amdocs billing

# Billing 6.0

Specification (Internal)

**For Internal Use Only**

amdocs



## Document Information

| | |
|---|---|
| Software Version: | **6.0** |
| Publication Date: | **January 2005; updated April 2006 for SP6** |
| Catalog Number: | **212410** |

# Contents

# 1. INTRODUCTION

Amdocs Billing compiles and summarizes the charges and credits that should appear on the customer's bill. It calculates discounts and taxes, collects financial information from other components, and produces a statement that can be sent to a bill layout utility for formatting.

Billing supports Unicode. Therefore, the data (in the Billing Configurator, Billing APIs, and Billing database) that appears on the bill, or in the interfaces accessible to the CSR, can be in any local language

Amdocs Billing uses the core-customization approach to provide the ability to extend and customize the application.

## Scope of this Document

This document provides the functional specification of Amdocs Billing. It describes the bill day and ongoing processes, as well as the Customer Management, Accounts Receivable and Rater extracts.

## Billing Processes and Components

Amdocs Billing comprises the following processes and components:

- Ongoing processes/components:
  - Charge Preparation
  - Quotation Server
- Bill Day processes – Run using pipelining (parallel processing). This means that independent processes can run in parallel, waiting for completion of other processes only where necessary. This ensures high performance.

  The Bill Day flow is configurable. The operator can define which processes and sub-processes are executed in a given flow. The user can choose the mode of operation (which can be modified or customized), which results in the execution of only a subset of the processes in the flow.
  - Bill Preparation:
    - Bill Day Extracts
    - Charge Creation
    - Invoicing
    - Document Creation
    - Write to database

- Quality Assurance:
- Bill Extract
- Bill Confirmation

- Auxiliary – Billing Cleanup rules

## Charge Preparation

Charge Preparation maintains charge rates for recurring and one-time charges based on customer information. This process receives customer information by subscribing to CM transactions on the TRB. It uses Pricing Engine to rate the charges.

Charge Preparation is a daemon process that constantly listens for Customer Management, AR, OME, and OMSE transactions on the TRB. Therefore, recurring and one-time charges are calculated when they are created or changed, not on bill day.

Charge Preparation uses a multi-threaded scalable platform to support high workload and to quickly finish backlog if one is created.

The Charge Preparation daemon performs the following tasks:

- Handling of billing entities – Creating and updating the billing entities as a result of changes made by modules outside of Billing, such as CM and AR. For example, the Customer entity is updated as a result of the creation of a new customer or opening a customer in CM.
- Handling of taxation details – Keeping track of tax-affecting parameters updated by outside modules/components for bill bay tax calculations.
- One-time and recurring charge handling – OC and RC are calculated, and the charge requests are gathered waiting for bill day run to calculate the current period charges. The Charge Preparation daemon uses Pricing Engine to determine the charge rates.


*Charge Preparation uses the same process flow as Quotation Server.*

## Quotation Server

Quotation Server provides quotes for customers who want to know in advance how much a service is going to cost them. The server operates in synchronous mode, supporting online users who wait for quotations.


*The processing flow used to calculate the quote is the same as the one used to calculate the charge once the service is purchased.*

## Bill Preparation

Bill Preparation comprises processes that are run on bill day. Subsequent sections describe these processes.

### Bill Day Extracts

The bill day extracts extract data from the databases for:

- Collecting accumulated usage charge requests from Usage Rater
- Collecting one-time charge requests
- Collecting recurring charge rates

### Charge Creation

Charge Creation is responsible for:

- Prorating recurring charges
- Creating a charge for each charge request
- Generating bill day discounts

### Invoicing

Invoicing is responsible for:

- Calculating taxes
- Summarizing all invoice charges and credits
- Converting charges to a single billing currency
- Preparing prepaid statements for prepaid pay channels

### Document Creation

Document Creation prepares additional financial information that may be required for statements. It is responsible for:

- Creating bill statements:
  - Receiving from Accounts Receivable information such as adjustments, payments, and other financial activities that affect the balance of an account
  - Calculating the total amount due
- Creating invoice statements by calculating the total amount due for the invoice

## Bill on Demand

Bill on demand (BOD) processes are responsible for producing specific bills or invoices upon request, outside the context of the customer's regular billing cycle.

## Quality Assurance

Quality Assurance (QA) selects a customer population and prepares sample bills and QA reports based on it.

### Bill Extract

Bill Extract prepares outputs for the bill layout utility.

### Bill Confirmation

When bills are ready to be sent to customers, Bill Confirmation marks as confirmed all customers not rejected by QA and extracted billing arrangements.

### Billing Cleanup Rules

The generic Cleanup process cleans Billing application data according to a set of rules defined in the configuration XML file named TLS1_Cleanup_Config_BL1.xml.

This configuration file contains definitions of cleanup rules, as well as parameters of these rules. Billing uses three types of rules:

- Deleting table records according to a specific criteria
- Truncating table partitions according to a partition key
- Deleting files from the Billing file system

The cleanup rules are divided into logical groups. Each group is processed independently. Rules within a group depend on each other. This means that if one rule in a group fails, the remaining rules from this group are not processed.

The rules include parameters, which can be either static (have a constant value) or dynamic (have a value determined by an environment variable or table data).

## Billing Configurator

Billing Configurator is an application used to configure Amdocs Billing. There are two types of Billing Configurator users: Expert Users and Billing Administrators.

Expert Users (typically, implementation team members) use the application to:

- Define file structures that are received from event processing, Customer Management, Accounts Receivable, and other components that send information to Billing
- Define attributes for each charge code
- Define additional processes in the Billing process flows
- Define the number of groups that Billing uses when dividing the cycle population into groups (for performance reasons)
- Define and manage complex entities – collections of fields related to a specific topic (e.g., taxes) that can be extracted from one or more databases
- Configure the Billing Transaction Listener (BTL) to support the update of billing customer entities, customer information, and OC/RC calculations based on transaction data received from the TRB

- Manage the distribution and tracking of changes made to the core reference tables in the Version Management subsystem
- Map physical data names (such as CM attribute names) to logical names defined in Billing Configurator for the CM Extract process
- Define and manage printing categories that support the Bill Redirection functionality

Billing Administrators use the application to:

- Define FYI messages and rules
- Define tax rules
- Assign values to charge code attributes
- Define rules for QA population selection

# Billing Operation Using AMC

Application Monitoring and Control (AMC) is a generic GUI utility used by billing operators and administrators to monitor and manage billing processes for billing operation and cycle management.

## Billing Operator Activities

AMC enables billing operators to:

- Start, stop and monitor ongoing billing processes:
  - Recurring and one-time charge preparation processes
  - Subscribed transaction processes
- Start, stop and monitor cycle operations:
  - Bill preparation processes
  - Undo flow processes
  - Extract flow processes
  - Confirmation flow daemon services

## Billing Administrator Activities

AMC enables billing administrators to:

- Perform ongoing error management:
  - Charge Preparation errors
  - Subscribed transaction errors
- Perform cycle management:
  - Lock cycle for billing (Event Processing lock cycle and Rerate)
  - Activate Bill Preparation
  - Perform Bill Preparation error handling and rerun
  - Undo bill
  - Extract billing information for formatting
  - Confirm billing products

### Viewing Cycle Progress Dynamically

Dynamic views (Currently Running Flows) make it easier for users to track the progress of bill day processes and spot potential problems. These views display the following data types:

- Progress data – Includes cycle code and description, status, flow description, start time and estimated end time, and the various progress indicators (including processing rates), shown both numerically and graphically.
- Error data – Included in the dynamic view is an indication of the number of groups that have been rejected so far.

## Interaction with Other Components

Amdocs Billing includes file interfaces for communicating with other Amdocs components (such as Customer Management, Rating, Accounts Receivable, Collection, and Customer Relationship Management), as well as with external systems.

In addition, Billing subscribes to transactions published by other components (such as Customer Management and Accounts Receivable) through a uniform interface. These transactions affect entities that Billing maintains, and are logged through the TRB. Billing also publishes transactions to other components.

### Customer Management

Billing receives the following files from Customer Management:

- Customer Info – Holds customer attributes used by the Invoicing process to enable charge accumulation and tax calculation based on customer attributes.
- Snapshot – An input file for the Discount Calculation process; contains snapshots of customer hierarchies with the discounts attached to each hierarchy.
- Subscribers and Services – Holds the subscriber services, parameters, and distribution definitions. The Billing EOD process uses these files to update the OC and RC charges.

In addition, Billing retrieves customer information on an ongoing basis via BTL (and TRB).

### Accounts Receivable

Billing interacts with Accounts Receivable in the following ways:

- For bill receivers, Billing prepares a bill statement with information received from Accounts Receivable regarding the previous balance and financial activities.
- The Bill Confirmation process sends the following information to Accounts Receivable:
  - Document
  - Bill statement and invoice statement

  - Invoice
  - Billed charges
  - Taxes
- When working with Accounts Receivable, Billing Configurator is used to define the information extracted from Accounts Receivable to the bill layout utility.
- Charge Preparation receives bill increases and reductions due to balance updates.

## Product Catalog

Product Catalog contains product-related definitions used by Billing to create and calculate charges. These definitions include recurring and one-time pricing packages, as well as discount packages, charge codes, and charge structures. Billing supports Unified Product Catalog (UPC) by supporting the following:

- OC and RC calculations based on customer-set product parameters defined at the offer level
- Incremental load of sub-release changes
- Future versions – A version with a future effective date is stored in the Billing database and activated accordingly
- Vertical versions

## Event Processing

Billing interacts with the event processing components, A&F and Rating. Subsequent sections describe the interaction functions.

### Lock Cycle

Bill Preparation cannot start processing a cycle before it is locked by the event processing components.

For each cycle, the Lock Cycle process informs Billing that Bill Preparation for that cycle can start.

### Performance Indicator Extract

Billing receives charge requests for performance indicators from Rating. These PIs are the accumulated usage charges.

Billing receives the following information for each charge request:

- Charge code
- Amount
- Currency
- Attributes defined in Product Catalog per charge entity type

### Rerate

Billing uses two modes of rerate:

- Cycle mode – Rerate the entire cycle as part of the Lock Cycle request
- Customer mode – Rerate a specific customer population (This mode is invoked by the Undo process)

### Event Extract

As part of the Bill Layout extract flow, the Rater Event extract is activated to extract the billed events from the current or other cycles.

### M-commerce

The M-commerce file holds purchase transactions made through the Internet, mobile phone network, or fixed phone network. These purchases are billed through the Billing system.

The service provider sends files to A&F, which edits them and passes them to Rating. Rating sends the records to Billing, and Billing creates a one-time charge request for each record.

## Replenishment Management

In the Bill Confirmation stage, Billing notifies Replenishment Management of prepaid charges created during Bill Preparation, such as discounts, credits, and recurring charges.

In addition, OCs created during charge preparation, or by billing APIs for prepaid PCNs, are immediately reported to Replenishment Management.

## Ordering Management System

Charge preparation servers receive one-time and recurring charges due to orders. Both quote requests and charge requests are performed by the charge preparation servers.

# Related Documents

- *Billing API Reference Guide*
- *Billing Reports*
- *Billing Transactions*

# Terminology

The following terms are used in this document.

| Term | Description |
|---|---|
| Account | Financial entity that carries a balance, and is liable for paying all charges for the services assigned to it. A customer can have several accounts. |
| AR | Accounts Receivable |
| Billing Arrangement | Entity that receives an invoice or bill. Only one postpaid pay channel can be mapped to billing arrangement, though it can include several prepaid channels. An account can have several billing arrangements, to which different pay channels are mapped, to reflect the different payment methods for different services. |
| Billing Cycle | A definition of a cyclic period with a frequency and a close day. A cycle is identified by a cycle code.<br>The period (usually, one month long) between billings for a given population (Cycle Population). The term is also used to refer to a given cycle population and its processing. |
| BOD | Bill on demand. A specific bill or invoice produced upon request, outside the context of the customer's regular billing cycle. |
| BTL | Billing Transaction Listener |
| Charge | A financial activity on a billing document that includes a monetary amount that affects the bill entity balance. A charge may be a credit or a debit. |
| CM | Customer Management |
| CSP | Communications Service Provider. Any business that provides phone, Internet services, or other communications services to a population of users. |
| Cycle Close Date | The last day of the cycle period on which events, charges, and adjustments are included in the current billing document. |
| Cycle Instance | A cycle with a specific calendar period. There is a relation of 1:*n* between a Billing cycle and a cycle instance. |
| GL | General Ledger |
| OC | One-time Charge |
| Offer | A set of services to be provisioned, or event-rate definitions packaged together, that can be assigned to a subscriber. An offer may be a basic package, an additional package, or a discount plan. |
| Pay Channel | A specified payment method to which subscriber services can be mapped. A pay channel can be prepaid or postpaid. The pay channel may carry instructions of how it pays for services on a regular basis, e.g., direct debit. The pay channel might also use ad-hoc payments, such as cash payments or vouchers. A prepaid pay channel defines a prepaid balance and all services that share that balance. |

| Term | Description |
|---|---|
| PI | Performance Indicator. A data structure containing counters used to tally measures of performance (such as duration, volume, or distance), which result in usage charges. |
| Proration | The proportional calculation of charges for a specific period within the billing cycle, rather than for the entire billing cycle. Also, the proportional calculation of credits is to be applied when, for example, a subscriber pays in advance for a service, and the service is interrupted and not provided for the full cycle. |
| QA | Quality Assurance |
| RC | Recurring Charge |
| RC Rate Frequency | Defines the period for creating the RCs. Possible values are: Day, Week or Month.<br>The RC Rate Frequency does not have to be the same as the Cycle Frequency. The RC Rate Frequency together with the RC Frequency Multiplier defines the RC billing period.<br>See RC Frequency Multiplier. |
| RC Frequency Multiplier | A numeric value used as a multiplier for the RC Rate Frequency. For example: one month, two weeks, 10 days, etc. |
| Service Payer | Entity responsible for paying for the service. Usually, this entity is the pay channel related to the billing arrangement for which the document is produced. Since the pay channel has no special meaning in Billing, sometimes Billing considers the billing arrangement as the payer. |
| Service Receiver | Entity that receives the service for which the customer is debited – usually subscriber or unit agreement. |
| Subscriber | A user with a clear identification that plays a key role. A subscriber carries a status (Active, Suspended, Cancelled, etc.), and is assigned services. The exact semantics of the subscriber may be implementation-dependent. A simple case is a user with a handset. However, the subscriber can have multiple SIM cards and multiple numbers. A wireless subscriber may have ISP services as part of the same subscription, or the ISP subscription may be implemented as a separate subscriber. |

# 2. ONGOING PROCESSES

This chapter provides an overview of the ongoing Billing processes that are part of the Billing Transactions Listener (BTL). BTL processes transactions published for billing purposes. Its ongoing tasks are divided between two separate processes:

- BTL SOR (Service Order Rating, or Charge Preparation) – A daemon that constantly listens to the TRB. Performs preparation of RC and OC for bill day. Provides constant updates of the billing entities – Customer, BA, etc. – and prepares pre-calculated charges using the Pricing Engine (PE) APIs.
- BTL Quotation (Quotation Server) – A daemon that constantly listens to socket requests. Its task is to provide online OC and RC quotes for customer orders. The calculation is done using the PE APIs.


*The Billing EOD flow also uses BTL in the EOD mode. This is not an ongoing process. It is described in the Billing EOD Process chapter.*

The following diagram depicts BTL interaction with other processes and components.

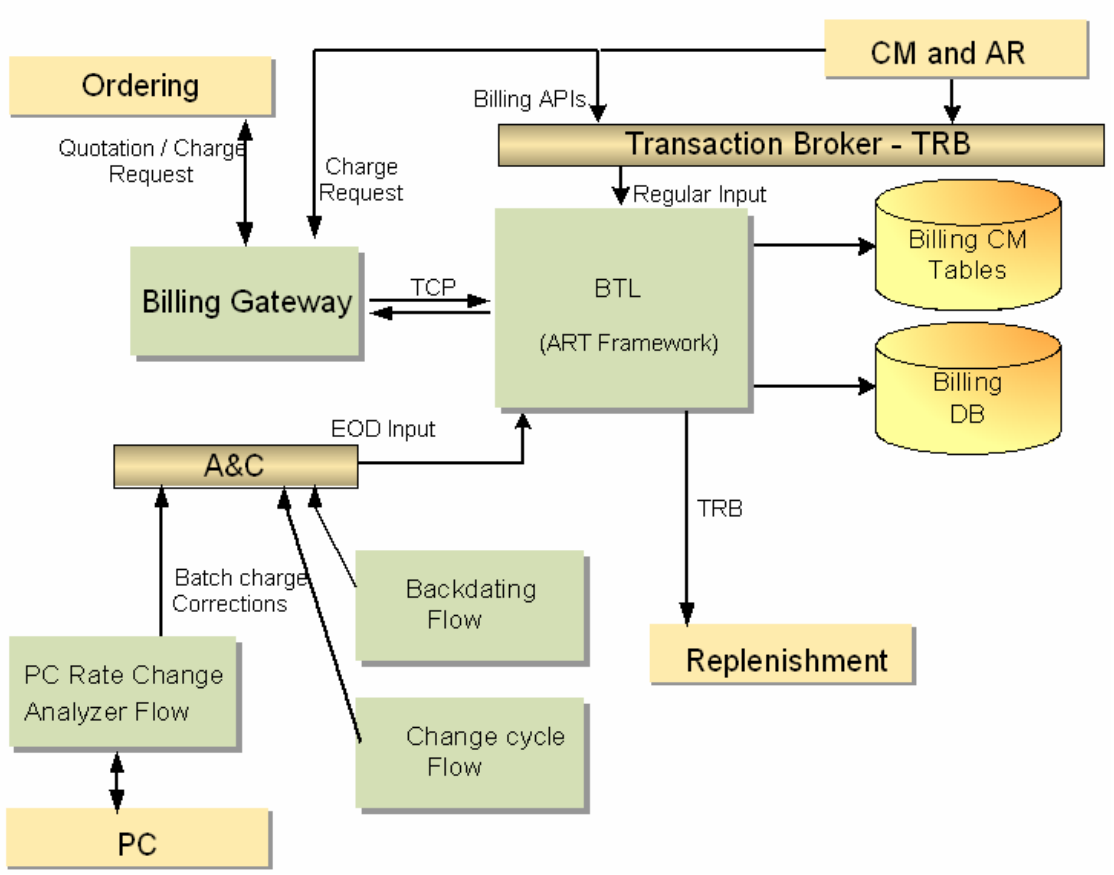


**Figure 2-1: BTL Interaction with Other Processes and Components**

# Charge Preparation

The BTL SOR process, Charge Preparation, maintains charge rates for recurring and one-time charges based on customer information provided by the Customer Management module. Charge Preparation inserts new OC charges into the Charge Request table, and inserts RC records into the RC Rates table only once. After that, the process maintains the RC records (updates or expires them) according to subsequent incoming transactions.

