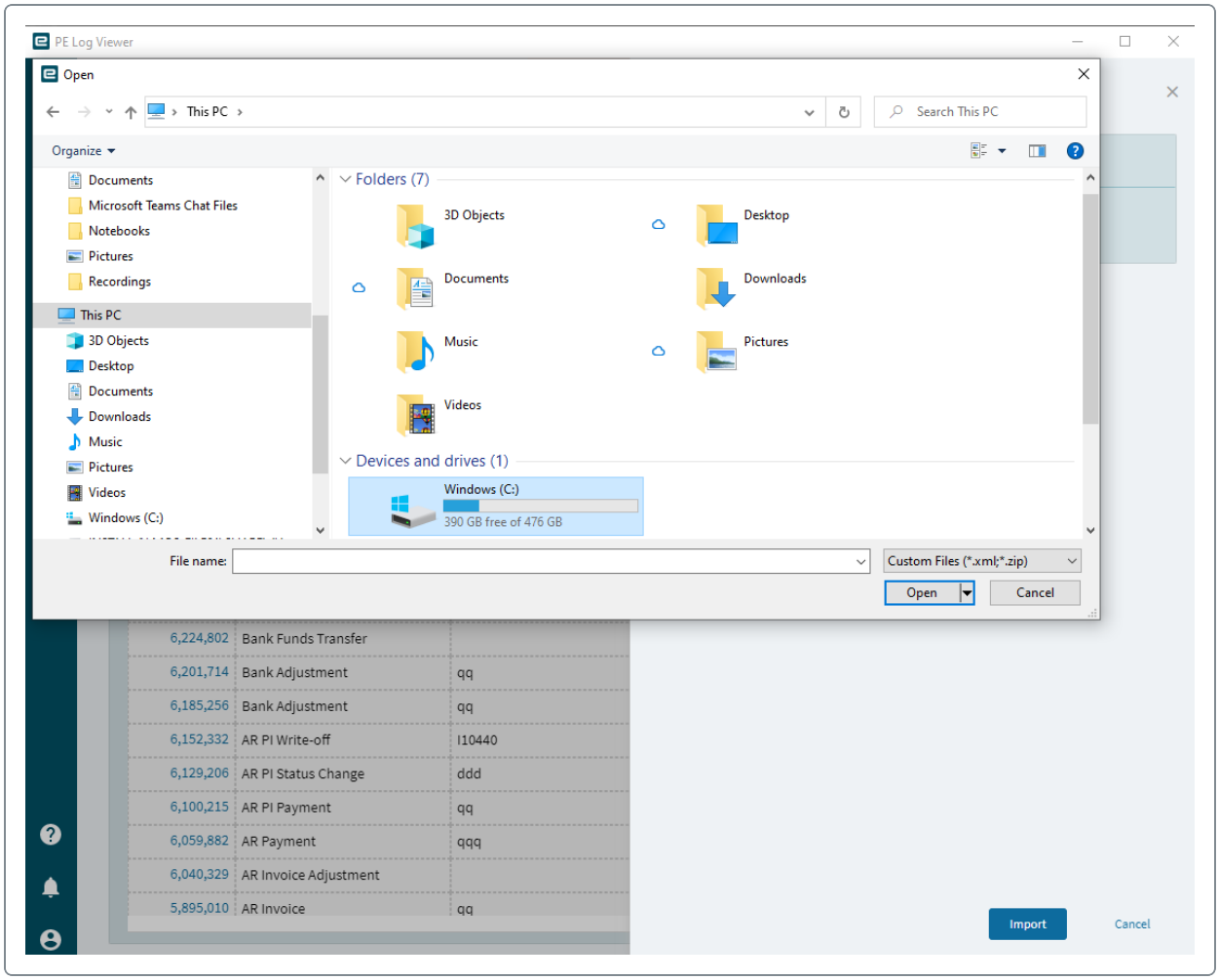


6. You can also search for a value and then move through the log to find this value. Enter the value in the search field, then select the **Next** and **Back** buttons.



Importing a Log File

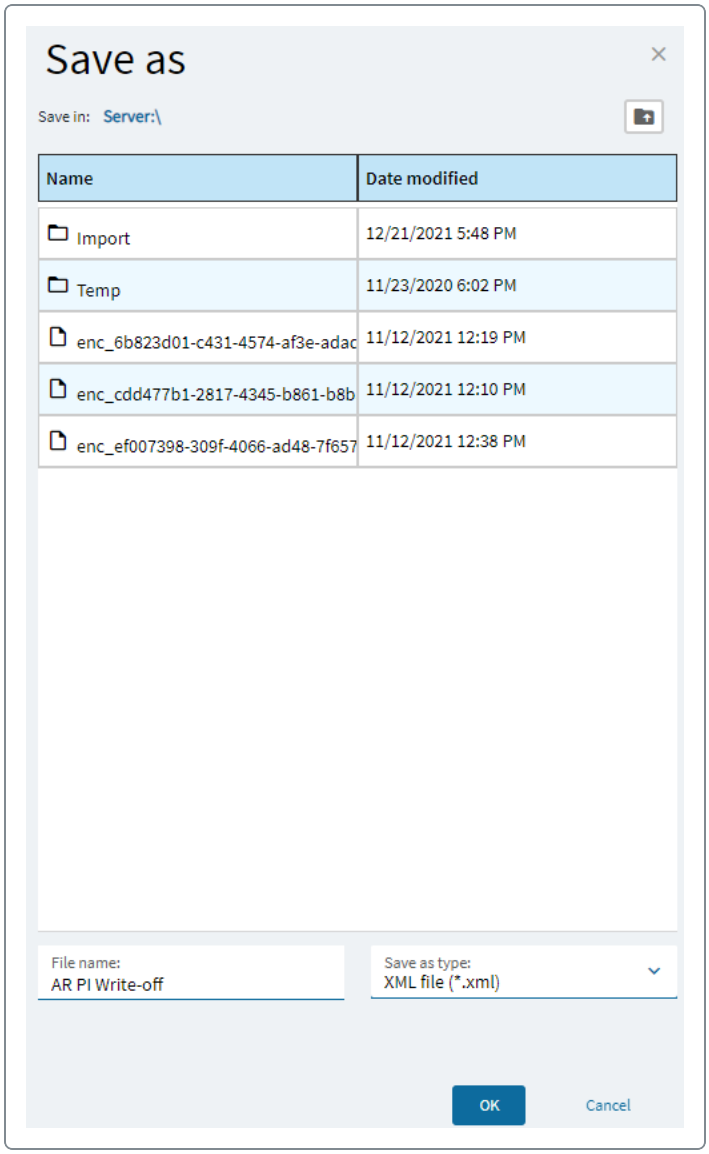
1. From the Overflow menu ••• , select **Import Log**.
2. In the **Import Log Configuration** group box, select to specify the file directory.



3. Select **Import**.

Exporting a Log File

1. Select the posting log you want to export.
2. From the Overflow menu , select **Export Log**.
3. In the **Export Log Configuration** group box, select to specify the export file name and location.
4. In the **Save as** panel that displays, select the folder where you want to located the file.
5. Enter the **File name** and select its **Save as type**.



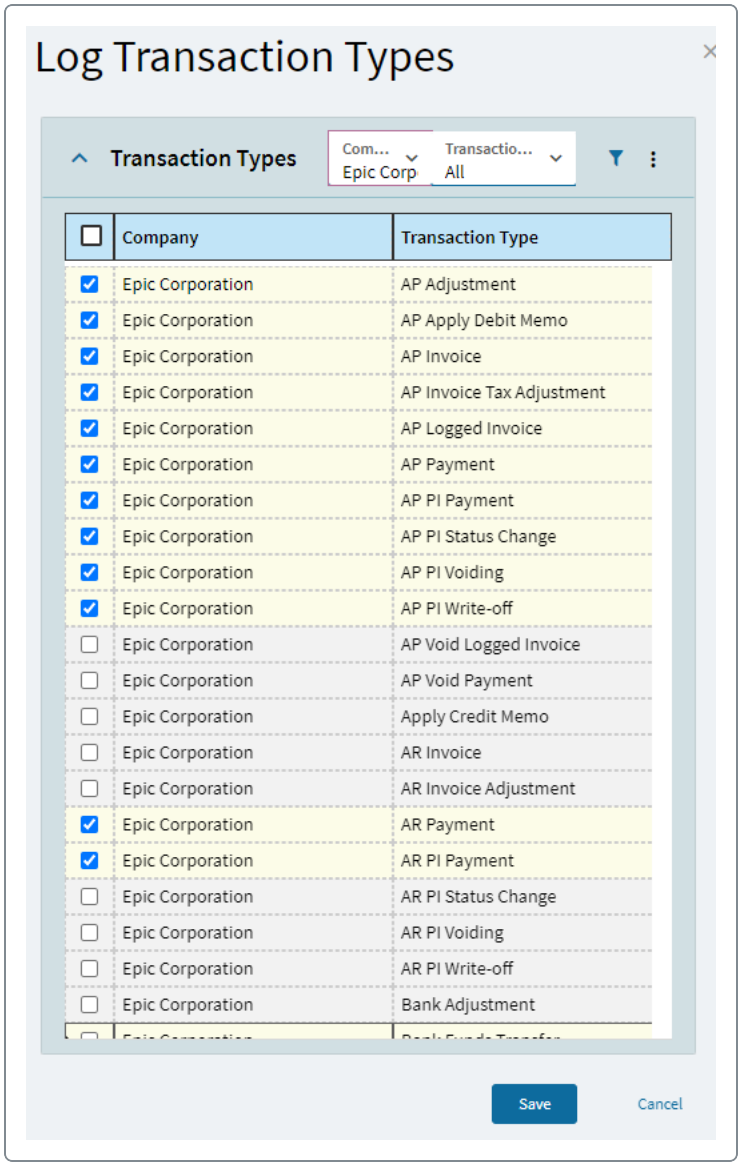
6. Select **OK**.

   The XML file is created in the specified directory on your system.

Turning Logging On and Off

1. From the Overflow menu ••• , select **Settings**.
2. The **Log Transaction Types** panel displays.

3. Select required lines from the list in the grid if you want these transaction types to be logged. Note that you can filter all transaction types by **Company** or **Transaction type**.



4. Select **Save**.

The Posting engine works faster if logging is disabled.

Posting Rule Reference

This section of the guide contains information on the specific items you can modify within each posting rule operation. Use these sections as a reference guide to help you understand the items you can modify on posting rules so that they configure to your business needs.

Posting Rule Data Types

The data type defines which arguments pull in basic data for the posting rule container (items which holds data generated for the operation) and which arguments calculate more complex data for each container.

Basic Data Types

The following table lists and describes basic data types.

Data Type	Description
String	This type defines character data. Any value assigned to the string container is trimmed to remove leading and trailing spaces. You can use posting rules to populate predefined string fields like GLJrnDtl.Description and user-defined string fields like GLJrnDtl.Character01. The default value for string field/-container is an empty string.
Date	This type defines calendar date values. You can use posting rules to populate predefined date fields like GLJrnDtl.CreateDate and user-defined date fields like GLJrnDtl.Date01. The default value for the field/container is the unknown (not selected) value.
Numeric	This type defines number (including decimal) values. You can use posting rules to populate predefined numeric fields like GLJrnDtl.FiscalPeriod and user-defined numeric fields like GLJrnDtl.Number01. The default value for the field/container is the unknown (empty) value.