Charge Preparation constantly listens for transactions published on the TRB for Billing so that it can create the RCs and OCs in Billing when they are created or changed in Customer Management, and not on bill day. Charge Preparation processes each transaction group using parallel threads.

Charge Preparation handles each transaction it receives on the basis of whether the transaction's effective date is the same as or prior to the current logical date. The flows for both scenarios are provided below (see the sections "Handling Current Transactions" and "Handling Backdated Transactions").

Charge Preparation reports each rejected transaction to the TRB as it rejects the transaction and stores the transaction in the Rejected Transactions table.

The process is shut down after it has processed all the transactions relevant for the specific day during the EOD operations.

## Handling Current Transactions

If the transaction's effective date equals the current logical date, Charge Preparation logs the transaction in the Activity History table and performs the following OC, RC and customer information activities (as applicable) for the transaction:

### OC handling:

- If the charge is waived, creates a credit equal to the charge amount.
- Invokes the installment exit point (for creating several smaller charges instead of a single charge).
- Invokes an exit point for modifying the OC and then saves the changes in the database.
- If the charge is for a pre-paid pay channel, publishes a transaction to the Replenishment Manager application to inform it that a credit or debit should be applied to the balance.
- Inserts the OC charge into the Charge Request table.

### RC handling:

RC records are inserted into the RC Rates table only once and maintained (updated or expired) according to the incoming events as follows:

- Gets the effective RC rate records of the relevant service receiver ID for the relevant offer instance from the RC Rates table (if they exist).
- Allocates the relevant Billing Activities objects to handle the new RC Rates records (according to the Billing Activities list in the RC Activity Policy table).
- Uses the Billing Activities to process each new RC record in one of the following ways: insert a new record, expire an existing record, or modify an attribute of an existing record (by expiring the existing record and inserting a new one). The effective and expiration dates of the RC records are calculated according to a set of policies defined in the RC Activity Policy table.
- Accumulates and saves all required changes in the RC Rates table.

### Customer Information handling:

Charge Preparation inserts customer, BA or subscriber information into the Customer Information table.

## Handling Backdated Transactions

If the transaction's effective date is prior to the current logical date, Charge Preparation only partially processes the transaction. It updates Billing customer information but does not perform the OC and RC calculations for the transaction.

Charge Preparation performs the following actions for such transactions:

- Records the transaction in the Backdated Request table for later processing by the Billing end-of-day flow.
- If the affected subscriber already has transactions in the Backdated Requests table, marks the subscriber's effective (unprocessed) transactions for processing in the correct order (from oldest to newest).
- If the transaction is backdated to a date prior to the open cycle, marks the affected population for re-billing (see the section "Marking Customer Population for Re-Billing".

## Marking Customer Population for Re-Billing

Charge Preparation enters a record for each customer that needs to be re-billed in the Technical Mark table. It marks the customers for re-billing on the earliest date among the following dates:

- Transaction effective date
- Minimal Charge Request effective date (if it is a financial transaction)
- Minimal RC Rate effective date (of the inserted/updated records in the RC Rate table)

The Technical Mark table's main attributes are:

| Attribute | Description or Values |
|---|---|
| Entity Type | Identifies the type of entity being marked for an operation. Possible values include:<br>"Customer" |
| Entity ID | Customer ID |
| Start Date | Start date of the first impacted customer cycle instance |
| End Date | End date of the first impacted customer cycle instance |
| Process Date | End date of the last successfully processed customer cycle instance |
| Request Type | Identifies the type of request being marked. Possible values include:<br>"Re-bill" |
| Status | Status of re-bill operation for the customer. Possible values include:<br>"Pending"<br>"Running"<br>"Finished"<br>"Abort"<br>"Failed" |
| Origin | Origin of re-billing request. Possible values include:<br>"Manual" – by AMC operator.<br>"BTL" – by BTL daemon.<br>"Undo" – by Undo process. |

| Attribute | Description or Values |
|---|---|
| Issue Date | The issue date of the transaction that initiated the backdating |
| Error Code | |
| Error Description | |
| Run Request ID | Identifies the run request |

As Charge Preparation tags customers for re-billing in a particular cycle, it also updates the Cycle Customers table with the customers that are to be re-billed in that re-bill cycle.

### Customer (Service Receiver) Marking

The following types of events will automatically cause a customer to be marked for a re-bill operation in a given cycle:

- *A financial activity with an effective date earlier than the start date of the current open cycle.* The cycle for the re-bill operation is the cycle in the Cycle Customers table whose date range includes the activity effective date. In other words, the activity effective date falls between the start and close dates of the cycle.
- *New or modified charge request with an effective date earlier than the start date of the current open cycle.* The cycle for the re-bill operation is the cycle in the Cycle Customers table whose date range includes the OC effective date. In other words, the OC effective date falls between the start and close dates of the cycle.
- *New or modified RC rate with an effective or expiration date earlier than the start date of the current open cycle.* The cycles for the re-bill operation are the cycles in the Cycle Customers table intersecting with the RC rate record dates (all cycles where RC effective date <= Cycle close dates +1 and RC expiration date >= cycle start date).
- *A backdated activation, where the customer could not be found in the Cycle Customers table.* The customer is added to the Cycle Customers table instead of updated.

### Manual Marking of Customers

Customers can be manually marked for re-billing using the AMC screens that support this activity. This option allows customers that were not marked for re-billing by Billing to be added to a re-bill cycle run.

## Charge Preparation Architecture

The Charge Preparation process:

- Is multi-threaded, which ensures parallel processing and configurable scalability
- Uses exit-point architecture, which simplifies incorporation of plug-ins, and enables easy customization and implementation of each plug-in
- Updates all the Billing entities in the database to be used on bill day

# Quotation Server

The BTL Quotation process, Quotation Server, calculates OCs and RCs associated with orders and customer activities. Quotation Server receives a quote request and sends the corresponding quote response using a TCP/IP socket mechanism. It does not save the calculated values.

Quotation Server works through Billing Gateway, with which the client applications interact using the socket mechanism.

The Quotation Server APIs work in a synchronous mode to support online users who wait for the quotation.

## OMS Support

OMS sends the Quotation Server order charge requests and the Quotation Server calculates the relevant one-time and recurring charges for the order. Billing does not save these calculations unless the customer decides to purchase the product or service affected by the order charge request.

The OMS order charge request transaction includes the following information:

- Customer activity header (for example, Change).
- Customer activities – may include more than one customer activity. Each customer activity may contain dynamic activity parameters.
- Customer parameters – global customer parameters, some are static (mandatory) and some are dynamic.
- Offer instances – and for each offer instance:
  - Offer-related activity (for example, Added or Removed).
  - Offer-related activity parameters (for example, change type, override or RC amount).
  - Offer customer parameters.

Billing rates each customer activity separately according to the definitions found in the Customer Activities, Customer Offer Activities and Offer Activities tables.

## Scenarios Handled

Subsequent sections describe the scenarios Quotation Server can handle.

### Charge Calculation for Potential Orders

During the ordering process, a customer may request a quote before completing the order. Quotation Server provides an API for calculating the charges (OC and RC) associated with the order. Charge calculation takes into account the existing customer offers. The output includes at least the amount and currency of each charge, as well as the frequency (for RC).

### Charge Calculation for Customer Care Activities

Customer care activities other than orders (e.g., get bill copy, change address for cellular customer) may also be charged to the customer. Quotation Server calculates the OCs associated with these activities.

### Commitment and Penalty Calculation

Removing an offer from a customer may result in violating the commitment terms. Quotation Server calculates the remaining commitment period and the corresponding penalty charge (OC).

### Immediate Charge Calculation

Equipment items may be charged not on Bill but immediately (e.g., a customer can pay for the equipment using a credit card). Quotation Server calculates the immediate OC.

### Charge Calculation for Anonymous Customers

Quotation Server calculates all the relevant charges for anonymous (potential) customers.

### Activity Charge Creation

After a customer activity is performed, Billing provides an API for creating the charges associated with that activity (except Immediate Charges). The API may receive charges from Quotation Server or from the order information. The Ordering/CRM system can override the amount received from Quotation Server before calling the Billing API.

### Flexible Charge Calculation

Quotation Server enables flexible charge calculation, in order to fully exploit the PC implementation capabilities. OC/RC calculation can be based on dynamically defined rating events. Offer-level parameters may be used to perform quote calculations.

### Complex Quotes

Quotation Server enables complex quote calculations originating from multiple or complex business cases (e.g., installation and additional offers in a single request).

## Business Flows

Following are examples of the business flows that include quote request and charge creation.

## Service Order

A customer asks for a quote before completing an order. This is a typical case when the customer orders a Price Plan or Additional Offer.

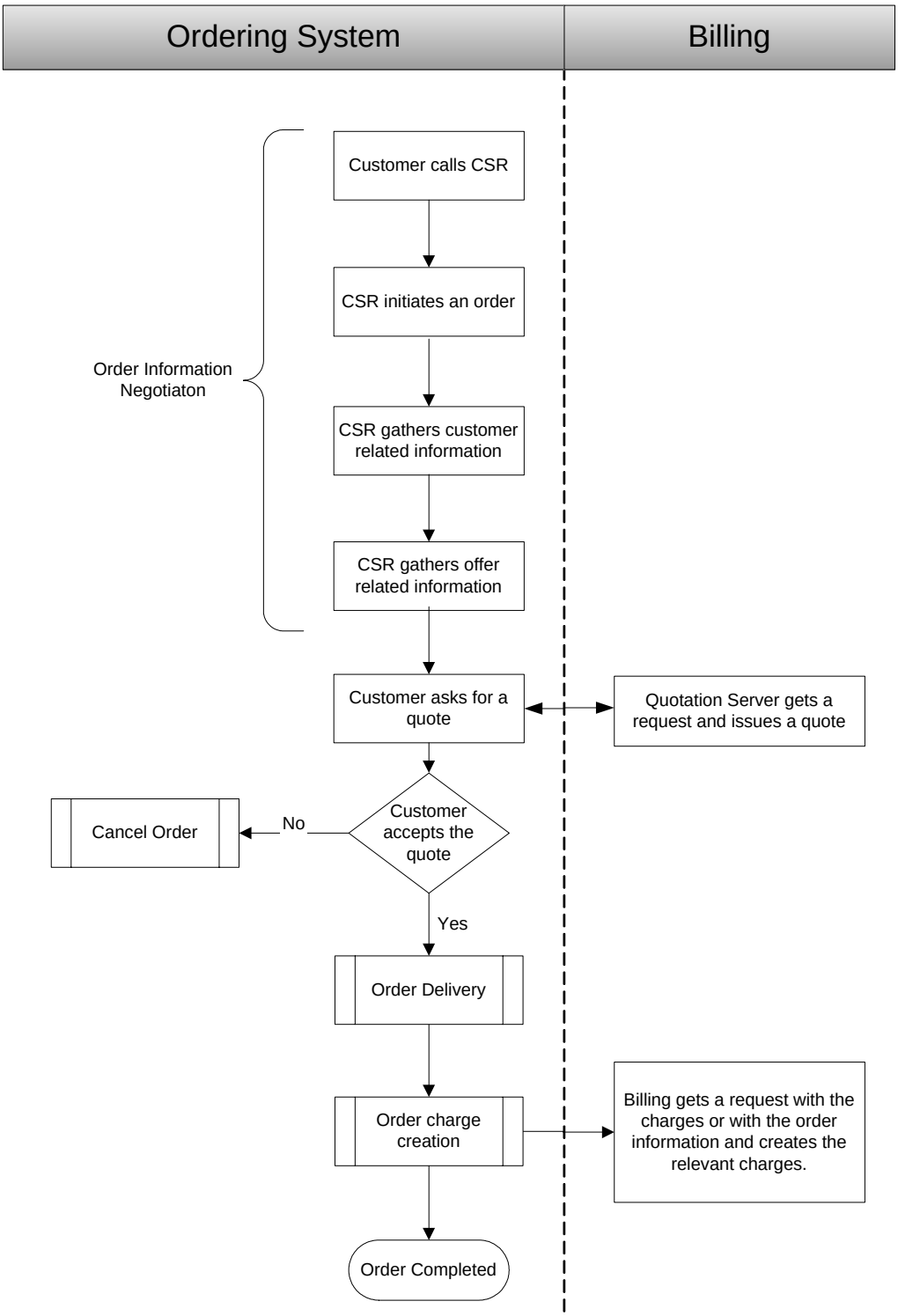


**Figure 2-2: Ordering with quotation**

## Customer Care Activity

This is a typical customer care scenario that does not involve service ordering.

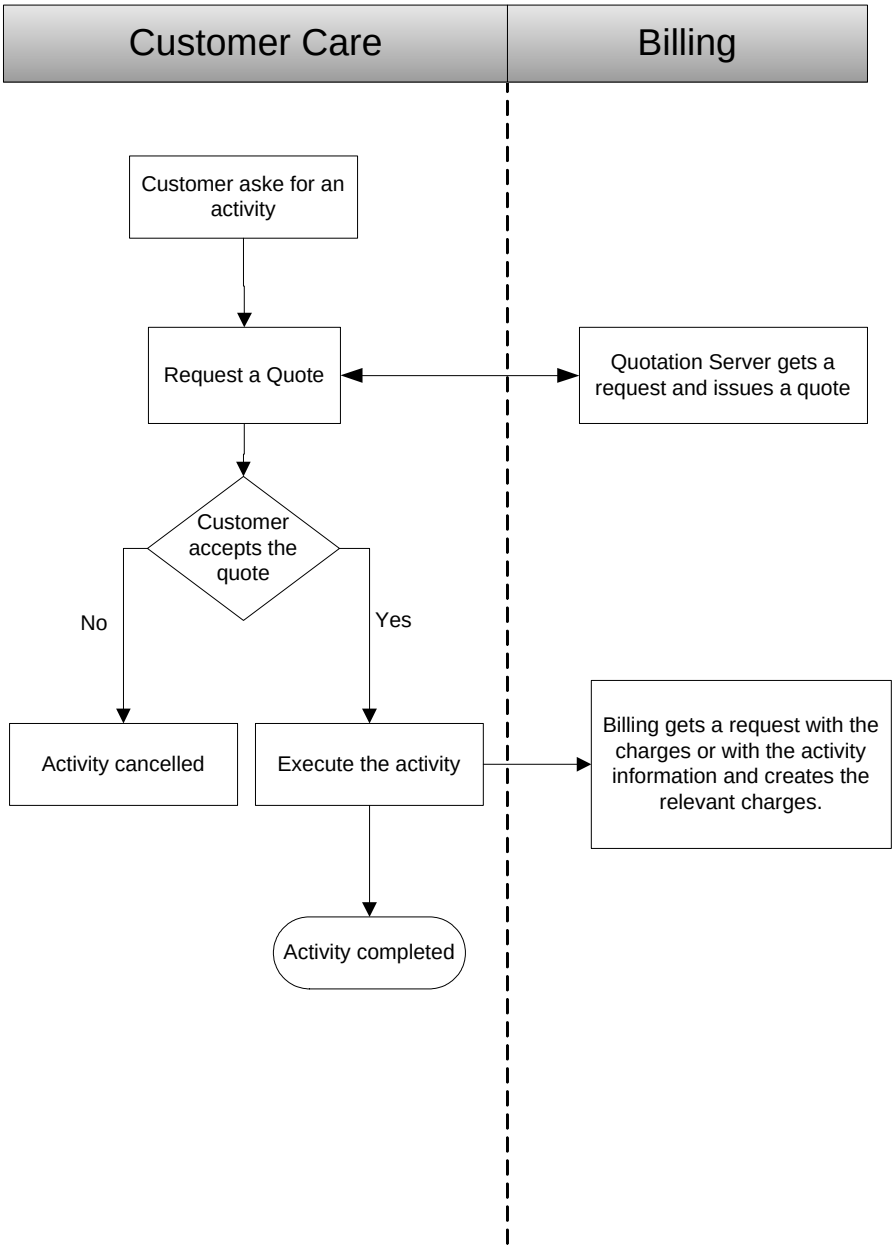


**Figure 2-3: Customer care activity with quote**

# 3. BILL DAY PROCESSES

This chapter provides an overview of the Billing processes that run on bill day. For each set of processes, it provides the process flow and a brief description of each process.