Complex Data Types

The following table lists and describes complex data types.

Data Type	Description
Account	The container of an Account data type is defined as a unique structure which stores an account as a collection of segments; it provides access to the account as a whole as well as to an individual accounting segment. Each container of an Account data type requires that a book be selected for it. The COA of that book is then used as a source for the account segments list. A special option indicates whether this COA is a Master Chart. The segments and accounts are synchronized. When and account is assigned, all segments refresh with new values which derive from the new account value. When a segment is assigned, the value of the whole account is modified using the new segment value. The default value is an empty string.
GL Control	A GL control is used as a simple string in all operations, but the metadata attached to the GL control make it a complex type. The metadata stores the GL control type ID, and you use it while you modify a rule to determine the appropriate list of account and journal contexts which apply to the GL control. The default value is an empty string.
Amount	A value of type Amount which is retrieved though a function; you select an amount value through its attributes. An amount is a complex type treated as a structure with the following elements: • **Name**- the business name of the amount (Sales, Discount, and so on) • **Value**- the numeric value of the amount • **Currency**- the currency of the amount (its three-letter ISO currency code) • **Currency Type**- which its transactional or reporting currency • **Calculation Method**- a free-form string identifier that distinguishes amounts calculated by different methods (for example, a method can be for different depreciation models or different inventory costing models) • **Conversion Date**- a fixed date for currency conversions (for example, a conversion date is can be used for a currency contract or forward cover)

Containers

Containers are fields which contain the target data which results from a posting rule operation. You create target data by populating containers with specific values.

You can set up modified containers for the types described in the following sections.

Transaction Header Containers

The following containers are always the same for the entire GL transaction and are considered part of a transaction header.

The data type for each container is in square brackets:

- Comment [string]
- Call
- Journal Code [string]
- Apply Date [date]
- Closing Period Num [string]
- Red Storno

Transaction Lines Containers

An accounting transaction line can have one of following containers. The data type for each container is in square brackets:

- Credit Account [account]
- Debit Account [account]
- Book Amount [amount]
- Transaction Amount [amount]
- Transaction Text [string]
- AR Invoice Num Ref [string]
- AP Invoice Num Ref [string]
- Bank Account [account]

- Bank Slip Ref [string]
- Bank Transaction Num [string]
- Check Num Ref [string]
- Cash Receipt Ref [string]
- Vendor Ref [string]
- Temp Account [account]
- External Account [account]
- Multi Company [string]
- External Company [string]
- Reference Account [account]
- Credit Context [string]
- Debit Context [string]
- Reference Context [string]

Constants

Constants are named values run by Kinetic which are then available for a posting rule. Each constant must have a fixed value throughout as it executes within the posting rule.

The only exception is the constant Today; this constant is fixed throughout the entire posting process for a single business transaction.

Within a rule, each constant is treated in the same way as any other value. The following table lists available constants.

Name	Description	Type
Today	The current system date	Date
Book Currency	The currency code of the current book	String
Default Rate Type	Default rate type	String
Space	A space character	String

Name	Description	Type
Empty String	A blank string	String
Current Book	The current book for which GL transactions generate	String

Conditional Operators

The following table details the conditional operators available for each operation. These operators define how operations for each rule interact with each other.

Keyword	Description
If	Logical branching operator
Else	Logical branching operator, must be preceded by If
Else If	Logical branching operator, must be preceded by If
Comment	Adds a line into a rule's code which contains text you enter.
Log Error	Writes a line into the Review Journal which displays as an Error; you can enter a text description for the error.
Raise Error	Writes an error into Review Journal and stops the rule from running. The transaction lines which create or do not create depend on the status of the Debit Account and Credit Account containers.
Log Warning	Writes a line into the Review Journal which displays as a Warning; you can enter a text description for the warning.
End Rule	Stops the rule from running. The transaction lines which create or do not create depend on the status of the Debit Account and Credit Account containers.
Log Debug Message	Writes a line into the Review Journal which displays as a Debug message; you can enter a text description for this message.

System Functions

A list of the system functions is below.

These algorithms define operations common across the GL Transaction Types. Items you can modify are in bold text.

- **A = B**
- **Value**
- Lookup **Map name** For **Field** Using **arg1**, **arg2**
- Lookup **COA map name** For **Segment** Using **Segment value**
- Lookup Business Data Using **Query Field**
- Convert **Amount** To **Currency** Using **Rate Type**
- End of Period which includes **Date**
- **Account** is Valid
- **Account** is not Valid
- **Value** is Available
- **Value** is not Available
- Get Account From **GLControl** For **Current Book** AND **Context** Context
- Get **Segment** From **GLControl** For **Current Book** AND **Context** Context
- Select Amount From **Doc Line** Where **Add Condition**
- **Comment Text**
- Log Error: **Log Error Text**
- Raise error: **Error Description**
- Log warning: **Warning Description**
- End Rule
- Log Debug Message: **Log Debug Message**
- **Argument1** and **Argument2** is Available
- Lookup **COA map name** Using Account Account
- Get **Book Amount** From **GLControl** For **Current Book** AND **Context** Context
- Get External Account From **GLControl** For **Context** Context
- Concatenate **String** + **String** ...

Modifiable Items

While you modify posting rule operations, you can change the items defined in this section.

Expression

Use these items to enter an arithmetic or string expression; this expression may contain several operators.

Syntax

Expression

Links Functionality

Link	Action
Expression	Click this link to display a pop-up box with a field for text entry. You can then select a field (VBD) or a container from a list and insert it into current cursor position.

Return Value Type

Either string, numeric, or a date value. These values are determined by the field or container you select. These values can be used within any container.

Value

Use these items to enter a single value (a numeric one or a literal or field or container).

Syntax

Value

Links Functionality

Link	Action
Value	Clicking this link displays a pop-up box you can use to enter a single value. This value can be one of the following: field (VBD), container, constant, literal

Return Value Type

As the selected value. These values can be used within any container.

Lookup (variant 1)

An item with accesses a user-defined COA map.

Syntax

Lookup map name For field Using argument1, ..., argumentN

Links Functionality

Link	Action
map name	Display a list of all user maps registered within the company.
field	Display a list of field names which make up the target value of the selected map. This feature only works when the map name is defined.
argument1, ..., argumentN	Each link opens a pop-up form for you to enter an argument (a part of the source value for the selected map).

Return Value Type

Can be either any simple type or Account type (cannot be controlled at design-time). You can use this value for any container of a basic data type or of an Account (complex) data type.

Lookup (variant 2)

An item with accesses a user-defined COA map linked to the current book.

Syntax

Lookup COA map name Using Account or Segment Account

Links Functionality

Link	Action
COA map name	Display a list of all those COA maps registered in the company where one of the charts is the chart of the current book.
Account	Link only works if COA map name is specified. Open a pop-up form for entry of a key value for the lookup, which may be either an account or a segment (string), defined by the selected map's type.

Return Value Type

Can be either a simple string or Account type (depending on the map type). You can use this value for any container of types String or Account.

Lookup (variant 3)

An item with accesses a business activity query (BAQ).

Syntax

Lookup Business Data Using Query Field

Links Functionality

Link	Action
Query	Display a list of all BAQ queries linked to this particular GL transaction type.
Field	Display a list of all fields of the selected BAQ. Only works if a business activity query (BAQ) is specified.

Return Value Type

Any simple type Applicable to any container.

Convert

Currency conversion using a default rate.

Syntax

Convert Amount To Currency Using Rate Type

Links Functionality

Link	Action
Amount	Opens a pop-up form which you use to enter or select a reference to one of the modified or standard containers of type Amount.
Currency	Displays a list of currency codes. The list includes the Book Currency.
Rate Type	Display a list of currency rate types. The list includes the Default Rate Type.

Return Value Type

The function returns value of type Amount. The elements of this complex type update with the new numeric value of the amount and the new currency code. The rest of the data is pulled from the Amount value. You can use this value for any container of type Amount.

End of Period

Returns the end of the fiscal period which contains the date on the business transaction.

Syntax

End of Period which includes Date

Links Functionality

Link	Action
Date	Opens a pop-up form where you can select either a field (VBD), a container, or the constant Today.

Return Value Type

The function returns value of type Date. Applicable to containers of type Date.

Validation

Validates an account against the current COA. Available validation operations:

- **Is Valid** and **Is Not Valid** operations are used to validate accounts. An account is considered valid when it satisfies the following conditions:

  1. The account is defined as **Active**.
  2. The transaction date occurs on or between the **Effective From Date** and the **Effective To Date** range.
  3. The account's static segments match the structure defined within **Chart of Accounts Maintenance**.

  If the account does not satisfy these conditions, it is not considered valid and an error message displays within the **Review Journal**.

  Note that the validation function accepts any dynamic segments added to the end of the account structure. Because of this, dynamic segments are included in the validation process and do not return errors.

- **Is Available** and **Is Not Available** operations are used to determine if a supplied value is defined. A value is considered defined if it is a non-empty string for a string data type or a non-undefined value for a date and numeric data type.

These operations are used in the conditional operators. The "Is Not" operations are basically the same as "Is" operations. Because the Not operator is used, however, the opposite validation occurs.

Syntax

Account Is Valid

Account Is Not Valid

Value Is Available

Value Is Not Available

Links Functionality

Link	Action
Account	Display a list of fields (VBD) or containers of type Account.
Value	Same as the Value function

Return Value Type

Treated as True or False

Get Account

Pulls in an account from a GL control for a given context and the current book.

Syntax

Get Account From GL Control For Context

Links Functionality

Link	Action
GL Control	Display a list of VBD fields or containers for the GL control.
Context	Opens a pop-up form for you to either manually enter a string context identifier or select one of the VBD fields. When the GL control is defined, you can select a context identifier from the list of default account contexts available on the GL control type.

Return Value Type

The function returns value of type Account Applicable to containers of type Account.

The default account is retrieved when you supply the special context <default context>.

Book-Specific Account

Pulls an account from a GL control for a defined context and the selected book ID.

Syntax

Get Account From GL Control For Book And Context

Links Functionality

Link	Action
GL Control	Displays a list of VBD fields or containers for the GL control.
Book	Lists all books registered in Kinetic with a special additional item for Master Chart.
Context	Opens a pop-up form for you to enter a string context identifier manually or to select one of the VBD fields. When the GL control is defined, you can select a context identifier from the list of default account contexts available on the GL control type.

Return Value Type

The function returns value of type Account Applicable to containers of type Account

The default account is retrieved by entering the special context <default context>.

Get Segment

Gets a single segment from an account from GL Control.

Syntax

Get Segment From GL Control For Context

Links Functionality

Link	Action
Segment	Opens a pop-up form for you to enter segment name manually or to select one of the VBD fields or one of containers (both of String type).
GL Control	Displays a list of VBD fields or containers for the GL Control.
Context	Opens a pop-up form where you can enter a string context identifier manually or select one of the VBD fields. When the GL control is defined, you can select a context identifier from the list of default account contexts available on the GL control type.

Return Value Type

The function returns value of type String Applicable to containers of type String

The default account is retrieved by entering the special context <default context>.

Select Amount

Selects one of amounts from an Amount record by its parameters.

Syntax

Select Amount From Doc Line Type Where Condition

Links Functionality

Link	Action
Doc Line Type	The VBD document line type from which you want to select an amount. Each posting rule evaluates only one VBD line of a specific Doc Line Type, so type is enough to point to a line.
Condition	Displays a popup window you can use to construct a condition based on Currency, Currency Type, and Calculation Method. All individual conditions combine by using the AND logical operator. You can define a condition based on just some (not all) of the fields.

Return Value Type

The function returns value of type Amount Applicable to containers of type Amount.

Concatenate Strings

Constructs text from multiple strings using single formula.

Syntax

Concatenate String + String ...

Links Functionality

Link	Action
String	Opens a pop-up form for entering string text manually or selecting a VBD field or a container in case they are of type String.
String	Opens a pop-up form for entering string text manually or selecting a VBD field or a container in case they are of type String.

Link	Action
...	Adds the next string for concatenation when you click on ... in the formula: Concatenate String + String + String ...

Return Value Type

The function returns value of type String applicable to containers of type String.

User-Defined Fields in GL Transactions

Follow these steps to store data in user-defined fields linked to GL tables.

Add User-Defined Tables and Fields

Leverage User Defined Column Maintenance to create user-defined tables and add columns to this table extension. This application is available if your user account has Security Manager or Global Security Manager rights.

The Kinetic tables do not contain UD fields. You use this application to add UD columns (_c) to tables and make them available for use in your customizations. Through this utility, you select any existing table and create a user defined table (_UD) that contains columns you define. These UD tables begin as empty tables with columns **SysRowID** and **ForeignSysRowID**, and these default columns link the record back to the associated record in the main table. You next add the user-defined columns required for your customization.

In **Epicor Cloud ERP - Multi Tenant or Epicor Cloud ERP - Dedicated Tenancy**, this application or feature may not be available or may operate under certain restrictions.

1. Launch **User Defined Column Maintenance**.

   Menu Path: System Setup > System Maintenance > UD Column Maintenance

   This application is not available in Classic Web Access.

   This application may not be available, or operate under certain restrictions in cloud-based Kinetic deployments.

2. Click **New Table**. The **Table Search** window displays.
3. Click **Search** and select the **GLJrnDtl** table to customize it. Click **OK**.

The **Table Name** field automatically displays the name of the database table you selected, adding a "_UD" suffix to the end of this table.

4. Enter a short description for the table in the **Description** field.
5. The **System Code** field displays the name of Kinetic which contains the UD table. Typically ERP will appear in this field.
6. Click **Save**.
7. Click **New Column**. The **Column > Detail** card displays.
8. From the tree view, select the new column.
9. In the **Column > Detail** card, enter a name for the new column in the Column name field.

The system assigns a _c suffix to the column name.

10. Enter column description in the **Description** field.
11. Accept the defaults for Data type and Format and click **Save**.

After you set up **GLJrnDtl** UD table, repeat the steps 1-10 to make the same setup for the **RvJrnTrDtl** table.

Regenerate Data Model

After you add user-defined columns, you need to update the database. These columns will then be available to select on custom components on Application Studio layers.

Do this by regenerating the data model. You do this in two ways:

Cloud Environments - Contact Epicor Cloud Support stating that you need to regenerate the data model for your database. They will notify you after they regenerate the data model.

On-Premise Environments - You or your system administrator accesses the server that contains the database, launches the Epicor Administration Console, and then runs the Regenerate Data Model process.

To regenerate the data model in your On-Premise environment:

1. On your server machine, press **<Windows> + F** to display the **Search** bar.
2. Use one of these options to launch your search:

a. From the **Search** drop-down list, select the **Everywhere** option. In the search field, enter **Epicor Administration Console**.

b. Click on the **Apps** button.

   The Apps screen displays.

3. Locate the **Epicor Software** section on this screen and click the **Epicor Administration Console** icon.
4. From the tree view, expand the **Database Server Management** node and the node for the database server that contains the database you wish to regenerate.
5. Select the database.

   The properties for the database displays in the center pane.
6. From either the **Action** menu or the **Actions** pane, select **Regenerate Data Model**.

   The **Generate Data Model** dialog box opens.
7. Adjust or verify the settings in the dialog box:

   - **Server name** - SQL Server **server\Instance** name where the database is located. Initially populated with the database server for the selected database.
   - **Database name** - The database for which you are regenerating the database model. Initially populated with the selected database.
   - **Authentication** - If you select Windows Authentication, the **User name** and **Password** default to your current login values. If you select SQL Server Authentication, enter the user account and password you use to log into SQL Server.
8. Click **Generate**.

If you receive an error that the .dll file for the data model generation is in use by another process, end the task. Launch the Internet Information Services Manager by clicking **Start > applications > Administrative Tools > Internet Information Services (IIS) Manager**. Select Kinetic **Pools** node; the center pane displays Kinetic pools available on your system. Right-click your application server node; from the context menu, select **Recycle**. You should be able to regenerate the data model.

Likewise if you receive an error that states some tables did not synchronize, you can review the log file to see more details about these table errors. The location of this log file displays in the error message.

When you successfully regenerate the data model, a dialog box displays. Click **OK**.

9. Now to complete this process, you must pull the latest data model from the database and copy it to the local application server by recycling Kineticpool. Recycling the Kinetic pool is a mandatory task after the data model successfully regenerates. To do this, click **Start > applications > Administrative Tools > Internet Information Services (IIS) Manager**.

10. Expand the tree view and select the Kinetic **Pools** node.

    The center pane displays the Kinetic pools available on your system.

11. Right-click on the Kinetic pool for your application server; from the context menu, select **Recycle**.

    > Optionally you can also recycle the Kinetic pool within the Epicor Administration Console. To do this, expand the **Server Management** node and select your application server. From the **Actions** pane, select the **Recycle IIS Application Pool** option.

The regenerate process stages the data model in the database. When the server restarts, Kinetic checks the data model on both the disk and the database. If a new data model version is available, this version is retrieved from staging and the database is updated to include the user-defined table. Kinetic users can now view and enter data in the user-defined columns.

> When you reference these user-defined columns through either application or a customization, the columns appear to be part of the base (primary) table because the data model merges the two tables into one logical entity. To identify the UD columns, their identifiers all use the "_c" suffix.

If you need to regenerate the data model more often, consider running this process through a command line, a desktop icon, or a recurring task. These launch options are explained in the Epicor Administration Console help, Command Line Tools Guide and the System Administration Guide.

The new columns are now available to add through the **Customization Tools Dialog**. You link custom text boxes, check boxes, and other controls to these UD columns by using the **EpiBinding** property. Users can now enter data in these extended user defined columns through the customized application.

Posting Rules in Action

This section gives you some examples which show how the posting engine arrives at its results. If you compare these examples against your own posting engine results, you will better understand the logic behind the calculations.

GL Transaction Type Conversions

This topic uses three typical scenarios to describe how the conversion application updates GL transaction types after an upgrade to a new service pack, or a new installation of Kinetic.

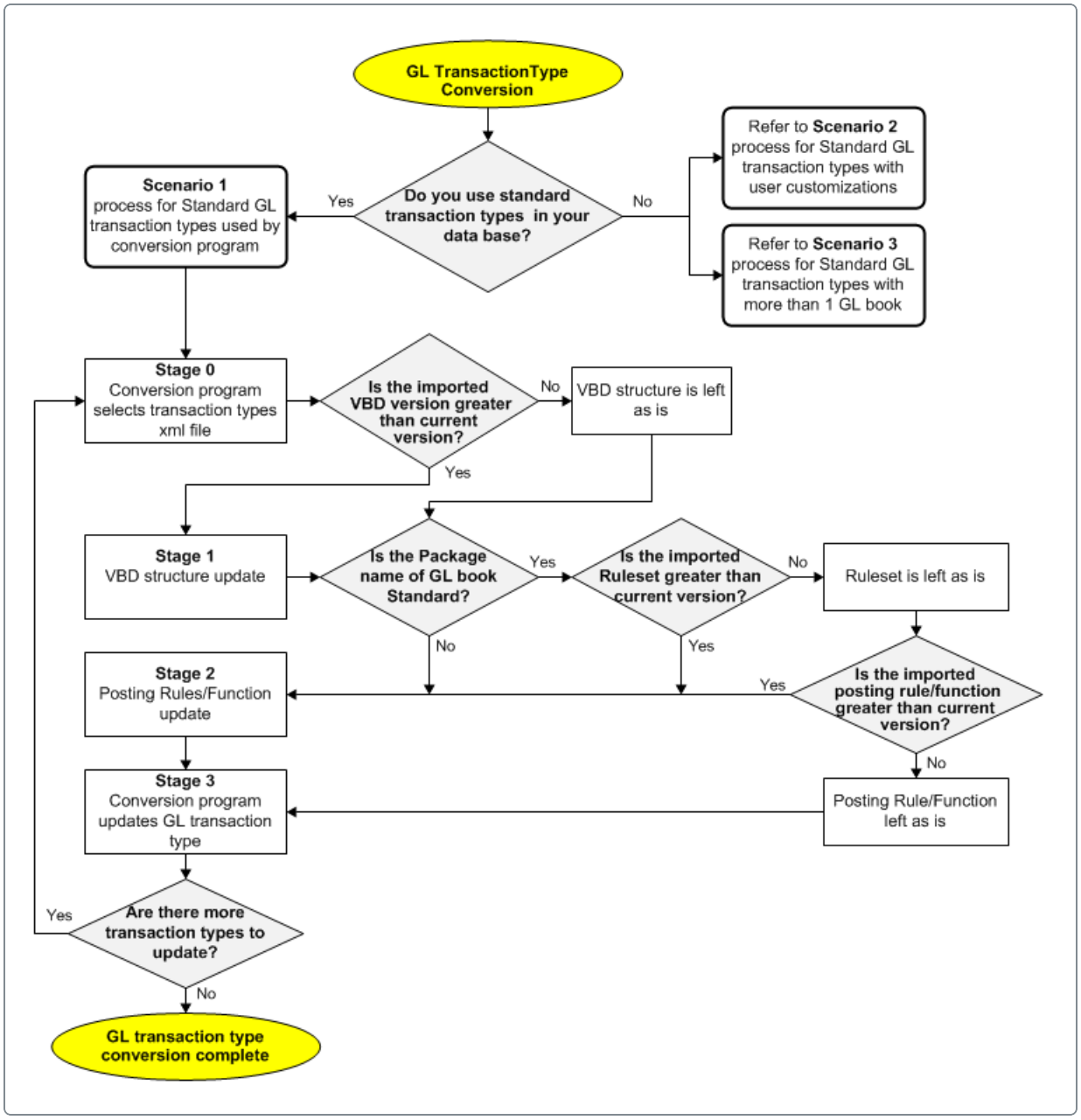
Three scenarios describe how the conversion application updates your GL transaction types, depending on how they are used in your data base.