To ensure optimum performance, the strategy of pipelining the Re-rating, Bill Preparation processes, and Bill Layout Extracts is used. Each bill day process flow comprises several processes that pass information to each other using files. In addition, the entire population is divided into independent customer groups to enable parallel processing.

Each process can run in more than one instance. Each instance of a process takes an input group (that may be represented as one or more input files related to the same group), works on that group, and releases group files to subsequent processes.


*System configuration should define more instances for heavy-load processes and fewer instances for light-load processes.*

## Run Request Status

Every request to run a billing flow via AMC is logged in the Run Request table. Every Initiator process of a requested flow updates the request status to "running" and updates the start date of the flow. Every Finalization process updates the corresponding request status to "finished" and updates the end date of the flow.

In cases where Initiator has no population to extract, it automatically updates the request to "finished".

## Cycle Population

Bill day processes run on a requested cycle population. A cycle customer population has two features:

- Customers as service receivers
- Customers as service payers (customers and their billing arrangements)

Charges incurred by a service receiver are generally billed in the same cycle. However, in some cases, charges are distributed to a billing arrangement in a different cycle. For more information, see the *Cross-cycle Distribution* section.

Each customer is assigned a separate Rating Partition ID. The result is that customers belonging to the same cycle can be distributed across different rating machines, enabling better load balancing of event processing, as well as more efficient change cycle operation.

# Rerate Lock Cycle

The Rerate Lock Cycle process locks a cycle in the event processing components, and performs final rerating before the accumulated usage is extracted to Billing.

After this step, Rerating is performed in customer mode only following a request to undo the billing processes. The event processing components transfer late events to the next cycle.

For additional information about the Undo process, see the *Undo Flow* section.

Usage rerating and locking is a mandatory prerequisite before bill day processes can be activated on a given cycle.

To reduce the running time span of the billing processes on bill day, pipelining between Rerating and Bill Preparation processes allows processes to work in parallel. The significance of this is that Bill Day Initiator is invoked in parallel with the Rerate map. Bill Day Initiator waits to receive the re-rate population from the Rerate map. This population is separated into rerate groups and identified by a group type. Thus, only groups of customers requiring rerate wait for Rerating to complete prior to continuing with PI/Event extracts. Customers not requiring rerate (typically, the majority of the bill cycle customers) are automatically processed, and billing is completed without having to wait for Rerating to finish processing.

# Bill Preparation Flow

Bill Preparation processes are grouped into the following functional groups:

- Bill Day Extracts
- Charge Creation
- Invoicing
- Document Creation
- Write to Database

The following diagram displays the Bill Preparation process flow.

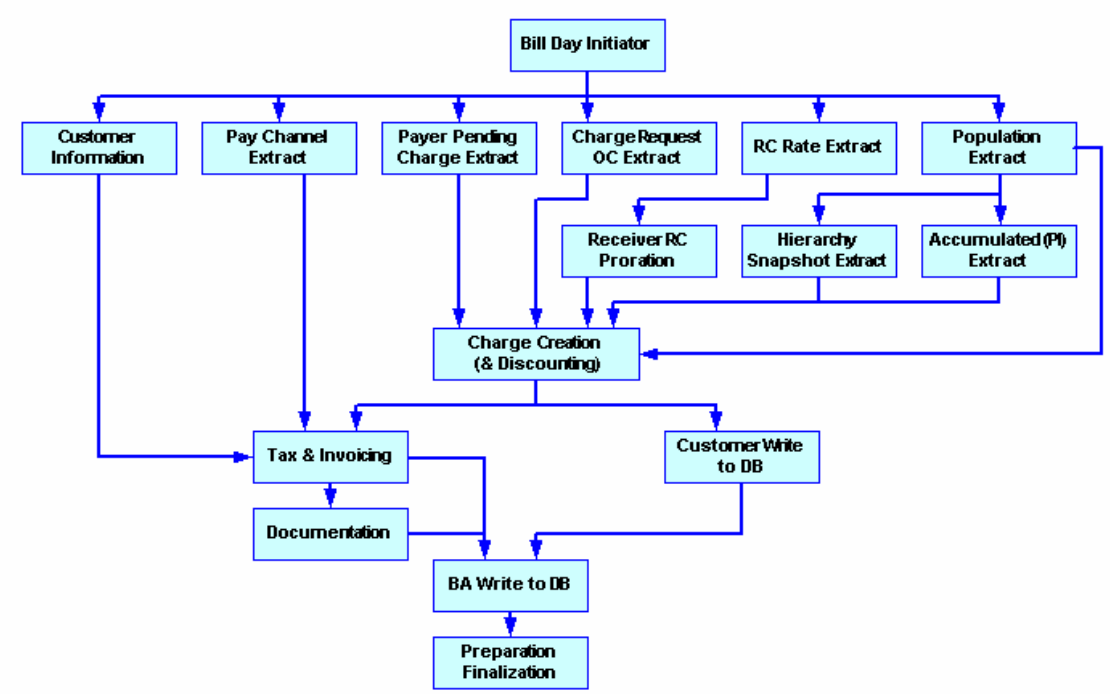


**Figure 3-1: Bill Preparation process flow**

## Bill Preparation Initiator

The Bill Preparation Initiator process extracts the cycle population from the Customer and Billing Arrangement tables and splits the entire cycle population into smaller customer groups. It also extracts the re-bill customer population for a re-billing cycle.

Each instance of the Bill Preparation process works on a single group, passes the information to the next process, and takes the next group in line.

Initiator uses the following rules when creating groups:

- Each group is closed – If customer Y is in a group, then all customers in the same cycle to which customer Y distributes its charges are also in the group.
- Rater Partition Key – Each customer is assigned a Rater Partition Key by CM. Customer groups are formed in such a way that the number of rater partition keys in each group is minimized.
- Customer Partition Key – The customer partition key is calculated based on the customer ID, and is used for the Billing table partition. For performance reasons, customer groups are formed in such a way that the number of partition keys in each group is minimized.

- The groups are balanced, and the following rules apply (as long as they do not contradict the rules above):
  - The customer is weighted by the number of subscribers and billing arrangements it owns
  - The customer group size or number of groups in the run are defined in Billing Configurator


*Group size or number of groups can be defined as fixed or approximate. The Fixed option makes the value mandatory for the code. The Approximate option serves as a guideline.*

- Customers requiring rerate are isolated in closed groups following the same rules as described above. If a customer requires rerate, and its charges are distributed to another customer that does not require rerate, both customers are grouped together into a rerate-type group.

Bill Preparation Initiator populates the cycle Customers, Payers, and Group tables with the population of the cycle being run.

Bill Preparation Initiator performs a set of checks to identify customers or BAs that are not ready for processing. The main checks are as follows:

- Customer is on hold
- Customer is rejected in BTL/Charge preparation transactions
- Customer is not confirmed in the last cycle

Depending on the re-bill reject policy defined in the Billing Configurator, the Bill Preparation Initiator does one of the following with customers that did not complete re-bill handling:

- Rejects the BAs – If a customer has incomplete re-bill handling, rejects all the customer's paying BAs from the current cycle run.
- Continues processing the BAs - Continues with the regular billing process for the BAs and customers. Defers the customer's re-bill compensation charges to the next bill.

## Customer Information Extract

The Customer Information Extract process extracts CM data records per group. This process reads the Customer Info and Cycle Customers tables, and generates a data extract file of CM information related to the Customer entity (Subscriber, BA, Customer) per group of customers.

## Accumulated (PI) Usage Extract

The Accumulated Usage Extract process runs on the customer file. It extracts the accumulated usage for customer populations into files. Each accumulated usage record becomes a charge request for Billing.

## Recurring Charges Extract

This process extracts all the recurring charge rates for subscribers or units in the population file from the RC Rates table.

This process extracts the subscriber and unit RC rates with the calculated RC rate, as they were rated by Pricing Engine based on Product Catalog definitions.

## Prorating Recurring Charges

The Prorating process is responsible for prorating RC requests according to the subscriber activities in the last cycle, and for submitting the RC requests to the Charge Creation process.

## Population Extract

This process extracts the customer and BA lists for processes depending on Bill Preparation. This process reads the file of customer keys, selects the relevant data from the database (major tables are Customers, Cycle Customer, Billing Arrangement, and Cycle Groups), and generates the following types of files per group:

- Customer – Lists all the customers in the group
- Customer billing arrangement – Lists all the billing arrangements related to the customers in the group
- Customer list for PI extract

## Hierarchy Snapshot Extract

This process extracts hierarchy snapshots for the relevant customers. A snapshot describes the customer hierarchy structure and the discount packages associated with this structure. Customer Management constructs the customer hierarchy and assigns discount packages based on the subscriber and unit agreements.

## Payer Pending Charge Extract

This process extracts pending charges from previous cycles per group. These charges result from either re-run (i.e., failure after the charges were written to the database) or cross-cycle distribution (i.e., charges that are created during processing of other cycles, and are distributed to billing arrangements in this cycle), or BAs whose frequency is greater than “1”. The extract selects all charges by BA that are not linked to an Invoice or Prepaid Statement.

## Pay Channel Extract

This process extracts all the pay channels by BA per group. It reads the file of customer keys in order to extract information from the Pay Channel and Payer Population partitioned database tables.

## Charge Request OC Extract

This process extracts OCs that were created for grouped customers during the cycle period. It reads the file of customer keys in order to extract information from the Charge Request, Payer Population, Cycle Customer, Pay Channel, and Billing Arrangement partitioned database tables. For a commerce charge, it also extracts the late charges (including those effective for the previous cycle).

## Charge Creation (and Discounting)

This process runs on both the customer and the customer billing arrangement files. It collects charge requests from the different charge request files (e.g., Accumulated Usage, Recurring Charges Request, and One-time Charge Request files) and creates credit and debit charges based on these files. In addition, this process invokes Discount Engine to calculate discount credit charges per customer (not per BA).

The outputs of this process are:

- Unified charges for the billing arrangements, generated in a file that is submitted to the Invoicing process (including the current cycle charges described below)
- Two Charge files generated per customer:
  - Current cycle charges
  - Cross-cycle distribution charges
- One charge file per BA of the current cycle, where the BA is the service receiver (e.g., late payment charges)

## Tax and Invoicing

The Invoicing process runs on the customer billing arrangement file. It summarizes charges for each billing arrangement, calculates tax for the charges, and relates the charge to an invoice or prepaid statement (that comprises all charges for prepaid pay channels).

This process uses information defined in Billing Configurator, and accumulates charges according to the accumulation definitions for each billing arrangement, (such as accumulating charges grouped by subscriber and accumulating charges grouped by pay channel).

The Invoicing process is also responsible for converting the various customer (offer) currencies to a single BA currency. While Billing maintains the charges in both the customer and BA currencies, only the BA currency is used for charge accumulations, tax calculations, and subsequent billing processes. The amounts are calculated using internal precision, and are rounded to the external (display) precision. Both internal and external precision values are configurable.

In addition, the Invoicing process links pending re-bill charges to an invoice or bill statement by linking the respective invoice's or bill's additional components entity to the respective customer charges entity.

The Invoicing process also supports the selection of BAs with regular or re-billed charges for QA and for accumulations. The *Category type* property has been added to the Invoice entity definition in the Billing Configurator. When the Invoicing process creates an invoice for a BA, it sets the *Category type* of the invoice to "Regular". While Invoicing processes the charges of the BA, if it encounters a charge with the origin "Diff", it modifies the *Category type* to "Regular with Re-bill". QA and accumulation populations can be selected according to the value in the *Category type*.

## Document Creation

This process runs on the customer billing arrangement file.

Billing produces a document for each billing arrangement. Each document is composed of one or more of the following statements:

- Invoice statement – For invoice receivers
- Bill statement – For bill receivers
- Prepaid statement – For prepaid pay channels

There are five types of documents:

- Document containing one invoice statement
- Document containing one bill statement
- Document containing one or more prepaid statements
- Document containing one invoice statement and one or more prepaid statements
- Document containing one bill statement and one or more prepaid statements

The Document Creation process creates the required statement and documents.

### Bill Receiver Statement

To create a bill receiver statement, Billing uses the following information extracted from Accounts Receivable:

- Previous Accounts Receivable balance for reconciliation
- Payments
- Adjustments
- Other Accounts Receivable activities that may affect the balance

Billing uses the previous balance of the billing arrangement and the invoice amount, along with the Accounts Receivable information, to calculate the new balance.

### Invoice Receiver Statement

Billing uses information extracted from Accounts Receivable to create an invoice receiver statement. The extracted Accounts Receivable information includes Overpayments. The Overpayments extract is optional, depending on the Accounts Receivable policy. Therefore, the file is not a mandatory input for the process.

The implementation team can use Billing Configurator to define the Accounts Receivable Overpayments extract as a process (which runs before the Invoicing process) that produces a file for the Invoicing process.

## Writing to Database

This process allows the writing of ASCII files to the database via a control mechanism. This process receives a population file and a set of data files (each file represents a database table), and writes their contents "as is" to the corresponding database table. The Bill Preparation flow includes two Write to Database processes. One updates the Customer data; the other updates the BA data.

## Bill Preparation Finalization

This process finalizes the flow when all the groups for a given cycle are processed, and then sets the new cycle status either to Processed (when the entire population is processed successfully) or Processed with Rejects (when the full population was not entirely successfully processed due to rejects). QA may be triggered automatically if the Auto QA Policy is set. Otherwise, QA is not automatically performed; it can be executed manually via the AMC screens.

## Rejects and Rerun Mode

Billing rejects bills due to lack of data integrity. Rejects can occur during all the processes described above including:

- Extract Processes
- RC Prorating
- Charge Creation
- Invoicing
- Document Creation

Reject errors have descriptions that include:

- Dynamic attributes
- Recommendations for action

Billing also generates reports on rejected customers, which can be filtered according to such criteria as error code, error type, and process. Each rejection can be displayed along with its recommendation for action.

### Extract Processes

An error in the extract process causes the entire group to be rejected.

### Charge Creation and RC Prorating

This stage involves the creation of charges. If a charge is rejected, the customer as a service receiver is rejected.

### Invoicing

If one of the customers related to the billing arrangement is rejected in the Charge Creation stage, the billing arrangement is rejected. Despite the fact that all the customers may pass successfully, a rejection can occur during invoicing or taxation. In this case, the billing arrangement is marked as Rejected, even though the customer's charges are committed to the database.

When the cycle is in Rerun mode and the problem is fixed, Charge Creation does not run again on customers with completed charges. The charge information is extracted from the database, and only Invoicing and Document Creation run again.

### Document Creation

If a billing arrangement is rejected during Document Creation, the invoice information is committed to the database, and the billing arrangement is marked as Rejected during the Document Creation stage.

When the cycle is rerun and the problem is fixed, Charge Creation and Invoicing do not run again. The invoice information is extracted from the database, and only Document Creation runs again.

## Re-bill Flow

Re-bill processes are grouped into the following functional groups:

- Bill Day Extracts
- Charge Creation

The re-bill flow is based on the Bill Preparation charge creation and discounting phase and uses the Bill Preparation processes with the modifications described below.

The re-bill flow prepares the customer charges and discounts based on the revised backdated information, and then calculates the difference between the new calculation and the last calculation for the affected cycle. The difference is stored in the Billing database as a pending charge, which is then included in the regular invoice for the open billing cycle. The re-calculated charge will be used as the basis for comparison if there is a subsequent re-bill operation.

The following diagram shows how two re-bill runs for two different cycles are incorporated into the regular run for the open cycle.

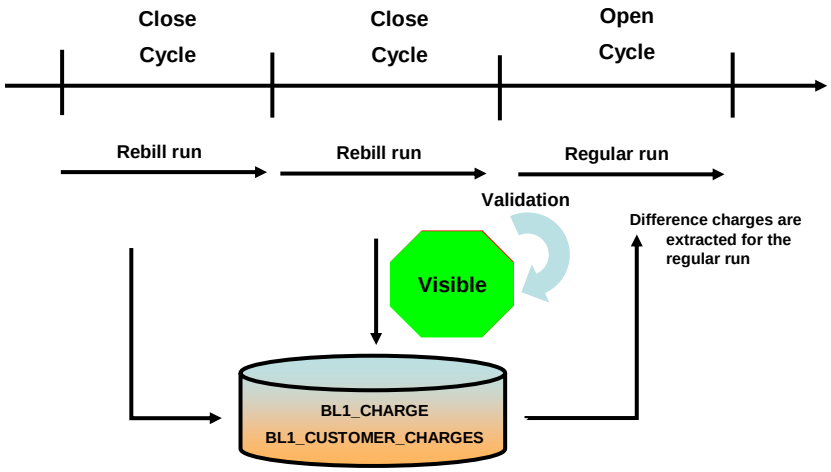


Figure 3-2 Re-bill to Regular Run Flow

# Re-bill Scope

This section discusses the Re-bill scope.