- Standard GL transaction types with no customizations
- Standard GL transaction types with user customization in posting rules (or VBD)
- Standard GL transaction type with two or more GL books; optional user customization in posting rules (or VBD)

These scenarios are also valid for manual import of GL transaction types. If you are using manual import, you must select an xml document for the import.

Standard GL Transaction Types

The following diagram shows how the conversion application works with standard GL transaction types with a single GL book and no user customizations (**Scenario 1**).



The following stages take place when running the conversion application with Scenario 1 conditions:

Stage 0: The conversion application selects the transaction type xml from the Standard package that comes as a part of the installation upgrade pack.

Stage 1: VBD structure update (VBD, or Virtual Business Document, is the data structure of a transaction type).

1. The conversion application compares the version of the VBD in the imported transaction type xml to the version of the VBD in the currently installed transaction type.

    a. IF the VBD version in the imported xml is **greater** than the VBD version in the installed transaction type, the conversion application updates the VBD structure as follows:

        a. The conversion application adds new VBD elements to the installed transaction type.

        b. The conversion application renames the existing VBD elements that should be renamed. If there are any renamed VBD elements, the conversion application updates **all** posting rules (both standard and customized) so the posting code will be correctly referenced by its new name.

        c. The conversion application deletes the existing VBD elements that should be deleted. If there are any deleted VBD elements, the conversion application searches for **customized** posting rules that use the deleted element. If such rules are found, Kinetic renames the deleted element in the posting rules, using the prefix "DEL_". These rules must be corrected manually to use a different VBD element.

        Result: The VBD version in the installed transaction type is set equal to the VBD version in the imported xml to signify the fact that the VBD structure was updated.

    b. If the VBD version is not greater in the imported xml, the VBD structure is left as is.

Stage 2: Posting Rules update

1. The conversion application selects the single GL book that exists in the installed transaction type and checks the Package name on it.

    a. If the package name is Standard, the conversion application proceeds to the next step using the current xml.

    b. If the package name of the installed transaction type is different from Standard, (which is the case when the installation uses the Extended rules package or one of country-specific rules packages, e.g. Peru or China), the conversion application finds a folder with a corresponding package name and selects a corresponding transaction type xml from there and proceeds to the next step.

2. The conversion application compares the version of the Ruleset in the imported transaction type xml to the version of the Ruleset of the selected book in the currently installed transaction type.

    a. If the Ruleset version in the imported xml is **greater** than the Ruleset version in the installed transaction type, the conversion application proceeds to the step 3 of Stage 2 (individual posting rules and functions update). The Ruleset version in the installed

transaction type is set equal to the Ruleset version in the imported xml to signify the fact that the Ruleset was updated.

b. If the Ruleset version in the xml is not greater, the conversion application leaves the Ruleset as is and proceeds to Stage 3.

3. For **each** posting rule/function in the imported xml, the conversion application compares the version of that rule/function in the imported transaction type xml to the version of the corresponding rule/function in the currently installed transaction type.

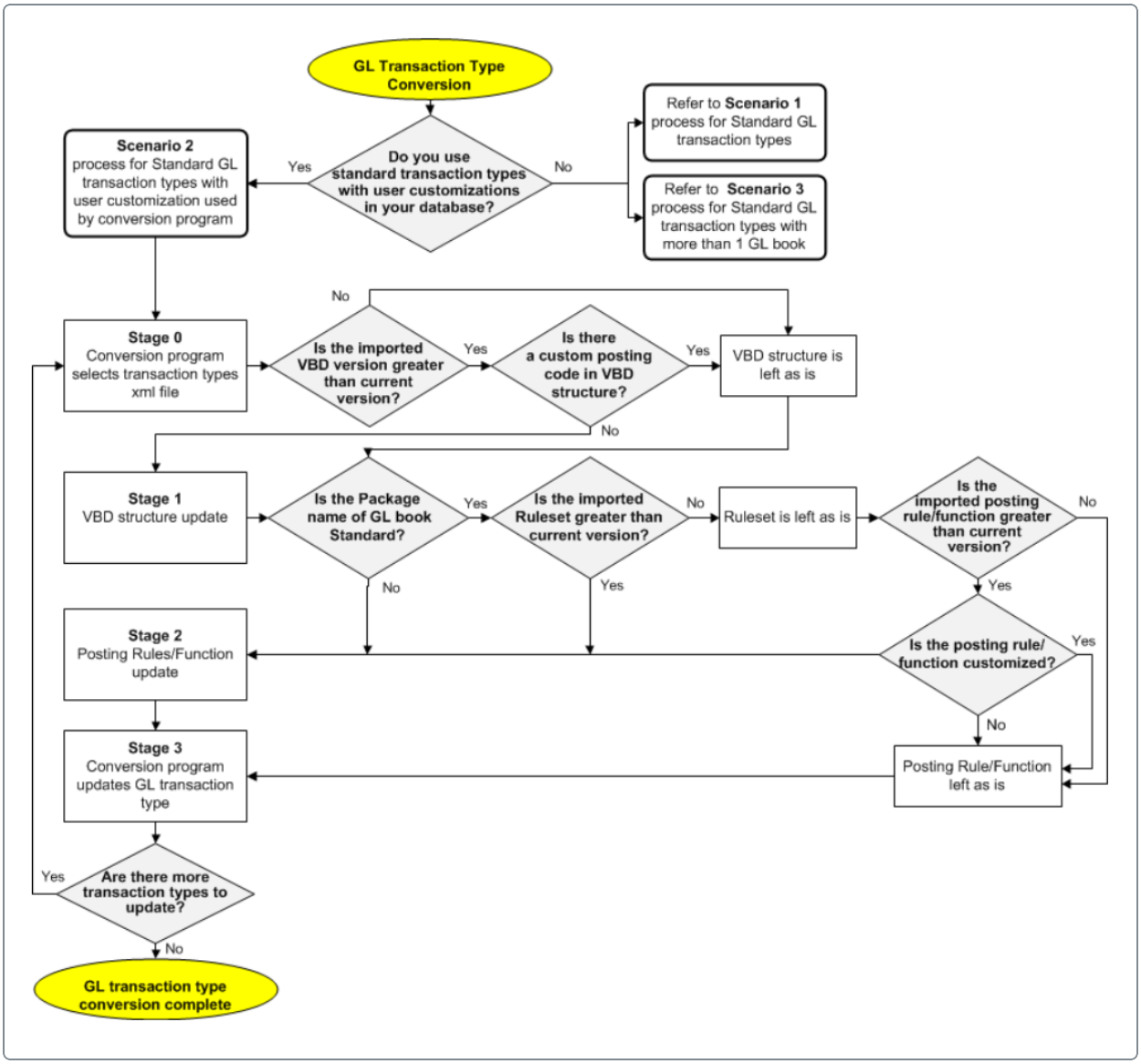
   a. If the rule/function version in the imported xml is **greater** than the rule/function version in the installed transaction type, the conversion application replaces the installed rule/function with the newer one from the imported xml. The rule/function version in the installed transaction type is set equal to the rule/function version in the imported xml to signify the fact that the rule was updated.

   b. If the rule/function version in the xml is not greater, the rule/function is left as-is.

Stage 3: The conversion application finishes the update and moves to the next GL transaction type.

Standard GL Transaction Types with Customizations

The following diagram shows how the conversion application works with standard GL transaction types with a single GL book and customizations in the posting rules or the virtual business document (VBD) (**Scenario 2**).



The following stages take place when running the conversion application with Scenario 2 conditions:

Stage 0: The conversion application selects the transaction type xml from the Standard package that comes as a part of the installation upgrade pack.

Stage 1: VBD structure update

1. The conversion application compares the version of the VBD in the imported transaction type xml to the version of the VBD in the currently installed transaction type.

a. If the VBD version in the imported xml is **greater** than the VBD version in the installed transaction type, the conversion application updates the VBD structure of the installed transaction type, adding new elements, replacing the elements that have changed, and

removing the elements that have been deleted.

a. The conversion application adds new VBD elements to the installed transaction type.

b. The conversion application renames the existing VBD elements that should be renamed. If there are any renamed VBD elements, the conversion application updates **all** posting rules (both standard and customized) so the posting code will be correctly referenced by its new name.

c. The conversion application deletes the existing VBD elements that should be deleted. If there are any deleted VBD elements, the conversion application searches for **customized** posting rules that use the deleted element. If such rules are found, Kinetic renames the deleted element in the posting rules using the prefix "DEL_". These rules must be corrected manually to use a different VBD element.

d. If the conversion application finds a custom user-defined posting code in the VBD structure, it leaves it as is **unless** it has the same name as a new **standard** posting code present in the imported xml and needs to be added to the VBD structure of the installed transaction type. In this case the conversion application marks the custom posting code as Standard, and it is used as standard from this point on.

Result: The VBD version in the installed transaction type is set equal to the VBD version in the imported xml to signify the fact that the VBD structure was updated.

b. If the VBD version is not greater in the imported xml, the VDB structure is left as is.

Stage 2: Posting Rules update

1. The conversion application selects the single GL book that exists in the installed transaction type and checks the Package name on it.

   a. If the package name is Standard, the conversion application proceeds to the next step using the current xml.

   b. If the Package name of the installed transaction type is different from Standard, (which is the case when the installation uses the Extended rules package or one of country-specific rules packages, e.g. Peru or China), the conversion application finds a folder with a corresponding package name and selects a corresponding transaction type xml from there and proceeds to the next step.

2. The conversion application compares the version of the Ruleset in the imported transaction type xml to the version of the Ruleset in the currently installed transaction type.

   a. IF the Ruleset version in the imported xml is **greater** than the Ruleset version in the installed transaction type, the conversion application proceeds to step 3 of Stage 2 (individual posting rules and functions update). The Ruleset version in the installed transaction type is set equal to the Ruleset version in the imported xml to signify the fact

that the Ruleset was updated.

b. If the Ruleset version in the xml is not greater, the conversion application leaves the Ruleset as is and proceeds to Stage 3.

3. For **each** posting rule/function in the imported xml, the conversion application compares the version of that rule/function in the imported transaction type xml to the version of the corresponding rule/function in the currently installed transaction type.

    a. If the rule/function version in the imported xml is **greater** than the rule/function version in the installed transaction type, the conversion application checks whether the rule/function in the installed transaction type is standard or customized.

        a. IF the rule/function is standard (with no changes made by the user), the conversion application replaces the installed rule/function with the newer one from the imported xml. The rule/function version in the installed Transaction Type is set equal to the rule/function version in the imported xml to signify the fact that the rule was updated.

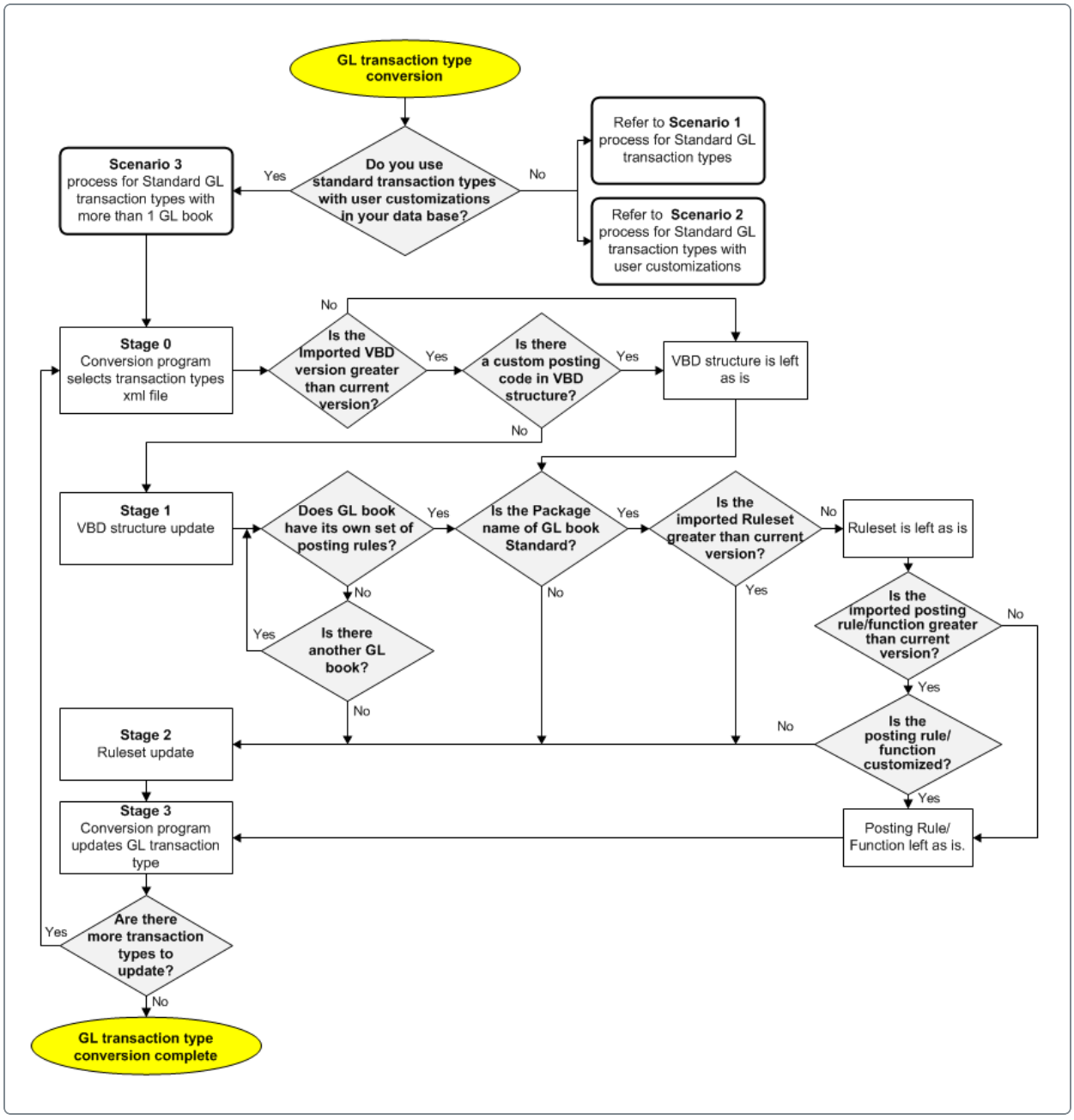
        b. If the rule/function is customized, the conversion application leaves it as is.

    b. If the rule/function version in the imported xml is not greater, the rule/function is left as is.

Stage 3: The conversion application finishes the update and moves to the next GL transaction type.

Standard GL Transaction Types with Two or More GL Books

The following diagram shows how the conversion application works with GL Transaction types with two or mores GL books. The GL books may have their own individual rulesets or use COA mapping. Some user customizations may be present in posting rules or VBD (**Scenario 3**).



The following stages take place when running the conversion application with Scenario 3 conditions:

Stage 0: The conversion application selects the transaction type xml from the Standard package that comes as a part of the installation upgrade pack.

Stage 1: VBD structure update

1. The conversion application compares the version of the VBD in the imported transaction type xml to the version of the VBD in the currently installed transaction type.

a. If the VBD version in the imported xml is **greater** than the VBD version in the installed transaction type, the conversion application updates the VBD structure of the installed transaction type, adding new elements, replacing the elements that have changed, and removing the elements that have been deleted.

    a. The conversion application adds new VBD elements to the installed transaction type.

    b. The conversion application renames the existing VBD elements that should be renamed. If there are any renamed VBD elements, the conversion application updates **all** posting rules (both standard and customized) so the posting code will be correctly referenced by its new name.

    c. The conversion application deletes the existing VBD elements that should be deleted. If there are any deleted VBD elements, the conversion application searches for **customized** posting rules that use the deleted element. If such rules are found, Kinetic renames the deleted element in the posting rules using the prefix "DEL_". These rules must be corrected manually to use a different VBD element.

    d. If the conversion application finds a custom user-defined posting code in the VBD structure, it leaves it as is **unless** it has the same name as a new **standard** posting code that is present in the imported xml and needs to be added to the VBD structure of the installed transaction type. In this case the conversion application marks the custom posting code as Standard, and it is used as standard from this point on.

    Result: The VBD version in the installed transaction type is set equal to the VBD version in the imported xml to signify the fact that the VBD structure was updated

b. If the VBD version is not greater in the imported xml, the VBD structure is left as is.

Stage 2: Rulesets update

The conversion application performs the following steps for **each** GL book in the installed transaction type.

1. The conversion application checks the selected GL book to identify whether the book has its own set of posting rules or uses Chart of Accounts mapping to generate GL transactions.

    a. If the book uses COA mapping, the conversion application skips it and moves to the next GL book in the transaction type.

    b. If the book has its own posting rules, the conversion application proceeds to step 2 (check Package name).

2. The conversion application checks the Package name on the GL book of the installed transaction type.

    a. If the package name is Standard, the conversion application proceeds to the step 3 using the current xml.

    b. If the Package name of the installed transaction type is different from Standard, (which will be the case when the installation uses the Extended rules package or one of country-specific rules packages, e.g. Peru or China) the conversion application finds a folder with a corresponding package name and selects the corresponding transaction type xml from there and proceeds to the next step.

3. The conversion application compares the version of the Ruleset in the imported transaction type xml to the version of the Ruleset in the currently installed transaction type.

    a. If the Ruleset version in the imported xml is **greater** than the Ruleset version in the installed transaction type, the conversion application proceeds to the step 4 (individual posting rules and functions update). The Ruleset version in the installed transaction type is set equal to the Ruleset version in the imported xml to signify the fact that the Ruleset was updated.

    b. If the Ruleset version in the imported xml is not greater than the conversion application the Ruleset is left as is and proceeds to Stage 3.

4. For **each** posting rule/function in the imported xml, the conversion application compares the version of that rule/function in the imported transaction type xml to the version of the corresponding rule/function in the currently installed transaction type.

    a. If the rule/function version in the imported xml is **greater** than the rule/function version in the installed transaction type, the conversion application checks whether the rule/function in the installed transaction type is standard or customized.

        a. If the rule/function is standard (with no changes made by the user), the conversion application replaces the installed rule/function with the newer one from the imported xml.

            The rule/function version in the installed transaction type is set equal to the rule/function version in the imported xml to signify the fact that the rule was updated.

        b. If the rule/function is customized, the conversion application leaves it as is.

    b. If the rule/function version is not greater in the imported xml, the rule/function is left as is.

Stage 3: The conversion application finishes the update and moves to the next GL transaction type.

Modifying Posting Rules

The posting process is a unified, multi-stage process that collects business transaction data and processes the data using user-defined rules to create appropriate general ledger (GL) transactions. **GL Transaction Type Maintenance** defines the processes you use to post accounts and journals.

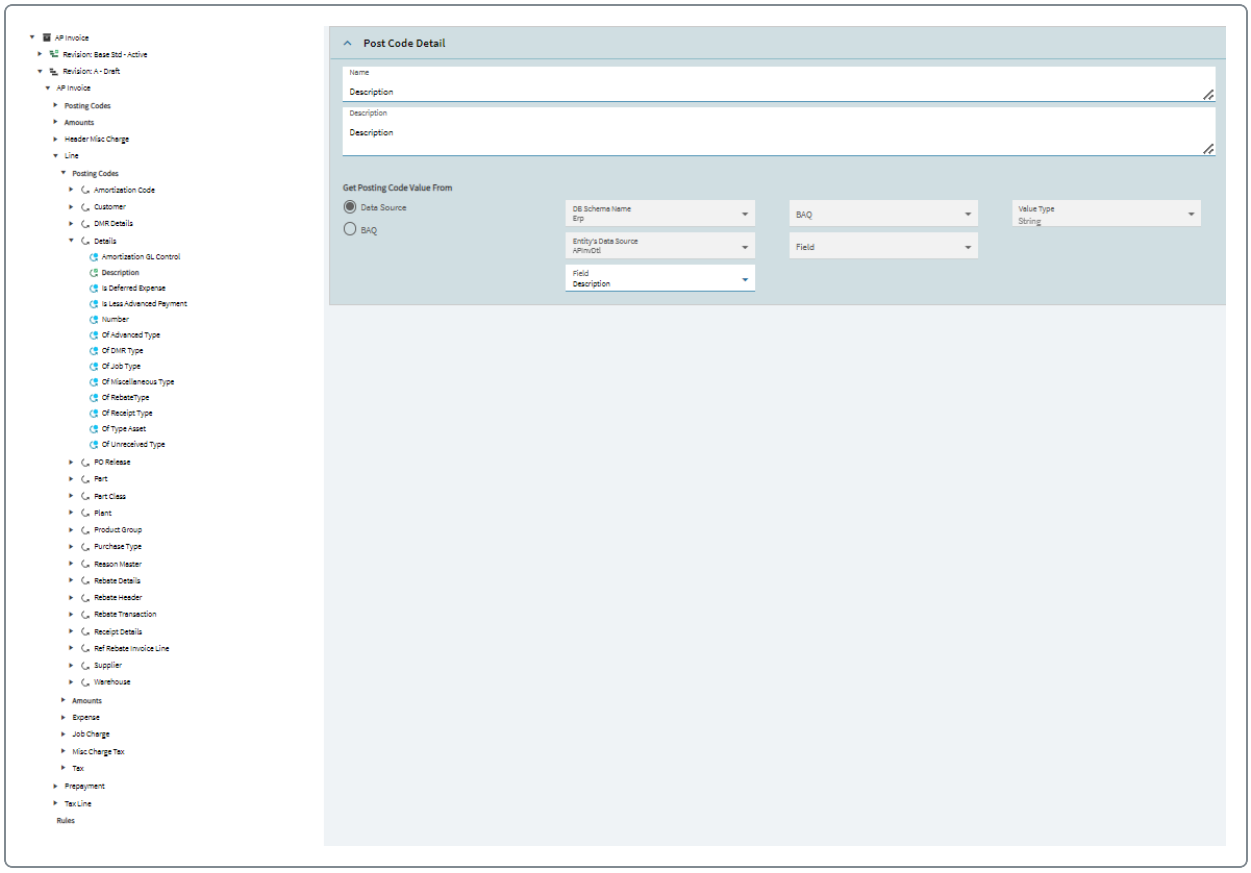
Each GL transaction type has a set of elements - posting codes, functions, and amounts - specific to its posting process. These elements contain data the posting rules need to build account and journal details. It also contains a set of rules each book uses; each rule set can contain the functions, pre-posting rules, reference rules, and posting rules needed to post business transactions which match the GL transaction type. These rules are flexible and can be modified to meet the needs of a specific book.