## Customer Management (CM) Activities

Any Customer Management activity can have an effective date that is in the past.

The following list of activities is delivered as part of the out-of-the-box implementation:

- Change offer, including offer parameters
- Add offer
- Remove offer
- Activate subscriber
- Cancel subscriber
- Suspend subscriber
- Restore subscriber
- Move subscriber within a customer hierarchy
- Change customer parameters
- Renew Subscriber

## Product Catalog Changes

A Product Catalog version may be released with a past effective date. The Re-bill process applies the impact of these backdated changes to the offers of the customer and subscriber.

## RC Backdating

Recurring charges, including their proration, can be backdated. Discounts that are affected by such recurring charges are re-calculated.

### Usage Charges and Charge Level Adjustments

Usage charges and charge level adjustments can not be backdated.

### Difference Charges

The difference charges (detailed credits or debits) appear on the current bill with the original period date.

### Multiple Backdating

There can be multiple backdated transactions for the same cycle period.

### Quotation Server Support

The Quotation Server supports calculations for backdated transactions.

## Re-bill Functional Overview

The main functions of the Re-bill process are discussed below.

### Marking Customers for Re-billing

The Billing Transaction Listener EOD process creates a record for each customer requiring re-billing in the Technical Mark table.

### Scheduling Re-bill Initiator Process

The Re-bill Initiator identifies the customer population for re-billing and the cycle instances for which the customers should be re-billed.

The Re-bill Initiator can be scheduled to run at any time before the regular bill run for the current open cycle. If the Re-bill Initiator is run at the same time or after the regular bill run, the re-bill results will affect the next regular billing cycle run. In extreme cases, where the difference charges need to be available daily, the process should be activated immediately after the BTL EOD process, in fact, as part of the EOD map.

### Managing Visibility/Invisibility of Re-billed Transaction Charges

A customer is considered to have completed the re-bill flow successfully after it has been processed successfully in all the cycle instances runs. When this happens, the re-billed charges (the differences or pending charges) that were written to the database become visible (visible to a regular run and to the APIs). If the customer has failed in even one of its cycle instance runs, the re-bill operation for that customer is considered to have failed. In this case, any charges that were written to the database will remain invisible (APIs will not get them and the regular run will not extract them for billing).

### Using Customer Charges Entity for Visibility/Invisibility

Re-bill uses the Customer Charges entity to manage the visibility/invisibility of charges. The Customer Charges entity does not contain charges; it contains references to the required charge fields in the Charges table, making it possible to group the charges of a specific run and to define the visibility status of all the charges created in that run.

### Handling Re-bill Run Errors

When a customer does not complete the re-bill flow successfully due to an error in one of the cycle instances, customer processing is halted. If the error occurred after the customer successfully completed one or more re-bill cycle instances, most likely, re-billed charges were already written to the database. These charges are rendered invisible.

Errors are reprocessed automatically when the next re-bill run for the customer is activated. When the customer is re-processed in a subsequent re-bill run (for the next cycle instance), the customer is resumed from the cycle instance that failed, after the process identifies which cycle instances succeeded When the customer successfully completes the re-bill flow for all the required cycle instances, all the re-billed charges become visible.

### Undoing Visible Charges

The Customer Charges entity also indicates whether the visible charges can be undone or not. The entity holds several attributes that reflect if the charges have been extracted for the billing of a BA or if a subsequent re-bill run has used these charges for calculating a new difference (diff on diff). In either case, the charges cannot be undone. If the charges can be undone (for example, if an additional backdated request is received before the charges are extracted again), the re-bill flow deletes the re-calculated charges, the difference charges, and the customer charges from the database.

### Re-calculating Charges

The re-bill flow uses the Charge Creation process to re-calculate the charges. The re-calculated charges are saved in the Charge Re-calculation table, which is used internally in the re-bill flow. These charges are not visible to the regular bill day flow or to the APIs.

### Creating Difference (Pending) Charges

The Difference Creation process compares the re-calculated charges of the re-bill run with those calculated for the last run of the same cycle (that run could have been a regular run or a re-bill run) and creates the difference charges (the difference between the charges calculated for the two runs).

### Extracting Difference Charges for Billing

The difference charges generated in the re-bill flow are stored in the Charge table and are treated as pending charges. In the next regular Bill Preparation run for the BA, these charges are extracted (by the Payer Pending Charges Extract), taxed, and accumulated to an invoice. The Bill Preparation flow marks the charges as charges that were extracted to a bill (links customer charges to a Bill Added Components entity).

### Performing QA for BAs with Re-billed Charges

The re-billed charges undergo QA in the regular cycle run. BAs with re-bill charges can be selected for QA using the new property *Category type* in the Invoice entity. When an invoice contains re-billed charges, the *Category type* of the invoice is “Regular with Re-bill” (as opposed to “Regular” for an invoice without re-billed charges). Billing Configurator enables the

implementer to define QA rules based on the category types in the invoice entity.

### Performing Undo due to QA

As the re-billed charges are examined by the QA team when the charges are attached to a bill, it may be necessary to correct the re-billed charges at this time. In this case, a request for "Invoicing Undo with Re-bill" is issued for the BA. This request causes two operations to be performed: 1) The BA undergoes invoicing undo. 2) The affected customer is marked for a repeat re-bill operation.


*"Invoicing Undo with Re-bill" is a new Undo type. AMC allows the operator to issue the Undo request.*

### Confirming Re-billed Charges

Re-billed charges are logically confirmed when the document that includes the charges is confirmed. There is no special handling in the confirmation flow for any re-bill activity.

## Re-bill Processes

Figure 3-3 Re-bill Process Flow depicts how the Re-bill process works.

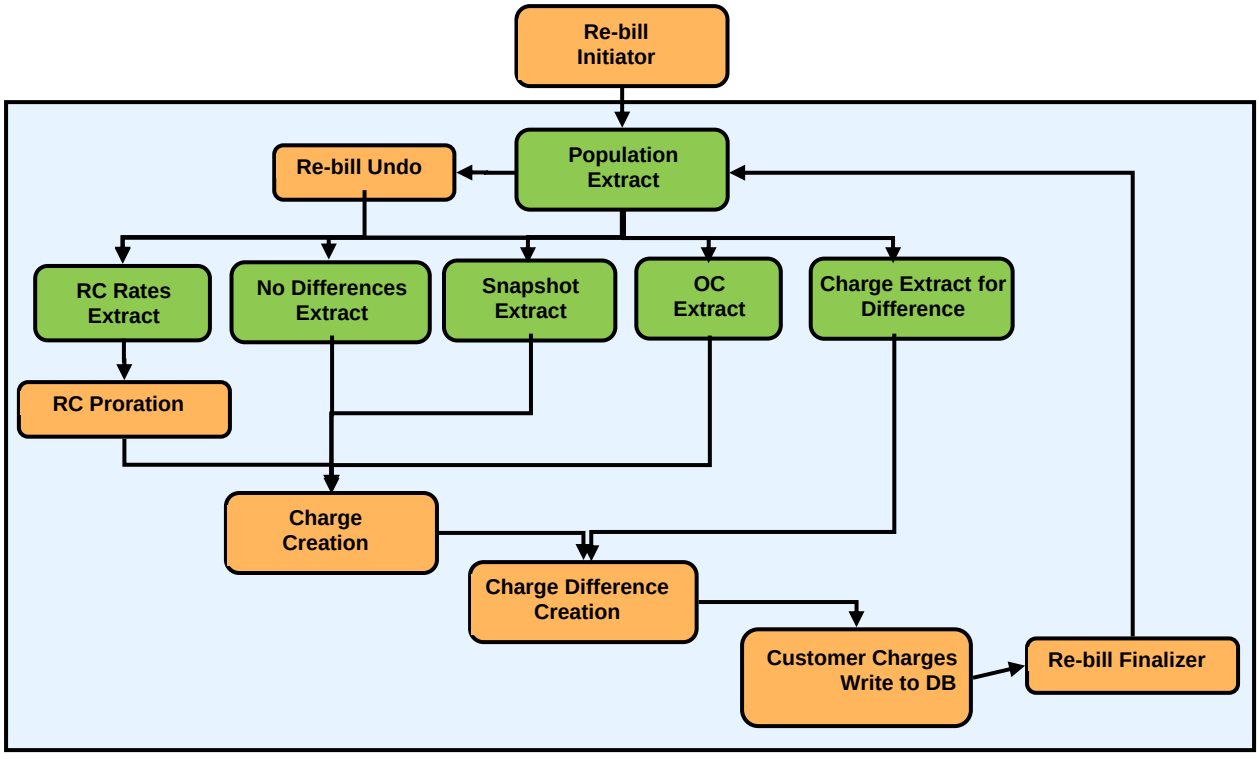


**Figure 3-3 Re-bill Process Flow**

The processes in the diagram are discussed in the following sections.

### Re-bill Initiator Process

The Re-bill Initiator is responsible for initiating the re-bill flow. The re-bill flow can be activated manually or as part of the end of day (EOD) map, after the backdate EOD process is completed.

The Re-bill Initiator identifies the customer or BA population for re-billing and the cycle instances for which the customers or BAs should be re-billed.

The Re-bill Initiator organizes the customers or BAs into groups and they are processed as groups for the entire run – that is, in all the relevant cycle instances. A group is assigned a life cycle ID, which identifies that group from the first cycle instance in which it participates through the last cycle instance. The group advances to the next cycle instance when all the customers in the group have completed the run in the current cycle instance. The Re-Bill Finalizer sends a dummy file to the Population Extract process to notify it to process the group in the next cycle instance.

The Re-Bill Initiator is responsible for the following tasks:

- Determining the cycles and the population to run
- Creating groups for the entire run. Unlike a cycle initiator, which creates groups only for one cycle instance, the Re-bill Initiator creates groups for all the cycle codes and all the cycle instances, which participate in the run.

The Re-Bill Initiator process can be activated in one of the following modes:

- Full – Process the entire customer population marked for re-billing.
- Cycle – Process the customer population of a given cycle code.
- Origin – Process records from a specific origin (for example, Undo or Manual).

### Re-bill Undo Process

The Re-Bill Undo process is part of the re-billing flow and is activated immediately after the Re-Bill Initiator. It deletes all the customer charges and re-calculated charges of the customer population participating in the current re-bill run.

The Re-bill Initiator process creates two new entities in the re-bill run: customer charges and re-calculated charges. The Re-bill Undo process deletes these entities.


*The re-bill run requires a different approach than the regular run due to the automatic nature of the customer population marking and the need to run a lot of cycles with small populations.*

### Population Extract

The process extracts a list of customers to be re-billed and a list of customers to be undone. The BA list and the PI list are not extracted as they are for regular runs, because the customers associated with them are not part of the re-bill flow. The process determines if a customer is a candidate for the undo process by examining the customer charges entities.

The extract receives two input files: the customer keys file and a dummy file sent by the Re-bill Finalizer. The dummy file allows the Finalizer to synchronize between the re-bill groups.

### RC Rates Extract

The RC Rates Extract process extracts all the RC information required by the RC Proration process from the RC Rates and RC Creation Frequency tables.

Some RC rates have an RC Frequency Multiplier greater than one. For such RC rates, the RC Creation Frequency table contains a record with the last date on which the respective RC was created.

Based on this date, the RC Rates Extract process uses a yes/no indicator to notify the RC Proration engine whether or not to create a new RC when it performs re-billing calculations. The yes/no indicator is used to prevent the creation of duplicate RCs when the RC Proration process re-calculates proration in the re-bill mode.

### RC Proration

The RC Proration process performs the RC proration calculations the same way as the in Bill Preparation flow.

If the RC Rates Extract process set the yes/no indicator to "yes", then RC Proration creates a new record in the RC Creation Frequency table for the newly created charge.

### No Difference Charges Extract

This process extracts the usage and commerce charges that were created in the original billing run and not modified since.

### Snapshot Extract

This process performs the same functions as it does in the Bill Preparation flow.

### OC Extract

This process extracts all the pending OC charges from the charge request created by the Billing Transaction Listener or from the charge request created manually through the Create Charge API. The process extracts the pending OC charges that have an effective date that falls between the cycle start date and cycle close date, regardless of whether a charge was created for them or not in a regular bill run or another re-bill run.

### Charge Extract for Difference

This extract process provides the input for the Difference Creation process. It extracts all the charges from the previous run (regular or re-bill). If the previous run was a regular run, the charges are extracted from Charge table; if the previous run was a re-bill run, the charges are extracted from the Re-calculated Charges table.

### Charge Creation

In the re-bill flow, the Charge Creation process re-calculates charges and saves them in the Re-calculated Charges table. This table is used internally, and is not visible to the regular Bill Preparation flow or to the Billing APIs.

The process performs the regular Bill Preparation flow plus the following tasks:

- Handles customers without dealing with the customers' BAs.
- Populates the Cycle Sequence Run field of every created charge entity with the cycle's sequence run. In addition, updates the Customer Status field in the Cycle Customers table to the "Re-billed" status.

### Charge Difference Creation

This process determines if there is any difference between the re-calculated charges and the calculated charges of the last run (regular or re-bill). If the two charges are not the same, a credit charge (charge * -1) is created for the older charge and a new charge is created for the re-calculated amount (full credit and full charge instead of a charge for the delta). The newly created charges are set to the Charge Origin "Diff".

The old charges and the re-calculated charges are compared on the basis of the following key:

- Receiver Customer ID
- Receiver ID
- Receiver Type
- Pay Channel
- Offer
- Offer Instance
- Offer Item
- Charge Code
- Item ID

The following table shows the possible charge comparison scenarios:

| Old | New | Diff | Comment |
|---|---|---|---|
| $10 | $10 | - | No difference |
| $10 | $9 | -$10<br>+$9 | Difference – old charge is credited and new charge is created |
| $10 | - | -$10 | Old charge is deleted |
| - | $10 | +$10 | New charge is created |

### Customer Charges Write To DB

This process receives two new files from the re-bill process:

- Re-calculated Charges
- Customer Charges

In addition, if a customer was rejected, the process skips the customer and does not write the customer data to the database.

#### Re-Bill Finalizer

The Re-Bill Finalizer is responsible for tracking the progress of the re-bill flow groups. As each group completes the re-bill flow, it is updated to the status "finished" in the Cycle Groups table. If a group is dependent on another group, the second group is released for processing as well.

When the last group of a chain is finished, the customer charges of all the customers that were not rejected become visible for all instances of the chain.

# QA Population Sampling Flow

The QA Population Sampling flow contains the processes responsible for finalizing the QA population and extracting its products (either the document that the BA receives or a customer charge report) to create sample bills and reports for QA.

The QA Population Sampling flow is activated after Bill Preparation (which sets the QA flag to "Ready") of the entire cycle is complete, and includes the processes described in subsequent sections.

## QA Initiator

Bill Preparation processes select customers and billing arrangements matching the QA criteria defined in Billing Configurator. QA Initiator finalizes the list of customers and billing arrangements, removes rejected billing arrangements, and verifies that each QA criterion has the number of requested customers (or billing arrangements) as initially defined.

Entities for QA can be selected manually via AMC. Once selected manually, an entity will always be part of the QA population (e.g., even if it is rejected).

## Bill Layout Extracts for QA

The Bill Layout Extract flow is activated on the QA population. This process extracts files for the bill layout utility. It is identical to the extract process for the entire cycle, except that it only extracts the QA population.

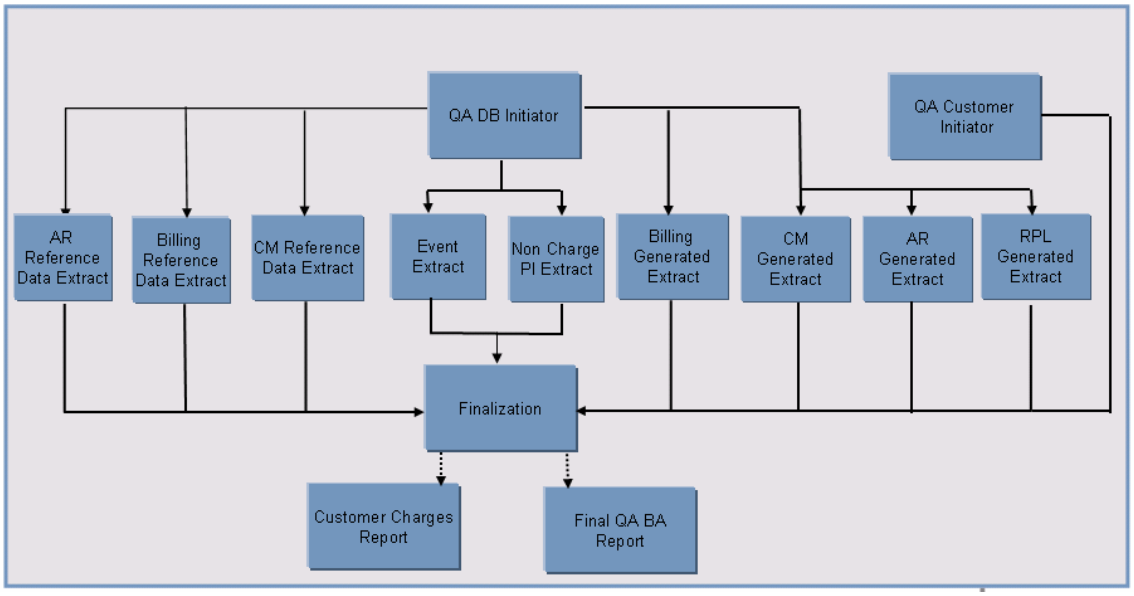


Figure 3-4: QA Extract flow

## Customer Charges Report

To perform QA on the customer charges, and not only on the resulting bills, the Customer Charges report is produced for the selected population.