Amend Posting Rules and View the Results

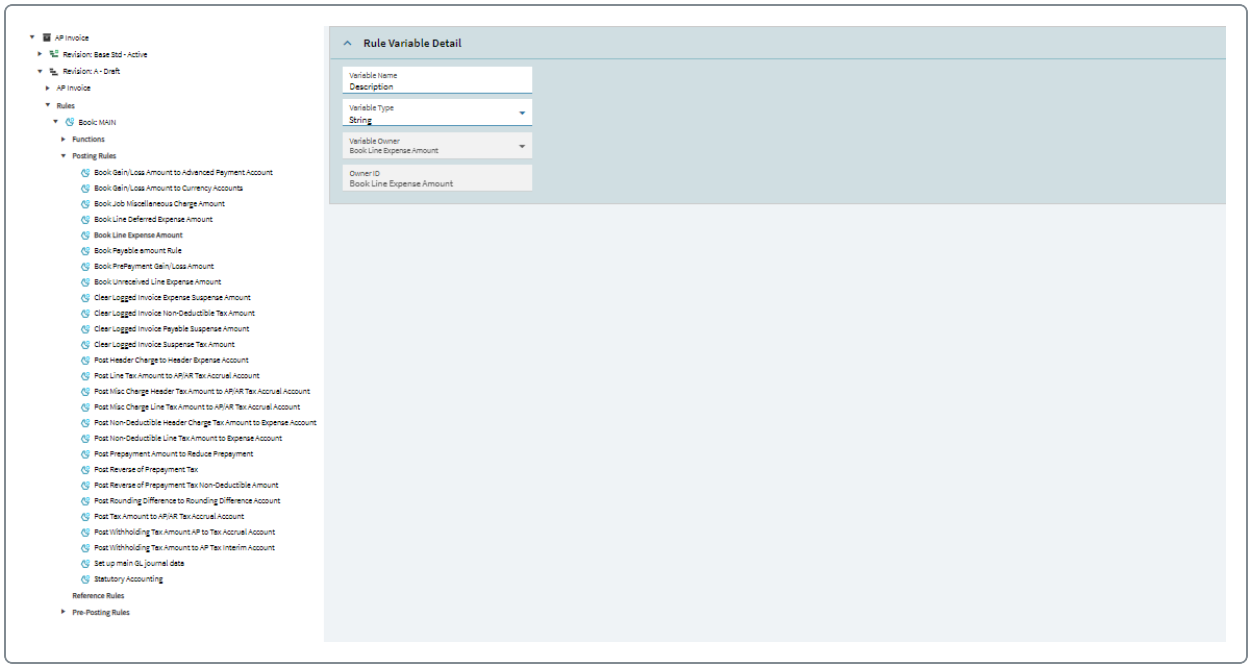
The following example shows the major steps in how to amend the posting rules to add an AP Invoice Line Description to the general ledger (GL). First you modify the customization part of the posting rules for the AP Invoice transaction type in **GL Transaction Type Maintenance**, then you enter an AP Invoice in **AP Invoice Entry** and view the results in **Journal Tracker**.

For this example, the supplier is AB Electronics and the amount of the invoice is $100.00.

1. From **GL Transaction Type**, select the **AP Invoice** transaction type and the **Add a New Revision** command.
2. From the Tree View add a **Description** node under **AP Invoice > Line > Posting Codes > Details**.
3. In the Posting Code pane of the screen, add **Description** in the **Name** and **Description** fields.
4. In the Get Posting Codes Value From pane, enter **APInvDtl** in the **Entity's Data Source** field, add **Description** to **Field**, and enter **String** in the **Value Type** field.



5. Highlight **Book Line Expense Amount** from **Rules > Posting Rules** in the Tree View and select the **New Rule Variable** command. Enter **Description** in the **Variable Name** field.



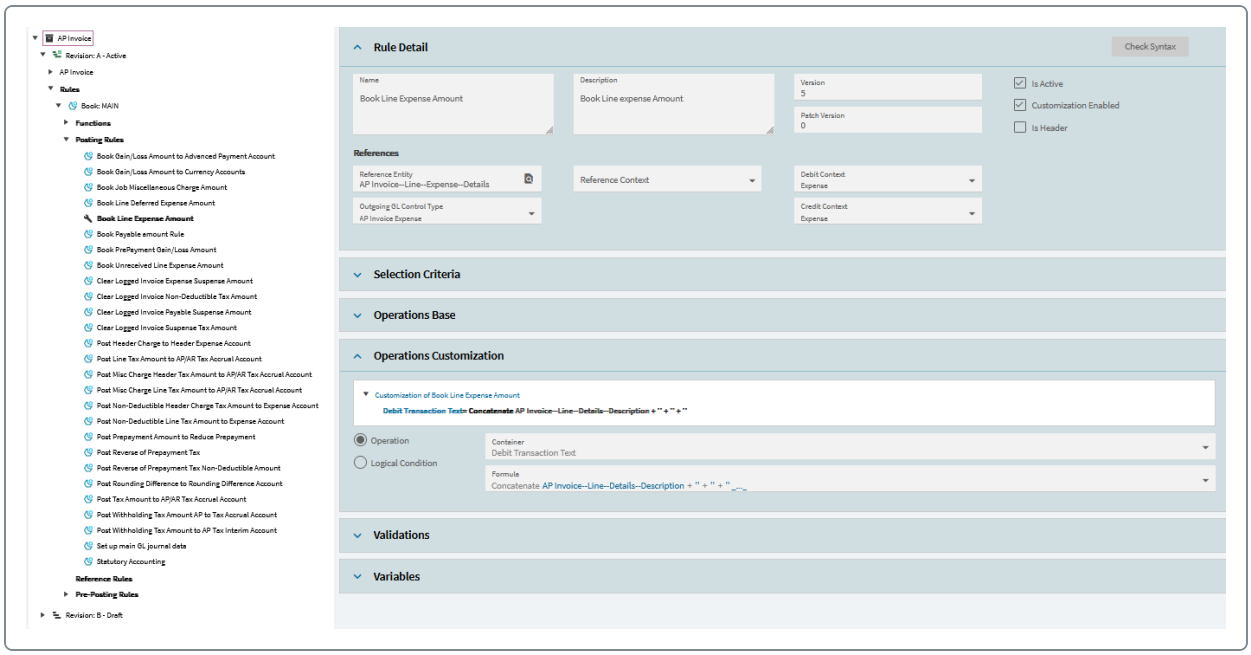
6. Navigate to **Revisions > Rules > Book: MAIN > Operations Customization**, right click on the **Customization of Book Line Expense Amount** and select the **Add**.

   A new line is added to the screen that says **Please select variable and value or function**. This is where you add your formula.

7. Select the **Operation** button and in the **Container** field select **Debit Transaction Text**. Select **Concatenate String + String_..._** from the **Formula** field. Click the underlined parameters and select the correct values to display. For example:

   - String1 - Select **ABT Field > AP Invoice > Line > Posting Codes > Details > Description**
   - String2 - Select **Value**; Delete any characters or spaces from the Enter Value screen that displays
   - String3 - Select **Value**; Delete any characters or spaces from the Enter Value screen that displays
   - String4 - Select **Value**; Delete any characters or spaces from the Enter Value screen that displays

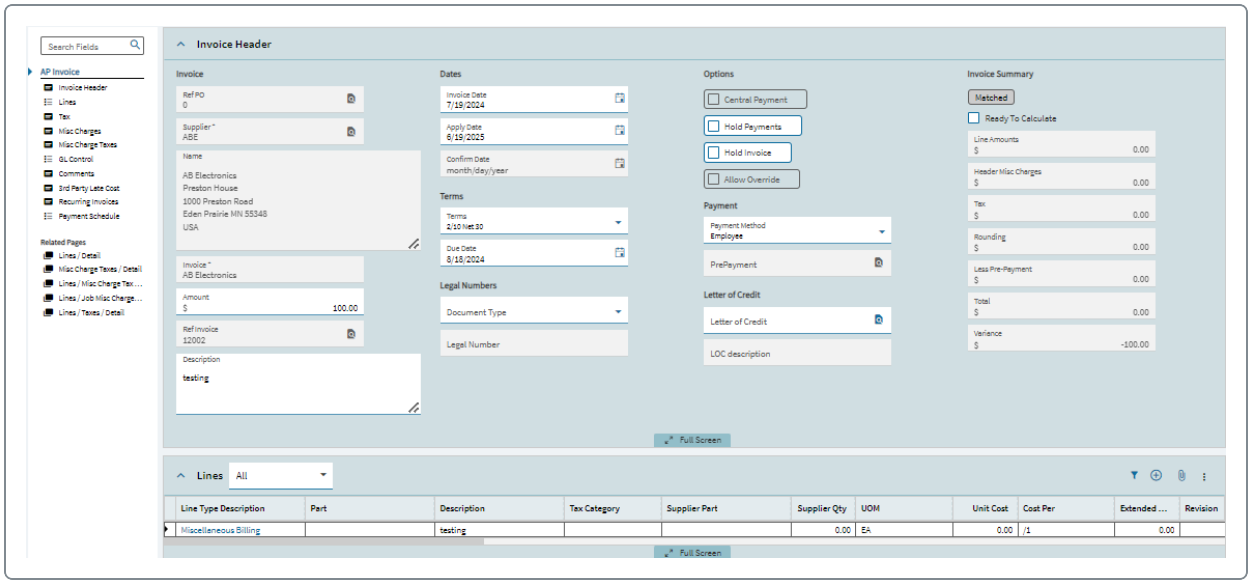
   The final phrase should look like: Debit Transaction Text = Combine String from AP Invoice-Line-Details-Description + " + " + ").



8. From **GL Transaction Type**, disable the old rule for posting AP invoices and activate the revision you created.

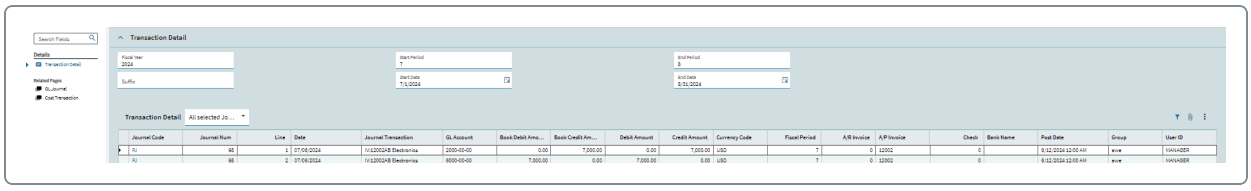
9. From **AP Invoice Entry**, create a new invoice for AB Electronics.