The report is an unformatted flat file. Any reporting tool can take this file and format it to a printable report.

## QA Finalization

This process returns the cycle's QA flag to None, indicating that the QA extract is complete.

# Bill Layout Extract Flow

Bill Layout extracts are defined using Billing Configurator, and the flow is activated for the approved service payer population for a given cycle. Figure 3-5 displays the Bill Layout Extract flow.

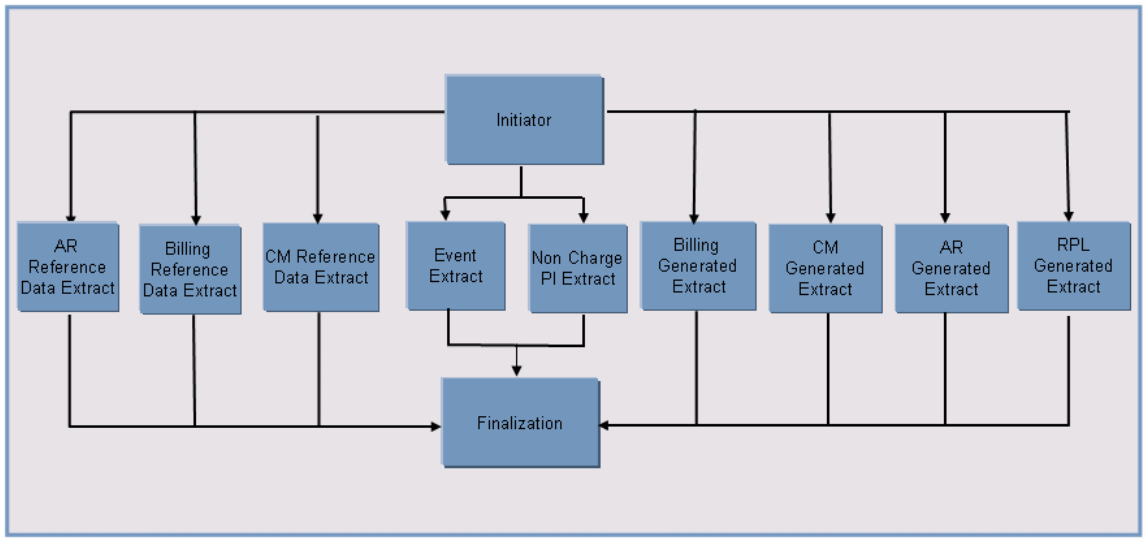


**Figure 3-5: Bill Layout Extract flow**

## Initiator

The Initiator activity can be run via either of the following two processes:

- Extract Initiator
- Bill Day Initiator


*When the Bill Day Initiator process runs, the Billing and AR-generated extract processes wait until an indication is set by the Bill Preparation Finalization process.*

The Initiator process is determined based on the flow activated via AMC. This process divides the population into groups, as defined in Billing Configurator. The number of groups for the extract is usually different from that used in the Bill Preparation flow, and can be configured to match the bill layout utility capabilities. All other processes in the flow work on customer files as defined in this process.

In preparation of the groups to extract, the document bill date and due date are calculated as follows:

- Bill Date – Extract Run Date + Added Days to Bill Date constant (as defined in Billing Configurator)
- Payment Due Date – Bill Date + Due Days (as defined for the billing arrangement)

An exit point can be used to change Bill Date for the Due Date calculation in the customization layer. The Due Days Initial value is derived from the

customer type/payment method of the payer customer, and can be overridden using a billing API.

## Reference Data Extract

The Extract Reference Data process (Billing, CM, and AR) is activated regardless of population and includes reference data required by the bill layout utility, such as the relation between message code and message text.

## Generic (Non-Reference Data) Extract

Extracts from the Amdocs Billing, Accounts Receivable, Replenishment, and Customer Management components are defined in Billing Configurator. Extract definition includes the information to be extracted and the file format. The Extract process takes the extract definitions from Billing Configurator and performs extracts according to them.

The information extracted from the system usually includes:

- Billing information:
  - Documents
  - Statements and invoices
  - Charges
  - Taxes
  - Accumulations
- Customer Management information:
  - Names and addresses
  - Bill formats
- Accounts Receivable:
  - Payments
  - Credits
  - Refunds
  - Reversal
- Replenishment – Recharges and cancellation

## Usage Extract

### Rated Events Extract

The Rated Events Extract process extracts the events invoiced in the cycle to enable call detail and usage archiving.


*The Rated Events extract works on subscribers from the current cycle only. If a billing arrangement receives charges from other cycles, their respective events are not extracted.*

### Non-charge Event Processing PI Extract

The Product Catalog user can define performance indicators (PIs) for purposes other than charge creation, such as allowances.

This extract is provided for billing arrangements whose subscribers are:

- In the same cycle with the arrangements
- Were charged in other cycles

### Extract Finalization

The Extract Finalization process updates the extract dates on all the BAs in the group that finished, and rejects BAs that failed the Rater extracts (Event and Non-charge PI extracts). This process creates a population file for Bill Formatter, which includes all the BAs that were successfully processed.

## Undo Flow

The Undo flow (Figure 3-6) enables a CSP to cancel bill preparation for a selected sub-population, or for the entire cycle population. Undo can be performed only for unconfirmed results.

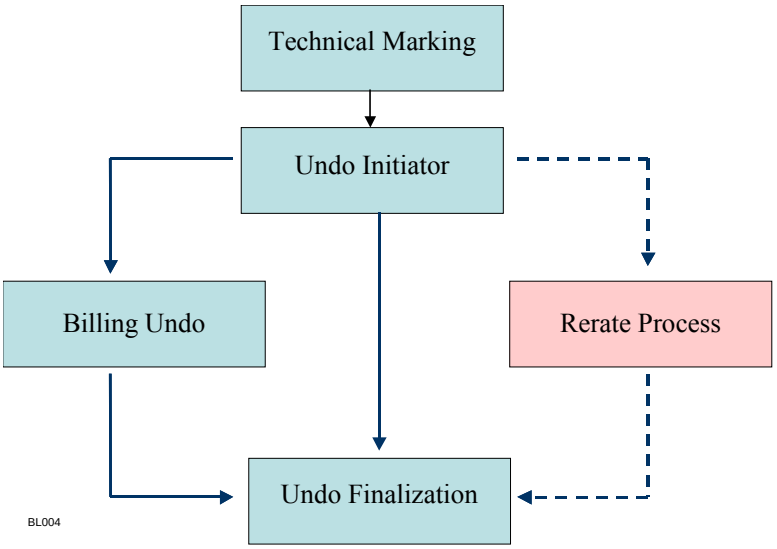


**Figure 3-6: Undo Preparation flow**

### Technical Marking

The Technical Marking process is not part of the Undo flow. It is used to mark a population for Undo. Since the cycle management GUI enables marking for Undo, the entire cycle or a specific customer/billing arrangement can be marked. As long as this process has not been run, the population for Undo can be changed. When complex selection criteria are required, IT professionals can prepare a list of the population for Undo. Using this process, the relevant cycle population is marked.

Each Undo request can be assigned an undo reason, which can then be used as one of the rerun criteria.

### Undo Initiator

The Undo Initiator process divides the population marked for Undo into groups for parallel processing. The size of these groups is based on definitions

set in Billing Configurator, and may be different from the group definitions for Bill Preparation or Bill Extracts.

## Undo Billing Products

The Undo Billing Products process works in two phases:

- Service Payer phase – Deletes document and/or invoice results from the database. For "Invoicing" and "Re-bill" Undo types, the process marks all the customers who contributed re-billed charges to the Service Payer bill. In addition, for "Invoicing" Undo types, it deletes the Bill Added Components entity that was created during the re-bill flow.
- Service Receiver phase – Deletes charges from the database.

When the entity is undone, it is considered as manually rejected, and the error description contains the Undo reason.

## Prepare Rerate Population File

If the billing administrator requests Undo with Rerate, this process moves over all the customers in the group, and extracts to the Rerate flow a list of customers that need to be rerated.

## Rerate Flow

The Rerate Flow process is an event processing flow activated by Billing for the population requiring rerating. When rerating of the given population is complete, Billing receives an indication (via a file), and the Undo flow is then completed (i.e., finalized).

## Undo Finalization

The Undo Finalization process validates that all groups initiated for Undo are complete, and that the Rerate map has finished. Finalization then updates the cycle status to Partially Processed.

## Rerun Flow

When the cycle population is processed partially (due to reject or undo reasons), the Rerun process is activated via AMC. The first step is to define the population for rerun, and the second step is to determine the Rerun flow.

### Define Criteria-based Rerun

The user can select the rerun population based on any of the following criteria:

- Cycle – The entire population of the cycle that is not completed
- Group – The rerun of a specific group which was rejected
- Error Code – The rerun of a rejected population with this error code
- Entity Type – Customer or BA
- Entity ID – Specific Customer or BA to be rerun
- Process ID – Rerun all the population that was rejected by the specified process ID
- Undo Reason Type – Rerun the population with the specific Undo reason

### Determine Rerun Flow

The user can determine which part of the Rerun flow will be performed on a given rerun population. The following options are available:

- Bill Preparation flow only
- Bill Layout Extract flow only

*The Bill Layout Extract flow does not require the definition of rerun criteria. The process extracts the entire population that Bill Preparation processed, and that did not complete Bill Layout Extract.*

- Full flow (i.e., both Bill Preparation and Bill Layout Extract)

## Billing Confirmation Flow

The Billing Confirmation flow is activated by request when bills are ready to be sent to customers. This flow confirms all completed and billed customers and extracted billing arrangements that are not marked for Undo.

*Billing arrangements marked to not produce a bill can be confirmed even though no Bill Layout extract was performed.*

Figure 3-7 displays the Financial Extracts flow.

*Confirmed billing products are final, and are not subject to the Undo process.*

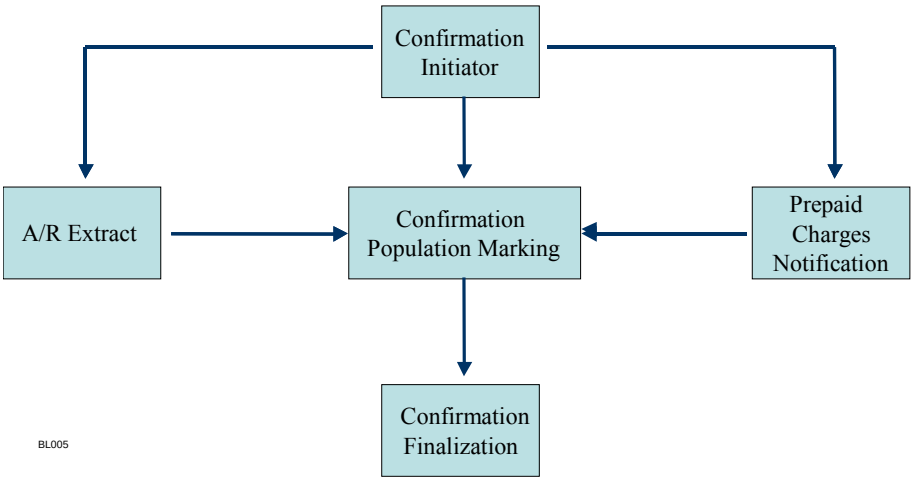


**Figure 3-7: Financial Extracts flow**

### Confirmation Initiator

The Confirmation Initiator process prepares the processed and extracted population that is not marked for Undo, and divides it into groups for parallel processing. The number of these groups is defined in Billing Configurator, and can be adjusted to the capabilities of the Accounts Receivable systems that receive the confirmation extracts.

Confirmation Initiator creates two files:

- Customer files – For Prepaid Charge extracts and confirmation marking
- Customer billing arrangement files – For Accounts Receivable extracts and confirmation marking

## Extract to Accounts Receivable

The confirmed billing products are reported to Accounts Receivable. The extract definitions for Accounts Receivable are defined in Billing Configurator (during implementation). The current implementation, for working with Amdocs Accounts Receivable, uses the following extracts:

- Extract of All Invoiced Charges – With original currency and amount, account currency and amount, charge code, etc.
- Extract Statements – With invoice date, due day, and amount due.


*If the Accounts Receivable Extract fails, the entity is not confirmed.*

## Prepaid Charges Notification

Charges for prepaid pay channels created during Bill Preparation (discounts, recurring charges) are reported to Replenishment Management through the TRB. Each credit charge publishes an increased prepaid balance transaction, and each debit charge publishes a reduced prepaid balance transaction.


*If Prepaid Charges Notification fails, the entity is not confirmed.*

## Confirmation Population Marking

This process marks the selected and extracted population as confirmed, disabling them from being marked for Undo, and enabling the next cycle to process them when the time comes.

## Confirmation Finalization

This process updates the cycle control status when the flow is completed for that cycle population. The status is set to *confirm* if the entire cycle population is confirmed, or to *partially processed*. Finalization is also responsible for publishing a cycle change confirmation to CM for confirmed customers who have requested a cycle change.

# 4. BILL ON DEMAND

Billing on demand (BOD) is the production of specific bills or invoices upon request, outside the context of the customer's regular billing cycle. This chapter provides the following:

- *Demand types.* A detailed description of the BOD type, which is configurable according to a variety of options.
- *BOD Flow.* A description of the BOD flow, including diagrams, lists of the basic steps, and detailed descriptions of each of the main processes or process groups included in the flow.
- *Priorities.* A brief explanation of how priorities are used to facilitate the timely processing of a BOD.
- *Reject and error handling.* A brief explanation of the ways in which reject and error handling for BODs differs from regular reject and error handling.
- *Limitations.* A list of the current limitations to BOD functionality.

## BOD Demand Types

A variety of BOD options are available by setting the attributes of the demand type. In addition to several demand types provided out-of-the-box, new demand types can be created and configured, using the attributes detailed in the table below:

| Attribute | Description |
|---|---|
| Final Indicator | Indicates whether it's a final BOD (Y = Final). |
| RC | Y – RCs are included in BOD (for final BODs only).<br>N – RCs are not included in the BOD. |
| OC | Y – All OCs are included in BOD.<br>N – No OCs are included in the BOD.<br>S – OCs are limited to a specific list. |
| UC | Indicates whether UCs are included in the BOD (Y = Yes). |
| DE | Indicates whether discounts are calculated for the BOD (Y = Yes). |
| Run Mode | Modes need to be configured to support the needs of the various revenue types. |
| Urgency | Immediate – BOD is run immediately (separate run).<br>Expedite – BOD is run in the next regularly scheduled cycle. |
| Specific Subscriber | Indicates whether the BOD is limited to a specific subscriber or includes all the subscribers of the selected BA. |
| Auto Confirm | If Yes, confirmation is triggered automatically. Usually set to Yes when Urgency is set to Immediate. |

The following table lists the demand types (and their attributes) provided out-of-the-box.

| Demand type | Final Ind | RC | OC | UC | DE | Run mode | Urgency | Specific Sub. | Auto Conf. |
|---|---|---|---|---|---|---|---|---|---|
| FINAL | Y | Y | Y | Y | Y | BODFNL | IMM | N | Y |
| SPNLMT | N | N | Y | Y | N | BODSOU | EXP | Y | N |
| SPCFOC | N | N | S | N | N | BODSOC | IMM | N | Y |

# BOD Flow

This section outlines the Bill on Demand (BOD) process flow.

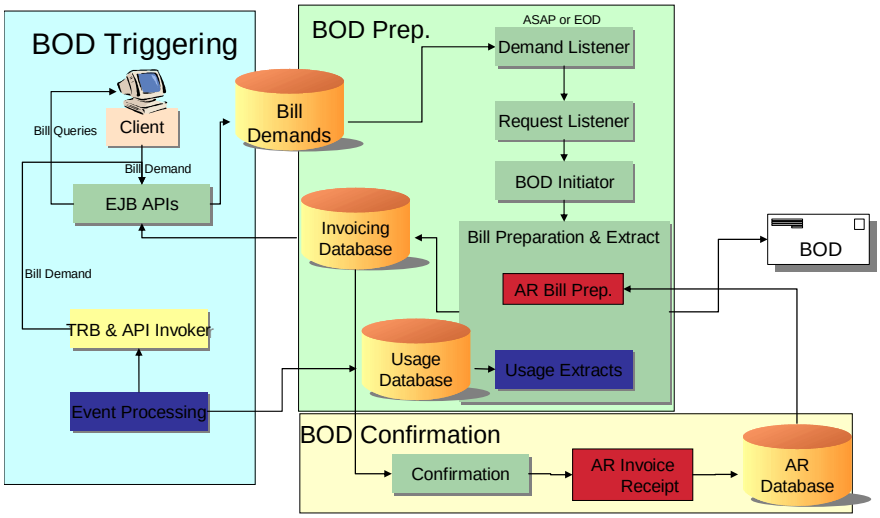


**Figure 4-1: BOD High Level Architecture**

The flow can be broken down into several sections, each of which is described in detail in the following sub-sections:

- *BOD Triggering.* The processes involved in creating the demand request and invoking the BOD Initiator.
- *BOD Preparation.* The processes beginning with the BOD Initiator and ending with the Finalizer (similar to the regular bill preparation flow).
- *BOD Post-Finalizer (Confirmation).* The Bill Layout Extracts, QA, Confirmation, and Undo processes.