10. Create a new invoice line with an **Ext Amount** of **$100.00** and enter **testing** in the **Description** field of the **Line > Detail** card. You can verify the technical details of the Description field by clicking on the field and viewing the **Field Help**.



11. Post the invoice.

12. From **Journal Tracker**, review the results.



Posting Engine Troubleshooting

Use this **Posting Engine Troubleshooting** section as an aid for easier understanding and troubleshooting of errors you may receive when creating posting rules.

The following troubleshooting tools will be discussed and how to use them to find and correct errors:

- System Monitor
- Review Journal
- PE Log Viewer

A table of common errors encountered and solutions is included to aid in the troubleshooting process.

The most effective way to prevent errors from occurring is to use best practices when creating new posting rules. This section will include tips and a checklist for working with posting rules.

Error Handling

During the posting process, various issues may occur during the different stages of the posting process. This section describes these errors, how Kinetic logs them, and what you can do to correct these issues.

The posting engine can report two types of issues that may occur during the posting process:

1. **Errors**- Transactions which have errors reported against it are evaluated as invalid and moved to the Review Journal. Typical errors occur because a GL control is not set up correctly. To resolve these errors, use **GL Control Maintenance** to verify the correct accounts and journal codes are set up for the posting process.
2. **Warnings**- A warning does not prevent the GL transaction from being evaluated as valid, but each warning does notify you about a GL transaction which is considered by Kinetic as correct but which may need to be reviewed later.

When errors and warnings raise, they happen when the posting engine applies validation rules. Validation rules are defined at the book level and are applied for all generated GL transactions. Since the validation rules are both defined and applied at the book level, each book has its own validation routine.

If you modify posting rules, you can also introduce new errors and warnings into the posting engine process. Just like default rules, errors and warnings generate when the modified rule runs against an incoming GL transaction.

Fatal errors can also occur, and when they do, they terminate the posting engine process. These fatal errors can appear as a specific message while a application runs or display as an error within the System Monitor. The System Monitor is a application which automatically runs when you launch Kinetic; it is available within the Windows system tray. It displays all of the processes and reports launched within your client installation for processing at the server. When you launch the posting process from a application like AR Invoice Entry, the process displays on a card within the System Monitor.

A number of system issues generate due to invalid, inconsistent data or improper configuration. These issues are fairly predictable and Kinetic contains a list of these common issues. Each issue is classified as either an error handled through a standard method or as a special issue which can be resolved through a predefined method.

Typically the predefined methods handle the issue as an error, a warning, an automatic correction, or an ignored issue. You can define how the posting process handles these issues through parameters you define on a specific book or the book parameters within a GL transaction type revision. The following Book Validations and Posting Rule Validations sections describe this functionality.

You can also create user-defined issues through modified posting rules. To do this, you create operations which check for the error condition, which then display in the Review Journal as an error or a warning. Use this functionality to validate the financial data processed by your modified posting rules.

An error does not stop the process immediately unless for some reason it is technically impossible for the process to continue. Issues which are not ignored or automatically corrected produce either warning or errors. If a system or user-defined error occurs, the error is logged for the appropriate GL transaction -- indicating this transaction is invalid. Warnings, however, do not make a GL transaction invalid. The warning displays for your review, but the GL transaction posts to the general ledger. Both the warnings and errors log within the Review Journal.

Fatal errors can also be generated. These unexpected errors occur because of hardware failure, a bad server connection, power outage, and so on. Fatal errors are not classified by the posting engine and so are outside of the error handing functionality. You review these errors on the System Monitor. You access this application on the Windows system tray; it automatically runs each time you launch Kinetic. Use this application to see a task was started but not completed. In this case, the process can be stopped using the System Monitor tools.

Book Validations

The Validations card within **Book Maintenance** defines the overall error handling options for journals posted to a specific book. By default, books ignore most posting errors. You can change the defaults, however, so the book blocks and logs the errors.

Kinetic handles invalid journals posted to the book in one of the following ways:

- **Error**- Blocks posting of the journal. You can view the journal error within the Review Journal.
- **Warning**- Allows the journal to post. An entry appears within the Review Journal detailing the potential condition which occurred.
- **Ignore**- The default option for most posting errors, selecting this option allows the journal to post. No entry appears in the Review Journal.
- **Autocorrect**- Causes Kinetic to use a pre-defined process to correct the error. For example, Kinetic can change the Apply Date for a journal posted to a closed period to the current period.
- **Warning and Autocorrect**- Combines two actions. It first logs the warning message within the Review Journal and the runs the autocorrect process to fix the error condition within the journal.

The following table lists posting errors for which the autocorrect functionality handles. It also describes how Kinetic resolves the error.

Error	Autocorrection...
Segment is defined as 'not used,' but segment value is specified (depends on natural account)	Removes the unused segment. This occurs when the Account Segment Values application blocks the use of the segment with a specific natural account.
Closing Period is specified, but does not exist	Finds the open period for the book and posts the journal to that period instead.
Apply Date is earlier than Earliest Apply Date	Changes the apply date of the journal to the first day of the first open period.
Fiscal Period is closed.	Changes the apply date of the journal to the first day of the first open period.
Self balancing segment does not balance.	Autocorrection is applying auto-balancing algorithm.

Posting Rule Validations

The Validations card in **GL Transaction Type Maintenance** defines how Kinetic handles rules that create invalid journal details for a book included in a revision. If a journal generated by the revision is not valid, it displays as an entry within the Review Journal.

Available posting rule validations:

- **Error** - Blocks posting of the journal. You can view the journal in Review Journal.
- **Warning** - The journal posts, but a warning message displays within the Review Journal.

- **Ignore** - The journal posts, but no entry displays within the Review Journal. This action is the default for most posting errors.
- **Autocorrect** - Automatically corrects the journal using a pre-defined process. For example, the Autocorrect process changes the apply date for a journal posted to a closed period to the current period.
- **Autocorrect with Warning** - Logs a warning in the Review Journal and automatically corrects the journal using a pre-defined process.

You can use the Business Process Management functionality to define additional validations. You can create a BPM method directive which reviews the data before, during, or after a business method runs. You can then validate additional items not reviewed by the default validation rules.

Tracing

In most situations, you can trace the progress of the posting engine.

You can monitor whether posting is running or complete and whether it was successful or not. As described previously, you typically leverage the System Monitor to view the posting engine process. The System Monitor displays the transaction as a task first on the Active Tasks card, and then when it completes, on the History Tasks card. Once the posting is complete, the results are available to review within GL reports and trackers.

A Posting Engine log can also be checked using PE Log Viewer. This log is a development tool you can use to evaluate the results of your posting rules.

When the posting engine process runs, data which the posting log generates writes to this log. Each time you run the posting engine process, it adds processing information to this log. Because of this feature, you can review the history of the posting process. Be aware, however, that this log contains a lot of business call detail. Only use this log if you understand how to trace the posting results.

Posting Engine Log

The Posting Engine Log tracks the results of your posting rules. You can then evaluate how effective the posting rules are processing transactions.

When the posting engine process runs, the data generated by the posting engine is traced by this log. Each time you run the posting engine process, it adds processing information to the log results. Because of this feature, you can review the history of the posting process. However this log contains a lot of business call details, so only use this log if you understand how to trace the posting results.

Log Options

The following log options are on the **PE Log Viewer**. You access them on the **Settings** card:

- **Clear PE File** - Click this button to clear the log results.
- **Show Details** - Select this check box to display all the posting transaction information recorded by the log.
- **Show Warning on Parent** - Select this check box to display posting warnings on parent records.
- **Use Bold Font for Details** - Activate this check box to display the transaction details using a bold font.

Log Location

You launch the PE Log View from these applications:

- **GL Transaction Type** - Right click the **Transaction Type** field; from the context menu, select **Open With > PE Log Viewer**.
- **Review Journal** - Right click the **Journal Entry** field; from the context menu, select **Open With > PE Log Viewer**.

Error Message Log

If an issue causes an error message to display, you have several features for viewing the log details.

Error messages display in a dialog box. This dialog box has options for reviewing the logged details for the message. To display, print, email and/or save the error message log:

1. Click the **Detail** button to display the error message log.

   The dialog box expands, showing you the following information:

   - **Application Error** - Displays Kinetic error generated by the system.
   - **Error Detail** - Contains the primary information about the error, including the **Message** that displays, Kinetic which initiated the error, and the **Method** that generated it.
   - **Client Stack Trace** - Displays the call trace that generated from your client installation.
   - **Inner Exception** - Contains the server stack trace and the exception thrown by the system.

2. Click the **Summary** button to restore the original error message view.
3. Click the **Print** button to make a hard copy of the error message log.

   The **Print** window displays.

4. Now print the error message log.

a. From the **Name** drop-down list, select printer to which you will send Kinetic error log.

b. Click **OK**.

The error message log is printed on the selected printer.

5. Click the **Email** button to send the error details in an email.

   The **EMail Message** window displays.

   a. Type in the recipient's email address in the **To** field.

   b. If you want to send a copy of this email to other recipients, add their email addresses into the **Cc** field.

   c. In the Toolbar, click **Send**.

6. You can also copy and save the error message log to a file. To do this, in the error message dialog, click **Copy**.

   The error log is added to the clipboard.

   This action also adds some system information to the error log. This system information includes:

   - **AppServer Connection** - specifies Kinetic server address and its instance - for example, **\\ServerName\Epicor\ERP11\11.2.300.0.0**.
   - **Form Name** - specifies the name of Kinetic the error occurred in - for example, **Sales Order Entry**.
   - **Customization Name** - specifies the name of the Application Studio layer applied to the current application, if any.
   - **Menu ID** - specifies the menu ID of Kinetic- for example, **OMMT3001**.
   - **Software Version** - specifies Kineticversion - for example, **11.2.300.0.0**.

7. Open a text editor like **Notepad** or a similar text editor.

8. **Paste** the error message log.

   The error message and related system information now display in the text editor.

9. **Save** the error message log.

You now have an error message file you can send to your Epicor consultant or Epicor Technical Support.

Common Error Messages

The following table displays the most commonly reported posting rules-related errors to Support, along with solution ideas. Shown in the table is the error message, application used when receiving the error, the tool used for troubleshooting, and the most common resolution.

Error Message	Error when running...	Troubleshooting Tool	Resolution
The COSAndWIP is locked by 'userID' on ,<date> [status: Posting]	CaptureCOS/WIP Activity process	System Monitor The task is showing in the system monitor under the History tab, but is still displaying as Active.	Verify if there is anything in the General Ledger > General Operations > Review Journal related to COSAndWIP. If not, then run conversion 680 found in System Management > Upgrade/Mass Regeneration > Conversion Workbench > User Run Conversions. Scroll down to 680 and click **Exec** box. Go to Actions and Run Pending Conversions. This will unlock any transactions with no Review Journal error. Once ProgressPercent is 100%, complete exit and log back into Kinetic. You can then run another Capture COS/WIP Activity process.