## BOD Triggering Flow

This section details the processes involved in creating the demand request and invoking the BOD Initiator.

BOD requests can be triggered in either of two ways:

- *Synchronously.* The BOD APIs are activated by the operator from a GUI application.
- *Asynchronously.* The BOD APIs are activated using the TRB/API Invoker mechanism.

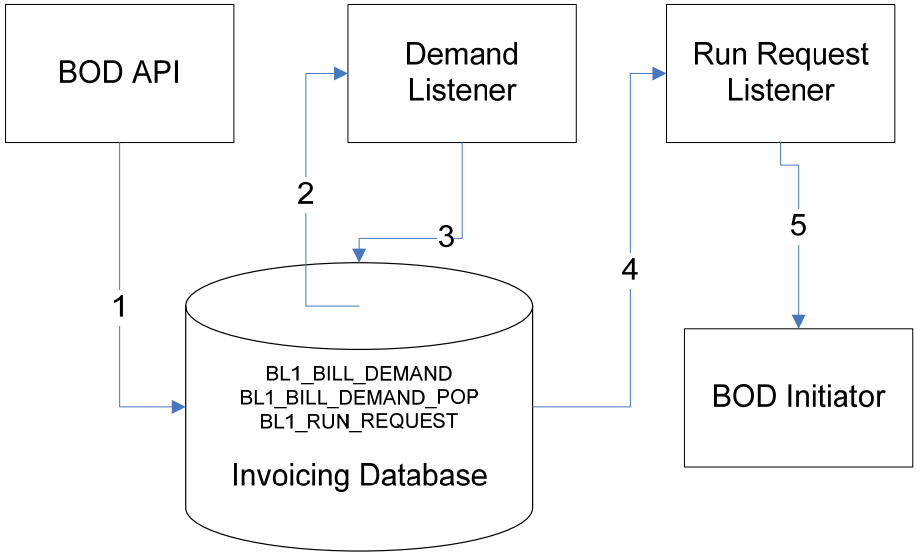


**Figure 4-2: BOD Triggering**

BOD triggering consists of the following basic steps (refer to the above diagram):

1. A BOD is requested using the BOD API. Each BOD request creates a new entry in the Demand Population table.
2. The Demand Listener daemon monitors the BOD requests.
3. For each selected group of BODs, the Demand Listener creates a run request in the Run Request table.
4. The Run Request listener monitors the database for run requests.
5. The Run Request listener launches the BOD initiator for each such request.

The following sections describe the Run Request Listener and BOD Initiator processes.

## Run Request Listener

The Run Request Listener monitors the Run Request table for BOD requests. For each BOD request in the table, it performs the following:

1. Activates the BOD Initiator for each newly arrived request. A single BOD Initiator run will handle all the BOD requests.
2. Updates the status to “Expedite” in the EOD control table (used by the Demand Listener) when triggering the Bill Preparation Initiator.
3. Launches the BOD Confirmation Initiator for all unconfirmed demands, once the regular Confirmation Initiator has been launched.
4. Launches the BOD Undo Initiator once the regular Undo Initiator has been launched. It checks whether a rejected BOD population is marked for undo in the Undo Tech Mark table, and runs the BOD Mark for Undo and Undo Initiator processes, if necessary.

## BOD Initiator

For each BOD request, a dedicated BOD initiator is launched. It is responsible for fetching the BOD preparation run population, and dividing it into groups to enable parallel processing.

Each group created by a BOD Initiator contains BOD requests having a common cycle sequence and demand type (as set in the Bill Demand Type table).

For each group created, the Initiator does the following:

1. Identifies the population according to the Request ID. Both the BA and customer populations are taken from the Bill Demand table. For Final BOD, only the payer customer and its related BAs are taken.
2. Validates the population, as follows:
   - Checks that the BA exists in the Billing database and that its status is "Open."
   - Checks the Cycle Population tables to ensure that neither the BA nor any of its related customers are rejected.
   - Checks BA activities versus cycle status, according to the type of run (regular or previous BOD). See the diagrams below for details. BOD requests submitted during the period shown in red are considered invalid and are rejected.

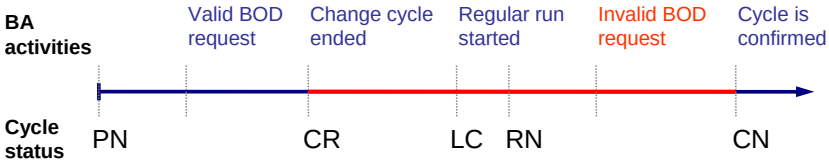


**Figure 4-3: Regular Run: BA Activities vs. Cycle Status**

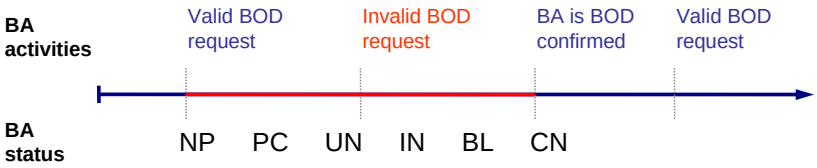


**Figure 4-4: Previous BOD Run: BA Activities vs. Cycle Status**

   - If a subscriber list exists, the Initiator checks that the subscriber's customer exists in the Invoicing database and that its status is "Open."
   - Checks that the next available cycle instance for the BA is closed (i.e., the cycle is ready for regular run).
   - Checks the Rejected Transactions table to ensure that the customer has not been rejected.
   - When run in Expedite mode, checks that the customer has no open requests for backdate handling.
3. Splits the population into groups. All the valid requests with the same cycle sequence and demand type are read together as a single population.

The grouping engine then divides it into groups according to the Bill day initiator's algorithm. For final BODs, all Customer BAs are automatically processed in the same group.

4. Writes the groups to the appropriate cycle tables and output files. Cycle sequence run number (in the cycle tables) is initiated to -1 for the first BOD run of the cycle. Each BOD run for the same cycle will decrease the cycle sequence run. For expedite mode it is decreased to the next odd value, while for immediate mode it is given an even value. The negative value of the cycle sequence run makes it easier to distinguish the BOD population from regular run populations. The cycle sequence run number is also saved in the Cycle Control table.
5. Marks the BOD requests as handled.

## BOD Preparation Flow

This section details the processes responsible for bill preparation, beginning with the BOD Initiator and ending with the Finalizer. While the flow is essentially the same as the regular bill preparation flow, there are some significant differences. Specifically, there are special run modes defined for the various demand types.

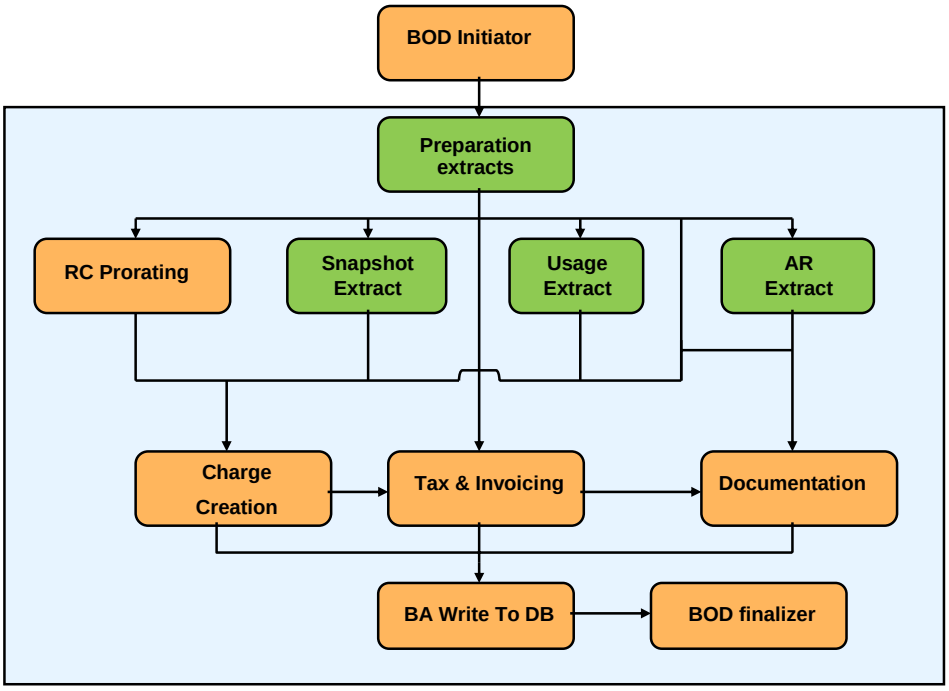


**Figure 4-5: BOD Preparation**

BOD preparation consists of the following basic steps:

1. The BOD Initiator identifies the selected population, validates it, and splits it into groups, where each group has a common cycle sequence and route, and will populate the cycle tables.

2. The Preparation Extracts phase filters the charges relevant to the selected population: by BA, specific subscriber or specific OC. This filtering guarantees that only the charges related to the BOD population will be processed during the current BOD run.
3. The Charge Creation process creates a compensation charge for each charge processed by the BOD flow (to offset the BOD charges in the next bill cycle – see Charge Creation, below).
4. The Tax and Invoicing process calculates tax, creates accumulations, and creates invoices and prepaid statements. No changes are needed for BOD.
5. The Documentation process incorporates the compensation charges from the previous run, and includes them in the total amount due calculation. See the compensation charges example, below.
6. The BOD Finalizer (in the Bill Preparation flow) monitors the BOD groups. A BOD group that finishes the Bill Preparation flow continues on to the extract flow. Once all BOD groups have completed the Bill Preparation flow (including the layout extracts), the BOD Finalizer triggers the QA, Undo, and Confirmation flows.

The following sections detail the BOD preparation processes, and are followed by a compensation charges example. Note that the BOD Initiator is detailed above, in the section on BOD Triggering.

### BOD Preparation Extracts

The BOD Preparation Extracts include the following extract processes:

- *Pay Channel Extract.* The same extract as in the Bill Preparation flow, it is included in all BOD run modes.
- *Payer Pending Charges Extract.* In BOD preparation, this extract is extended to support the Specific Subscriber option. When this option is selected, all subscribers that distribute their events to the BA (including cross-customer distribution) are included in the extract. The Payer Pending Charges Extract is excluded from the Specific OC BOD run mode.
- *Population Extract.* This process extracts the customer and BA lists for processes relevant to BOD (i.e., Snapshot, Usage, AR Extract, and Tax & Invoicing processes), and generates the following types of files for each group:
  - Customer – Lists all the customers in the group.
  - Customer for PI extract – Lists all the customers for PI extract, including the BA number. If the Specific Subscriber option is selected, then the subscriber ID is also extracted. However, only subscribers of the given BA are included (i.e., no cross-customer distribution) – no cross-distributed PIs are fetched from the Rater.
  - Customer billing arrangement – Lists all the billing arrangements related to the customers in the group, including printing categories captured by the BOD request, cycle sequence number, cycle sequence run, run type, and cut-off date.

  The Population Extract is included in all BOD run modes.

- *Charge Request OC Extract.* This process extracts one-time charges (OCs) created for the customer (Final BOD), the specific BA (Specific BA BOD), the subscriber (Specific Subscriber BOD), or the specific OC. Note that OCs subject to cross-customer distribution are ignored. The Charge Request OC Extract is not run when the demand type indicates that OCs are not included.
- *Customer Information Extract.* The same extract as in the Bill Preparation flow, it is included in all BOD run modes.
- *RC Rates Extract.* The same extract as in the Bill Preparation flow, it is included in the Final BOD run mode only.

### Snapshot Extract

The same extract as in the Bill Preparation flow, it is excluded from BOD run modes that do not calculate discounts.

### Usage Extract

This is the same as in the Bill Preparation flow, except for the following:

- Adds BA number to the extract.
- Adds subscriber number to the extract when Specific Subscriber option is selected.

### Accounts Receivable (AR) Extract

This is the same as in the Bill Preparation flow, except for the following:

- Late payment fees are received for final BOD only.

### RC Proration

The RC Proration process is included in the Final BOD run mode only.

For RCs with generation frequency greater than one, the output file is not written to the database. This is to ensure that the RC rates are fetched again from the database in the next run (i.e., for the revised final bill).

### Charge Creation

When creating a BOD, the Charge Creation process performs the following:

- Enables diverse run modes for creating charges (e.g., OC only, UC only, or Final BOD, which includes all charges).
- Creates a compensation charge for each charge processed by the BOD flow. Note that:
  - The value of the compensation charge is equal to the opposite of the processed charge. Since the next bill (either regularly scheduled or BOD) will include all charges for the cycle (until that point), the compensation charges are used to offset the portion already billed in the BOD.
  - No compensation charges are created for late payment fees, pending charges, or newly created cross-customer distribution charges during the BOD run.
  - During the next run, the Pending Charges Extract fetches the compensation charges as pending charges.

- Discounts can be calculated or deferred according to the demand type and configuration settings.

### Tax and Invoicing

When creating a BOD, the Tax and Invoicing process updates the invoice type on the created bill or invoice with one of the following BOD invoice types:

- BOD run with compensation charges
- BOD invoice
- BOD with rebill charges
- BOD run with rebill and compensation charges

### Documentation

When creating a BOD, the Documentation process updates the BOD printing category on the document, as well as the document demand type.

### BA Write To DB

When creating a BOD, this process performs the following:

- For non-final BOD run: Updates the customer's DB status to Completed, assuming that the status of at least one of its BAs is updated to Billed in the same BOD run. If, on the other hand, all of the customer's BAs are rejected, then the BA Write to DB process does not update the customer's status to Completed.
- For final BOD run: If any of the BAs is rejected, the process rejects all the customer's BAs and rolls back all its products.
- Rejects any BA that is rejected in the Cycle Payer Population table, or does not exist in the current cycle sequence run in the Cycle Payer Population table.

### BOD Finalizer

The BOD Finalizer completes the BOD flow for the incoming BOD run requests. It sends files to the Bill Layout Extracts (same as in the regular Bill Preparation Finalizer).

Note that:

- When a new file arrives at the BOD Finalizer, this indicates that the file's related group has completed the BA Write to DB process.
- More than one request can arrive while the BOD Finalizer is running.
- A single BOD flow can handle populations belonging to different cycles. The status of each cycle group is updated to Finished (but unconfirmed).

### Compensation Charges Example

The following is an example of how compensation charges are used to offset the charges already billed in the BOD.

Assume the following BOD run results:

| Invoice | Revenue type / orig. | Amount | Tax |
|---|---|---|---|
| BOD Invoice | RC | 50 | 5 |
| | OC | 10 | 1 |
| | UC | 100 | 10 |
| | LPF | 20 | 2 |
| BOD Invoice Total | | 180 | 18 |

Now assume the following results for the next regular run:

| Invoice | Revenue type / orig. | Amount | Tax |
|---|---|---|---|
| Regular Invoice | RC | 90 | 9 |
| | OC | 20 | 2 |
| | UC | 210 | 21 |
| | LPF | 10 | 1 |
| Compensation Charges | RC | -50 | -5 |
| | OC | -10 | -1 |
| | UC | -100 | -10 |
| Regular Invoice Statement Total (Regular Run + BOD compensation charges) | | 170 | 17 |

## BOD Post-Finalizer Flows

This section describes the flows run following the BOD Finalizer (Bill Layout Extracts, QA, Confirmation, and Undo).

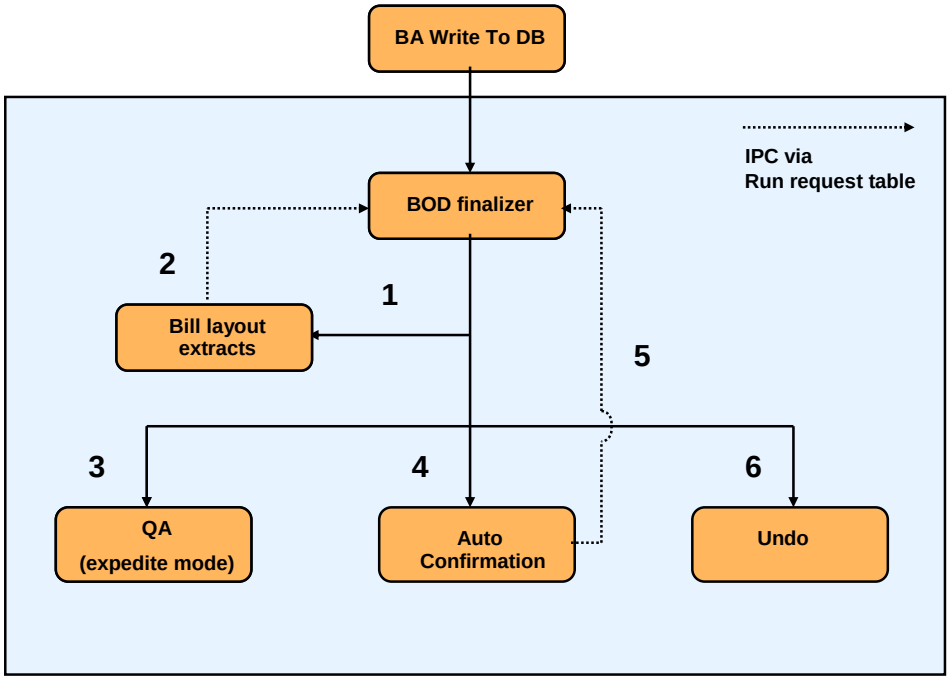


Figure 4-6: BOD Finalizer and Post-Finalizer Processes

The post-finalizer processes consist of the following basic steps:

1. Bill Layout Extracts – The BOD Finalizer invokes the Bill Layout Extracts process (same logic as in the regular Bill Preparation Finalizer). The BOD flow differs from the regular flow as follows:
   - BOD routes are assigned a higher priority in the Bill Layout Extracts flow.
   - Non-UC routes exclude the event and non-charge PI extracts from the Rater.
   - The document type field is extracted in Billing Generic Extract (application data extract).
   - BOD bill layout format is extracted according to the BOD Bill Format BE policy.
   - If call details are not required for the bill on demand (as set by the BOD Call Details BE policy), then the event extract is disabled in the Rater.
   - All BOD requests are extracted in the Billing Generic Extract process.
2. Once the Bill Layout Extract is complete for all groups of a single BOD request, the BOD Finalizer performs the following:
   - When in Expedite mode (i.e., the Urgency attribute of the BOD Demand Type is set to Expedite), it invokes the BA and Customer QA Initiators, according to the Auto Run BOD QA policy (see step 3, below).