Error Message	Error when running...	Troubleshooting Tool	Resolution
"Header Rule: Can not set up Journal Code"	Posting Process	Review Journal	From System Setup > Company Maintenance > Company: Navigate to Modules > All Modules > GL Control > List. GL Control type **Inventory COS And WIP** is not listed. File > New > New GL Control Type: search and select Inventory COS And WIP Control, search and select company defaults and Save.
Segment value is wrong or inactive [Line 3] COACode CDD. SegmentNumber 2. SegmentValue 00 Combination of controlled segments is invalid [Line 3] Company EPIC03. COACode CDD. Account 5100|00|AAA	• General Ledger Account Maintenance • GL Account Segment Values	Review Journal	Check to be sure that the full GL account exists and is active in **General Ledger Account Maintenance**. Check that each account segment exists and is active in **GL Account Segment Values**.

Error Message	Error when running...	Troubleshooting Tool	Resolution
Transaction Out of Balance	• COS/WIP Capture Process • Posting Process	PE Log Viewer	Use the PE Log Viewer to troubleshoot any posting with the out of balance issue. If the Out Of Balance is from the COS/WIP Capture process, there can be a large amount of data. Recommendation is to find the specific transaction type to see if the issue is a setup problem in the GL Controls. Run the Inventory/WIP report for the specific day out of balance and save to Excel. Sort the spreadcard by the Tran Type, Job, and then PartNumber fields. Subtotal each of the transaction types to verify the debit and credits match. For the transaction type that is not in balance, you can verify the GL controls required using the Inventory Hierarchy or use the PE Log Viewer.
Conversion application is stopped. GL Transaction Type is locked by another process (company:<Company>, user: <UserID>).	Any Posting	System Monitor	Make sure you complete all processes that post to GL and any GL Transaction Type import before you run the conversion application. If any posting is running, wait till it is completed, then run pending conversions.

Tools

The Posting Rules functionality contains several troubleshooting tools you can use to find and correct errors.

The error messages that display in these various tools in many cases provide you with the details necessary to correct an error. Use these tools as an aid to troubleshooting and to ensure posting rules are created correctly:

- System Monitor
- Review Journal
- Posting Engine (PE) Log Viewer
- Check Syntax Functionality

System Monitor

Use the System Monitor to verify that the posting rules-related processes, reports, and other scheduled tasks started and completed successfully.

Use the cards of the System Monitor to review the status of a task and any error messages that occur. If there is an error, the review journal number also displays.

The System Monitor is an application which automatically runs when you launch Kinetic. It displays all of the processes and reports launched within your client installation for processing at the server. When you launch the posting process from a application like AR Invoice Entry, the process displays on a card within the System Monitor.

For example, from the History Tasks > Task Detail card you can see information in the Task Log. A process can be stopped using the System Monitor tools.

Menu Path

Navigate to this application from the Main Menu:

- System Setup > System Maintenance > System Monitor

Review Journal

Use the Review Journal to adjust, re-validate, cancel, and post accounting transactions. You use the Review Journal to locate posting errors and to debug custom posting rules.

This application logs invalid GL transactions. You can display errors by module, date, and by user. You use the Review Journal error log to locate and adjust source business transactions. You can then re-run the posting process to verify the errors no longer occur.

Additionally, the Review Journal is an important tool for reviewing GL transactions which generate while you are testing a custom posting process. Use this feature to review all GL transactions created by your modified posting rules. This review makes sure the changes to the posting process generate valid, desired results.

In order for all transactions generated by the revision to display in the Review Journal, select the **Manually review all transactions** check box on the Revision Detail card of the GL transaction type revision. You can then evaluate the generated GL transactions and correct any errors the revision may cause.

You can also adjust the description and line account numbers for a transaction within the Review Journal. Leverage this feature to correct common posting errors. After you make these adjustments, you can both re-validate and post the journal. Typically you use this functionary when invalid journals generate from an external application.

You can access the **PE Log Viewer** from the Review Journal by right-clicking and selecting **Open with > PE Log Viewer** from the Journal Entry field. Or, you can access the PE Log Viewer from a right-click in the Transaction Type field in GL Transaction Type Maintenance.

The Review Journal contains a report which you can run to review the posting results. Run Print Review Journal from the Overflow menu.

Lastly, you can use the Review Journal to both cancel and confirm entries. Both actions remove the journal entries. When you confirm a journal entry, it updates the GL transaction in the book to which it posts. If the updated GL transactions are still invalid, the Review Journal displays the GL transaction again with its posting errors.

The Review Journal displays the journal entries by book. It does this because:

- Posting a single GL transaction can create different journal details in multiple books. A single journal entry groups together all journals posted to all the books by a single GL transaction. For example, posting an AR invoice can post a journal with one set of details to the company's financial book while a journal with completely different lines to the company's legal book.
- Validation settings for the books and the book node in the revision determine the errors and warnings logged in the Review Journal. The same business transaction can produce different validation results when posted to a different book. For example, one book might ignore a GL transaction that generates an error or a warning in another book.

Summarization can be applied to both natural accounts and accounting processes, so the summary lines become transactions in the Review Journal. Kinetic displays the source details for each summarized line, but no adjustments can be made.

Menu Path

Navigate to this application from the Main Menu:

- Financial Management > General Ledger > General Operations > Review Journal

PE Log Viewer

Use **PE Log Viewer** to open log information on the client. You can access the Posting Engine (PE) Log Viewer from the Main Menu or from the Review Journal or GL Transaction Type applications.

To access the PE Log Viewer from GL Transaction Type Maintenance, use a right mouse click on the Transaction Type field. To access from the Review Journal, use a right mouse click on the Journal Entry field.

With the PE Log Viewer, you can:

- Turn logging on and off
- Set detailed logging
- Clear the log file
- Export or import the log file
- Read logging

Examples of what to look for when reading the logging:

- Is account and journal code in the GL Control?
- What amounts are in the VBD?
- What amounts were selected in each rule?
- What input and output parameters were selected for the function?
- Versions of the rules and functions used?

Menu Path

Navigate to this application from the Main Menu:

- Financial Management > General Ledger > General Operations > PE Log Viewer

Check Syntax Functionality

Use the **Check Syntax** button on GL Transaction Type Maintenance > Revisions > Detail card, GL Transaction Type Maintenance > Revisions > Book > Functions Detail card, GL Transaction Type Maintenance > Revisions > Book > Booking Rules > Operations Base and Customization cards to validate check of the selected revision/rule/function for compilation errors.

If Kinetic detects an error, the revision/rule/function will be marked in the navigation tree, and the operation causing this error will be highlighted. You can hover your mouse over the highlighted operation to view the tip for the error.

Best Practices

Using best practices when creating or customizing posting rules results in saving time by creating valid and effective posting rules without having to troubleshoot error messages.

Creating a New Posting Rule

A posting rule is a constructor of the GL Journal Detail. If the posting rule Operations populate all the mandatory fields, the Journal Detail will be created. The mandatory fields are:

- Book Amount
- Credit Accounts and/or Debit Account
- Journal Code
- Apply Date

To design your new rule:

- Determine which amount shall be posted.
- Select the document line based on the amount to be posted.
- Decide which table (posting entity) will be linked to the GL and as a result will drive the logic for posting.
- Decide where to take the account from and if necessary, create or adjust the GL Controls.

When creating a new posting rule:

- Create the Selection Criteria first.
- Give the posting rule a Name.
- Enter Operations for the posting rule.
- Use the Reference GL Control to set up parameters for establishing a link.

Checklist for Creating Posting Rules

Make sure the following tasks have been completed when creating a new posting rule in **GL Transaction Type Maintenance**:

- Specify the **Selection Criteria**.
- Select the **Transactional Amounts** and **Book Amounts**.
- Make sure the Transactional Amounts and Book Amounts are positive.
- Determine the **Debit Accounts** and/or **Credit Accounts**.
- Select the **Reference GL Control** along with the contexts.

Tips for Creating Posting Rules

- You can resize both the Tree View and the Operations card. Do this as you need so that you can get a better view of various items and leverage the interface easier.
- If you need to copy a posting rule or a set of posting rules, first highlight a node on the Tree View. Then click on the Overflow menu and select either Copy Rule or Copy Ruleset. The rule or rules then appear under the selected Tree View node.
- If you enter something in error, always click the Undo button.
- You can make data available for future rules.
- All rules are based on the GL controls.
- To help you locate the GL control you may need for a posting rule, navigate to a application where you think the control is being used. When you find the GL control, right-click this field and select GL Control Maintenance from the context menu. You can then review the complete GL control record.
- Use the logical condition drop-down list to see more of an operation's context.
- The Temp Account is a user-defined variable.
- Reference accounts used within pre-posting rules and the resulting journals are different. If a reference account is used within a posting rule, the reference account is either a debit or credit account.
- When you create an If-Else condition, make sure this condition displays on the node below the operation you need, and not on the same node level. If you do not, the If-Else condition will not receive the data generated by its parent operation.
- Only parent-child relationships are available for a GL transaction type. For example, you can use a GL control from the header for use within a line rule. You cannot, however, use GL controls across child to child or from child up to parent.
- Each rule should first select the amount value then determine the account to which the amount belongs. You can also set up transaction text within the rule which automatically displays on

trackers and reports.

- Any variables you create display within the topmost level on the context menu within the Booking Rules > Operations cards.
- The base part of the posting rules is executed in the first place. Then the customized part is executed.
- Do not use a single quote (`) in any context. It generates an error.
- Each selection criterion is case sensitive, so be sure to enter items using the correct case.
- If you will populate something within a posting rule, always use an operation expression to automatically pull in the data you need.
- Once a revision is defined as Active, you can no longer change it. Instead, you will need to create a new revision based on the active revision. Only one revision for each transaction type can be active at a time.
- Blocking a revision does not affect application performance, as a blocked revision is inactive and uses no system resources. It is ignored by Kinetic.
- You can add alert messages in the code to trace what happens when a new or modified rule runs. You can then use the Review Journal to check out the alert text.
- When you are satisfied with the test results of your posting rule, be sure to clear the Manually review all transactions check box on the revision. This function is a verification step you can use to make sure the posting results are correct, but you do not want this option running when the posting rule works properly. When the check box is clear, only transactions which generate an error display within the Review Journal.

Further Steps

If you are not able to solve your issue after reviewing the Troubleshooting topics above, before you contact Epicor Support, review the Gather Information for Support topic, to ensure you can provide as much data as possible to the Epicor Support representative.

You can find current contact details for Epicor Worldwide Support on EpicWeb https://epicweb.epicor.com/resources/Lists/Worldwide%20Support%20Contacts/Allitems.aspx.

Gather Information for Support

This section lists some of the data you should prepare if you have to refer your issue to Epicor Support. The data you collect during troubleshooting may help your support representative to solve the issue more quickly.

- PE Logs (.xml)
- The full text of any error messages that displayed
- The steps in the process that lead up to the issue
- GL Transaction Type Active Revision if rules are Customized (.xml)