- It invokes the Auto Confirmation process (see step 4, below), regardless of Urgency (Immediate or Expedite).
- If there are BOD undo requests, the BOD Finalizer invokes the Undo Mark and Initiator processes (see step 5, below), regardless of Urgency.

3. QA – This step depends on the Urgency attribute of the BOD Demand Type:
   - Immediate: QA flow is disabled.
   - Expedite: Regular QA is automatically triggered by the BOD Finalizer or is invoked by the operator, depending on the Auto Run BOD QA policy.
4. Confirmation – This step depends on the Auto Confirm indicator (of the BOD Demand Type):
   - Yes: The BOD is confirmed immediately (e.g., for "customer in store" scenario). The confirmation flow is automatically triggered by the BOD Finalizer.
   - No: The BOD undergoes a regular confirmation process (if Urgency is set to Expedite).
5. Undo – Rejected BODs are automatically marked for undo, and the Undo flow (Undo Mark and Undo Initiator) is automatically triggered by the BOD Finalizer. Thus, for rejected requests, no billing products are saved in the Billing database. Only immediate demands are undone for expedited requests.

The following sections detail the Bill Layout Extract, QA, Confirmation, and Undo flows.

### BOD Bill Layout Extracts

The Bill Layout Extracts are responsible for structuring the data sent to the Bill Formatter. It extracts information from several areas, (such as Customer Management, Accounts Receivable, Rater processes and Invoicing).

The BOD flow differs from the regular Bill Layout Extracts flow in the following ways:

- Non-UC routes are excluded from the Event Population extract and the non-Charge PI extract.
- BOD routes in the Bill Layout Extracts flow are given a higher priority than the regular run routes.
- Extract Reference Data: Add BOD layout format extract.
- Generic Extract:
  - Adds the document demand type to the Generic (Non-Reference Data) Extracts.
  - Adds BOD Request Extract.
- Usage Extract:
  - Adds BA number to the extract.

- Adds subscriber number to the extract when Specific Subscriber option is selected.
- If no call details are required for the BOD, then the usage extract is disabled.

### QA Flow

The QA flow is disabled for Immediate BOD requests. However, for Expedite requests, regular QA is automatically triggered by the BOD Finalizer, as determined by the Auto Run BOD QA policy.

### Confirmation Flow

The BOD Confirmation flow differs from the regular Confirmation flow in the following ways:

- The statuses of the BOD entities (i.e., Customer, BA) are updated to "BOD confirmed".
- The status of the BOD request is updated to "Confirmed".
- The confirmation mechanism is determined by the Auto Confirm indicator (of the BOD demand type):
  - Yes: The BOD is confirmed immediately, but only those entities marked for automatic confirmation are confirmed.
  - No: The BOD is confirmed normally as part of the cycle's regular confirmation process (so long as Urgency is set to Expedite). However, only those BOD entities NOT marked for automatic confirmation are processed.
- A BOD route is added to the Confirmation flow and is given a higher priority than the regular run routes.
- The only bill that can be issued after a final BOD is a revised final bill. If a BA is cancelled, the type of final bill is determined by the Bill Day Initiator:
  - If the BA's last production date is before the BA's status date, a final bill is produced.
  - Otherwise, a revised final bill will be produced.
- Payment method can be overridden (for example, to allow cash payment to bypass automatic direct debit), depending on the Enable Direct Debit indicator on the BOD demand:
  - Yes: Payment method is determined by the payment method of the pay channel.
  - No: Override is required (i.e., direct debit is activated).
- BA last production date is updated with the run date in all BOD groups.
- The Invoice Statement extract to Accounts Receivable is enhanced to include the following information:
  - Cycle sequence
  - Cycle sequence run
  - Cut-off date

- Bill run type – “Regular” or the BOD demand type

### Undo Flow

The BOD Undo flow differs from the regular Undo flow in the following ways:

- BODs rejected either during BOD processing or using the Cancel Bill Demand API are marked for Undo.
- Undo can either be invoked immediately (for customer in store scenario) or can be deferred and carried out normally (for expedite scenarios). The undo mechanism is determined by the urgency, as set on the BOD demand type:
  - Immediate: Undo is launched automatically by the BOD Finalizer. Undo erases only those BOD entities related to an immediate request.
  - Expedite: A regular undo is performed. All BOD entities whose Urgency is set to Expedite are processed. (Note that more than one request may be included in the population.)
- All the BOD preparation products are erased from the database. Undo is performed on either the BA or customer level, according to the BOD demand type. Final BODs are performed on the customer level, while non-final BODs are performed on the BA level.
- BOD routes in the Undo flow have a higher priority than the regular run routes.
- Undo marks the undone demands as “Canceled”.

## BOD Priority

The Invoicing flow mode mechanism enables the defining of a priority for each run mode. When an Invoicing process is free to handle the next group, groups with higher priority are preferred. This mechanism is used to allow a BOD to run during the processing of a regular bill day run, without waiting for the latter to finish.

## Reject and Error Handling

BOD reject handling differs from regular reject handling in the following ways:

- There is no real solution for Immediate BODs that are rejected, since BOD rejects are handled by back-office procedures. The CSR will have to apologize and promise that the bill will be sent as soon as the “technical problems” are resolved.
- When a BOD is rejected, the reject indicator of the rejected entity is updated in the cycle population tables. If the reject occurred after the BA Write to DB process (as indicated by the DB status), it is marked for undo as well.
- When a BOD is rejected, the bill is deleted by undo and the BOD request is “Canceled”. There is no re-run for BODs; a new BOD request is needed in order to run it again.

- When a BOD is rejected, a new BOD cannot be initiated until the rejected demand is canceled. In the meantime, any new BOD requests are rejected.

# 5. BILLING EOD PROCESS

This chapter describes the Billing End of Day (EOD) processes. Billing EOD performs the following major activities:

- Rerating and backdating initiated by PC rate changes or version fixes, whereby the RCs or OCs change due to PC changes
- Backdating initiated by CM activities, whereby the RCs change due to a CM activity
- Change cycle initiated by CM activities

Backdating means performing a task by CM with an effective date in the past (i.e., prior to the issue date). Billing 6.0 enables backdating to dates in the open cycle and prior to the open cycle. The following activities are supported:

- Backdating initiated by either CM activities or PC rate changes or version fixes (i.e., RCs due to CM activities; RCs or OCs due to PC changes).
- Rerating of backdated events due to potential interactions with other rates. Thus, backdating is supported even for the events with subsequent changes (i.e., where status, agreement parameter, or PC rate changed after the effective date but prior to the issue date).

## Billing End of Day Flow

The Billing EOD flow is activated daily. This flow performs the following:

- Billing EOD Listener – Triggers the BTL-SOR daemon shutdown
- Handle PC Changes – Rerating of RC and OC due to PC changes
- Handle Backdating – Replay activity for backdated activities
- Handle Change Cycle activities
- Cycle Maintenance – Opens future cycle instances as configured in the system

Figure 4-1 illustrates the Billing EOD process flow.

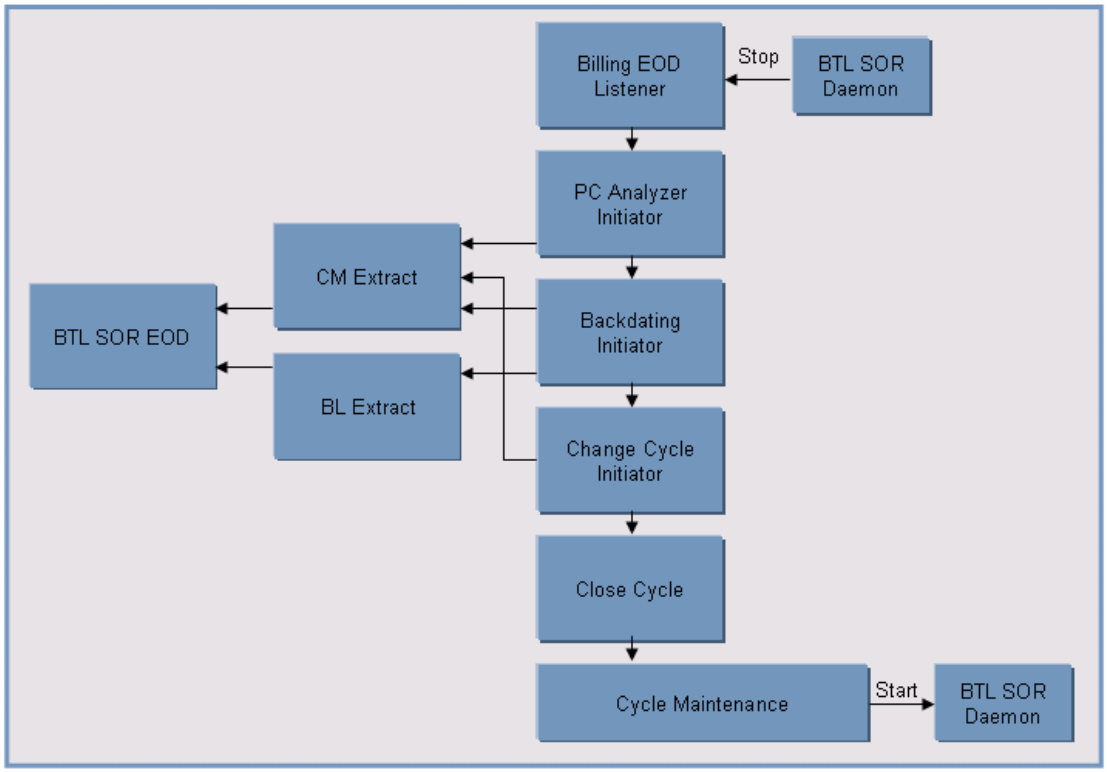


**Figure 4-1: Billing EOD process flow**

## Billing EOD Initiator

The first step in the Billing EOD flow is to trigger the BTL-SOR daemon to shut down. From this point on, any new CM activities will be considered next day activities.

## PC Change Handling

The first step is to define population for backdating initiated by PC rate changes or version fixes, and then to generate the new RC charge request while expiring the old RC.

The process is composed of the following three sub-processes.

### PC Analyzer Initiator

This sub-process processes the PC change rate transactions of the offers and pricing packages (e.g., offer rate change, package rate change, and backdates), and groups them by version and cycle code.

### CM Extract

The CM Extract process can be configured to extract the relevant CM information required for rerating due to PC changes. When invoked by the PC Analyzer Initiator file's rate change transactions, the process extracts the corresponding subscriber information by offer, cycle code and offer effective date, pay channel, and other information required for rerating of the RC.

### BTL-SOR EOD

Configured to run in the PC Analyzer route mode, BTL-SOR-EOD handles only Product Catalog transactions. BTL-SOR EOD processes the PC rate change transactions, and updates the RC in the RC Rates table and the OC in the Charge Request table. If backdating is required, a backdating request is entered into the Backdated Request table, and is treated later on, in the Backdating flow.

When BTL-SOR EOD processes a backdated Product Catalog version, it checks if a change cycle transaction was received for the affected subscribers after the effective date of the Product Catalog version release. If such a transaction was received, BTL-SOR EOD marks the subscribers for the backdating process with the Rate Effective Date of the old cycle.

## Backdating Handling

The first step is to identify the population for backdating by extracting the relevant entries from the Backdate Request table. Then, the newly calculated RC and OC charge requests are generated while expiring the old ones.

The process is composed of the four sub-processes described in subsequent sections.

### Backdating Initiator

This sub-process identifies the population for backdating (from PC Analyzer and daily BTL-SOR transactions) by extracting entries from the Backdate Request table. The process then calculates the starting date of the rerate and groups the subscribers for parallel file processing. The number of groups, and number of records per group, are based on the applied configuration.

### CM Extract

The CM Extract process can be configured to extract the relevant CM information required for rerating due to backdating. When invoked by Backdating Analyzer, the process retrieves additional subscriber information (e.g., services, agreement parameters, pay channel, cycle code, currency, etc.) required for rerating.

### BL Extract

The BL Extract process is required for backdating. This process extracts the history of financial transaction activities to be used for backdating recalculation.

### BTL-SOR EOD

Configured to run in the Backdating route mode, the BTL-SOR EOD process rerates the RC and OC charges for the subscriber population selected for backdating from the Backdated Request table.

For each subscriber, the process identifies all the subscriber's activities for the relevant period in the Activity History table. This encompasses all the activities from the earliest cycle involved in the backdating to the open cycle.

For each activity, rating is invoked, resulting in re-calculation of OC and RC. The subscriber's RC is expired for the backdated cycle start date.  OC charges

are deleted if their origin is “charge preparation”. OC charges received from the Create Charge API or from the Commerce Charges files remain untouched. If the CSR made corrections to an OC charge with the origin “charge preparation”, all the related charges (original and all corrections) are not deleted and not re-calculated.

If BTL-SOR EOD encounters a transaction that is backdated to a date earlier than the start date of the open cycle, the customer (service receiver) is marked for re-billing.

## Cycle Change Handling

The first step is to identify the population marked for a cycle change. Then, the newly calculated RC charge request is generated while expiring the old one.

The process is composed of three sub-processes described in subsequent sections.

### Change Cycle Initiator

This sub-process extracts the cycle change population from the Customer table (with an expiration date for the old cycle).

### CM Extract

When invoked by Change Cycle Initiator, the process retrieves additional subscriber information required for rerating of the RC.

### BTL-SOR EOD

Configured to run in the Change Cycle route mode, BTL-SOR EOD rerates and generates the newly calculated RC charge request while expiring the old RC charge request.

In addition, the customer change cycle activity is recorded in the Activity History table for re-use in cases where backdating is to a date before the change cycle date.

## Close Cycle

This process updates the cycle status of the old cycle in the Cycle Control table to “closed”, so that Bill Preparation can run on the cycle.

## Start BTL SOR Daemon

The final step in the Billing EOD flow is to trigger the BTL-SOR daemon start-up with the new logical date.

# 6. CUSTOMER ENTITIES

This chapter describes the customer entities that are relevant for Amdocs Billing, and their life cycles.

## Customer Entity Life Cycle

The following major customer entities, which reside in Customer Management, are utilized by Billing:

- Customer
- Billing Arrangement
- Pay Channel
- Subscriber
- Unit

Amdocs Billing maintains a full life cycle for the following entities:

- Billing Arrangement – The document receiver and main billing entity
- Pay Channel – The service payer
- Customer Service Payer Relation – The entity that holds the cross-customer distribution information
- Customer – The main entity
- Customer Information – This entity is used by bill day run (the tax module, charge accumulation group mechanism, and FYI messages)

Billing maintains these entities by subscribing to Customer Management activities.

### Billing Arrangement Life Cycle

Billing subscribes to the following transactions associated with the billing arrangement life cycle:

- New Billing Arrangement – A new, tentative billing arrangement exists in the system. It is not active yet, and Billing cannot produce a bill for this billing arrangement.
- Open Billing Arrangement – A new billing arrangement is active in the system. Billing changes the billing arrangement status to Open, and this Billing Arrangement is processed in its next bill cycle.
- Cancel Billing Arrangement – No active service receivers are distributing their charges to the billing arrangement. Billing should produce a final bill for this billing arrangement.
- Update Billing Arrangement External ID – This transaction informs Billing that a billing arrangement's external ID has been modified.

- Update Billing Arrangement Dates – This transaction informs Billing that a billing arrangement's dates have been modified. Typically, this action is a result of backdated activities performed on the billing arrangement.
- Change Account Management – This transaction informs Billing that the document type of a specific billing arrangement has been modified (from invoice to bill or *vice versa*).

Billing maintains the following major attributes for each billing arrangement:

- Status – Tentative, Open, Cancel, or Close
- Status Date – The date the current status was assigned
- External ID – The ID of the billing arrangement in an external system
- Last Production Date – The date on which the last billing document was successfully produced for the billing arrangement
- Document Type – Bill or Invoice Receiver
- Document Production Indicator – The bill information for the billing arrangement is not extracted to the bill layout utility. This information is determined by the customer type or set manually (using Customer Management APIs), and extracted from Customer Management. This may be used for a billing arrangement reflecting the communications service provider's (CSP's) employees or to test handset expenses, which are calculated but no bill is sent.
- Zero Balance Indicator – Indicates if the invoice should be set to zero by adding a debit or credit to the invoice total. Typically, the zero balance is used for internal billing arrangements. This indicator is determined by the customer type, or is set manually (using Customer Management APIs) and extracted from Customer Management. This may be used for billing arrangements of the CSP's employees, or for testing handset expenses, which are calculated but whose bill is not sent.


*Usually, the Zero Balance and Document Production indicators are set as a pair*

- Itemized Tax Indicator – Indicates whether to calculate the tax on each charge or on the total amount.
- Last Document ID – Identifies the last document produced for the billing arrangement.
- Document Format – The bill format (e.g., Individual or Business).
- Bill Due Days – The number of days to add to the invoice date to determine the due date for payment.
- Bill Frequency – The cycle frequency multiplier related to the billing arrangement.

## Pay Channel

Billing subscribes to the following transactions associated with the pay channel life cycle:

- New Pay Channel – Billing opens a new tentative pay channel with the following attributes:
  - Billing Arrangement – The billing arrangement to which the pay channel belongs
  - Customer – The customer to which the billing arrangement and pay channel belong
  - Account – The account to which the billing arrangement belongs
  - Payment Category – Indicates whether the pay channel is prepaid or postpaid
  - External ID – The ID of the pay channel in an external system
  - Payment Method – Indicates the method used for the pay channel payments, such as bank direct debit or credit card
  - Status – The pay channel status is set to Tentative
- Change Pay Channel – Billing modifies the Bill Due Days attribute of the pay channel's billing arrangement. with the new value
- Open Pay Channel – Billing makes the pay channel active in the system. The following Pay Channel attributes are updated:
  - Status – The pay channel status is set to Open (i.e., active)
  - Status Date – The date on which the current pay channel became active
  - Bill Due Days – If a value is provided, the number of days (to add to the invoice date to determine the due date) attribute is updated
- Update Pay Channel External ID – Billing updates the Pay Channel's External ID attribute.
- Cancel Pay Channel – Billing cancels the pay channel and updates the following attributes:
  - Status – The pay channel status is set to Cancelled
  - Status Date – The date on which the current pay channel was cancelled

Billing maintains the following major attributes for each pay channel:

- Status – Identifies the pay channel status (Tentative, Open, or Cancel)
- Status Date – The date of the pay channel's status update
- Payment Category – Indicates whether the pay channel is prepaid or postpaid
- External ID – An external identifier of the pay channel (this attribute is used when the pay channel is also represented in external systems)
- Payment Method – The method of payment (e.g., cash, credit card, or direct debit)

## Customer Service Payer Relation

This entity holds information of cross-customer distribution, and is derived from Customer Management charge and event distribution.

Billing subscribes to the following transactions associated with the payer relation life cycle:

- Add Cross Distribution:
  - Customer – The payer customer
  - Billing Arrangement – The billing arrangement to which the pay channel belongs
  - Pay Channel
  - Effective Date – The date the cross-customer distribution was added
  - Distributing Customer – The customer that added distribution to the given pay channel
- Remove Cross Distribution:
  - Pay Channel – The pay channel from which the cross-distribution was removed
  - Expiration Date – The expiration date of the cross-distribution (cross-customer distribution is canceled by expiring the customer service payer relation)
  - Distributing Customer – The customer that removed the distribution to the given pay channel

Billing maintains the following major attributes for each Customer Service Payer Relation:

- Customer – The payer
- Billing Arrangement – The billing arrangement to which the pay channel belongs
- Pay Channel – The pay channel to which the cross-customer distribution was made
- Distributing Customer – The customer distributing to this billing arrangement

## Customer

Billing subscribes to the following transactions associated with the Customer entity life cycle:

- New Customer – A new customer is added to the system. Billing adds the new customer to the Billing database. The following Customer attributes are updated:
  - Customer – The payer customer
  - Bill Cycle Number – The number of the cycle in which the bill must be prepared for all the billing arrangements of this customer
  - External ID – The ID of the pay channel in an external system
  - Bill Production Indicator – Indicates whether to produce a bill for this customer
  - Offer Currency – The currency of the customer
  - Business Entity ID – The business entity to which the customer relates

  - Status – The customer status is changed to "O" (Open), which means that the customer is active
  - Status Date – The date on which the customer has become active
- Change Bill Cycle – Change Billing Cycle request is initiated by Customer Management, with the following attributes:
  - Old cycle expiration date – The expiration date of the old billing cycle
  - New billing cycle code – The number of the new cycle in which the bill must be prepared for all the billing arrangements of this customer.
- Change Cycle Immediate –Applicable to customers without subscribers.
- Cancel Bill Cycle Change Request – Occurs when a Cancel Bill Cycle request for a specific customer is issued. The following customer entity attributes are modified:
  - New billing cycle code – The number of the cycle in which the bill must be prepared for all the billing arrangements of this customer. This attribute value is set to null.
  - Old Cycle Expiration Date – The old cycle's expiration date value is set to null.
- Subscriber Activation/Cancellation – The number of subscribers per customer is maintained on the billing Customer entity, to be used for bill day parallel group balancing. The Customer Weight attribute of the Customer entity is updated to reflect the new or cancelled subscriber (value increased for a newly activated subscriber, decreased for a cancelled subscriber).
- Subscriber Resume/Renew – Billing supports reactivation of a cancelled subscriber. This can be done in one of the following ways:
  - Resume – If reactivation occurs in the same cycle as the cancellation, the subscriber can be resumed. This activity is backdated to the date immediately following the cancellation. The intended effect is to make it (as much as possible) as though the cancellation never occurred (e.g., expiration dates are removed, and penalties due to cancellation are rescinded).
  - Renew – The renewed subscriber is treated as a new subscriber based on the cancelled subscriber information (including the old subscriber ID applied by CM). The renewal activity can be either immediate or backdated.

  Pay channels and billing arrangements closed due to cancellation can also be reactivated, provided that this occurs during the same cycle in which they were closed.
- Update Customer External ID – Billing updates the external identifier of the customer.
- Change Customer Type – The change of a customer type may trigger changes in the billing arrangement. The following billing arrangement attributes are updated:

- Document Type – Indicates whether the billing document is an invoice or a bill
- Itemized Tax Indicator – Indicates whether tax is calculated on each charge, or on the total amount of the bill
- Bill Format – The bill format, e.g., Individual or Business
- Zero Balance Indicator – Indicates whether to zero the balance of the billing arrangement at the billing run
- Bill Due Days – The number of days to add to the invoice date to calculate the due date
- Bill Production Indicator – Indicates if a document should be produced for the billing arrangement.

- Hold Bill Indicator – Billing provides an API to set the Hold Bill indicator for a customer. This indicator is set for customers that should be ignored during bill preparation. The cycle preparation ignores the subscribers and agreements that belong to customers with hold bill indications, and prevents charges from being created for these customers.

  This option is used to skip (postpone) a set of customers for various reasons while processing the rest of the cycle's population.

Billing maintains the following major attributes for each Customer:

- Customer.
- Status – Identifies the Customer's status (Open, Close).
- Status Date – The date the status was assigned.
- Billing Cycle Code – Cycle that processes the customer.
- New Billing Cycle Code – When a customer wishes to change its billing cycle, this attribute is populated with the new cycle code. The billing cycle code attribute is then populated with the new billing cycle during the confirmation of the old cycle.
- Old Billing Cycle Code – The previous cycle code, if the customer has changed billing cycle.
- Hold Indicator – Indicates whether Billing needs to handle the customer.
- Weight – The estimated impact on processing in the current cycle. This attribute reflects each subscriber or billing arrangement of the customer to be processed in the cycle.
- Business Entity – Identifies the business entity to which the customer is related.
- Customer Currency – The customer offer currency.

## Customer Information

The Customer Information table maintains the customer information that is loaded from Customer Management (CM) via the BTL daemon.

Billing maintains the following major attributes of the Customer Information entity:

- Customer

- Entity Type – The type of entity that holds the information (customer, billing arrangement, or subscriber)
- A list of dynamic attributes that are defined in Billing Configurator, and are mapped through the CM transaction XML files that are published on the TRB

### Subscriber and Organizational Units

Subscriber and organizational units are not maintained in Billing.

Subscriber and organizational unit information is extracted from Customer Management on bill day by the Billing End of Day flow.

## Billing Customer Entities

### Customer Attribute Definition

The implementation team can use Billing Configurator to define the customer attributes that are used in tax and accumulation rules. Billing Configurator includes the GUI Mapping tool that enables mapping of physical data names (such as CM attribute names) to logical names defined in Billing Configurator for the CM Extract process.

The Customer Management extract then populates values for these attributes.

Each attribute has the following elements:

- Attribute Name
- Attribute Type – String or Integer
- Attribute Entity Origin – Customer, billing arrangement, or subscriber

The attribute name is not necessarily the same as in the Customer Management database. However, the implementation team should define the Customer Management extract in such a way that the corresponding value from Customer Management populates the required attribute.

Examples for such attributes are listed in the following table:

| Attribute Name | Attribute Type | Entity Origin |
|---|---|---|
| subscriber_state | String | Subscriber |
| Customer_geo_zone | String | Billing Arrangement |
| Customer_type | String | Customer |

Each attribute name is unique. If the same attribute is extracted from a subscriber and billing arrangement, then a different attribute name is given to each (e.g., customer_state and billing_arrangement_state).

### Using Customer Attributes

When using customer attributes to define tax or accumulation rules, the implementation team must define such rules for each attribute, including:

- Attribute Name – Name of one of the attributes defined as a customer attribute
- Origin – Service Receiver or Payer

- Payer – The attribute entity origin should be billing arrangement or customer
- Service Receiver – The attribute entity origin should be subscriber or customer

When Billing uses customer attributes, it should know whether the defined attribute refers to a payer or service receiver entity, e.g., the tax may depend on the state of the service receiver (subscriber state), or payer (billing arrangement state), depending on the local rules.

# 7. FLEXIBLE BILLING CYCLES

Billing is not limited to monthly cycles. Cycle frequencies, as well as bill production frequencies, can be any multiple of a week or month, and several cycles may feed into a single bill.

This chapter describes billing cycles, and explains how they are managed. It includes a list of cycle statuses, as well as an explanation of how cycles are controlled and how they are changed for a given customer entity.

## Cycle and Bill Production Frequencies

### Flexible Cycle Frequencies

While cycles are assigned to customers in Customer Management, they are defined using Billing Configurator. A flexible bill cycle can be defined by a combination of two attributes:

- Cycle Frequency – Weekly or monthly
- Cycle Multiplier – Any positive number

The following cycle periods can be defined: Monthly (M,1), Bi-monthly (M,2), Quarterly (M,3), Weekly (W,1), etc.

The billing reference date specifies the first cycle instance's start date. It can be a past date or a future date. The use of this attribute is for the cases where the cycle multiplier is greater than one.

### Cycle Instances

Cycle instances (i.e., the calendar periods defining a single occurrence of the cycle) are managed by Billing and communicated to other components (i.e., Rating and Customer Management) via transactions published by Billing.

### Flexible RC Definitions in PC

Frequency of certain Product Catalog definitions (i.e., RC rates, rating steps, discount packages, and allowances) is flexible. Recurring charges are defined in Product Catalog assuming a base frequency, which can be daily, weekly, or monthly. This base rate, together with the cycle definition, is used to calculate the charge taking into account RC rates for the different frequencies.

The RC period can be defined by a combination of two attributes:

- RC Frequency – Day, Week, or Month
- RC Multiplier –Any positive number

Thus, the following RC periods can be defined: Daily (D,1), Weekly (W,1 or D,7), Quarterly (M,3), etc.

### Flexible Bill Production Frequency

Bills can be produced at a frequency of any multiple of the cycle frequency. Bill production frequency is an attribute defined per billing arrangement. This

introduces an additional billing dimension, which supports different billing frequencies at the BA level. The BA base frequency is always the frequency of the bill cycle to which the BA's customer is attached.

For example, a customer with a monthly bill cycle (M,1) can have two BAs with bill production frequencies 1 and 2. A bill will be produced every month (cycle) for the BA with bill frequency 1, and every two months (cycles) – for the BA with bill frequency 2.

# Controlling Billing Cycles

The progress of both cycle processing and cycle population is tracked using a set of control tables with statuses that are updated as the processing progresses.

The following terms describe the cycle status:

- Cycle Code – Each customer has a cycle code. The cycle code determines the customer population to be processed and the cycle close day.

  For example, for Cycle Code 4, the close day is the 4$^{th}$ of the month and the cycle period is bi-monthly (cycle frequency = M, Cycle Multiplier = 2).
- Cycle – An instance of the cycle code. Each cycle has a cycle code, cycle instance, and year, which together determine the cycle close date.

On bill day, Billing collects all the activities for which the customer is charged until the cycle close day.

**Example:**

- Cycle Code 4:
  - Cycle instance – 2
  - Cycle year – 2003
  - Billing reference date – Feb 5, 2002
- The cycle covers charges created for activities:
  - From Feb 5, 2003
  - To April 4, 2003

## Cycle Statuses

Cycle statuses are controlled and viewed by Application Monitoring and Control (AMC). Subsequent sections describe the main cycle statuses.

### Pending (PN)

Cycles are created in Pending status; the cycles are created automatically for X months in advance, during the Cycle Listener initiation. When a new cycle code is defined in Billing Configurator, pending cycles are created the next time the cycle maintenance is run (EOD job).

Pending status is used for cycles that need to be processed on bill day, but are not yet ready for Bill Preparation.

### Change Cycle Running (CR)

This status signifies that the Change Cycle EOD process is running. The status is then changed to "Closed" by the Close Cycle EOD process.

### Closed (CL)

This status is set by Change Cycle on the cycle close date, after the process is completed.

### Locked for Billing (LC)

Locked for Billing status indicates that the cycle is locked in event processing, and that the final rerating is in progress. Once the final rerating of all the cycle is completed, the billing administrator will have the option to run Bill Preparation via AMC.

### Running Preparation (RN)

Bill Initiator changes the status to "Running Preparation" when a request for Bill Preparation is issued by the bill administrator via AMC.

### Processed (PS)

After the entire cycle population is successfully processed (with no rejected customers detected in the population of the run), the cycle status is changed to "Processed".

### Processed with Rejects (PR)

If part of the cycle population is rejected, or if an operator requests to undo part of or the entire cycle, the cycle status is changed to "Processed with Rejects".

### Rejected (RJ)

The Rejected status is populated by the cycle maintenance job in cases where the system logical date is greater than the cycle close dates of the newly created cycle instances.

### Confirmed (CN)

Set at the end of the Confirmation flow, when the entire cycle population is confirmed.

## Request Listener

The Run Request Listener is a daemon that listens for requests, which are usually submitted by the billing administrator via AMC.

The mechanism is used to run billing flows. For example, a request to run Bill Preparation or Confirmation. A request is the main component in the Run Request Listener process. According to its status, different operations are performed.

The request statuses are as follows:

- Request (RQ) – The initial state when entered by AMC
- Pending (PN)– The request has a dependency on another request that is currently running
- Invoked Process (IP) – The request has been handled. The possible operations involved in processing this request are:
  - Call an exit point
  - Send a message to AMC to invoke a process.
- Abort Fail (AF) – Failed to send a message to AMC to execute the required flow or to process exit point
- Running (RN) – The flow is running
- Finished (FN) – The flow has finished

The responsibility for updating the request's status is as follows:

- AMC – Entering requests with status Request.
- Run Request Listener – Update status to Pending , Invoked Process, or Abort Fail
- Initiator – Update to Running
- Finalizer – Update to Finished.

## Customer Status

Billing maintains different statuses for a customer as a service receiver entity, and for billing arrangement as a payer entity.

### Service Receiver Run Status

- Not Processed (NP) – The customer charges are not processed yet.
- Customer Completed (CM) – All the customer charges, including one-time, recurring, and usage charges, and their discounts are calculated.
- Confirmed (CN) – The service receiver charges passed the confirmation process.

The Customer entity has the Rejected indicator, which is set when the customer is rejected. This indication together with the customer status indicates the stage in which the customer is rejected.

### Service Payer Run Status

- Not Processed (NP) – The Unified Charges file for the billing arrangement is not ready yet, and the Invoicing process has not started
- Payer Completed (PC) – Payer charges are calculated (one-time charges for activities and late payments)
- Unified Charges (UN) – The payer charges are completed, all the related customers are completed, and the Unified Charges file is ready for the billing arrangement, including all the related charges
- Invoiced (IN) – Invoicing process completed.

- Billed (BL) – The billing arrangement passed the Document Creation process
- Confirmed (CN) – The billing arrangement passed the Confirmation process

The billing arrangement (service payer entity) has the Rejected indicator, which is set when the billing arrangement is rejected. This indicator, together with the status, indicates the stage in which the billing arrangement is rejected.

# Changing Cycles

Each customer is assigned to a cycle during customer creation. The cycle determines the day on which the customer receives the bill, and the period of the bill – from the cycle start date to cycle close date.

When a customer wishes to change cycle, the change only takes effect on the close date of the cycle closest to the new cycle close date. For example, a customer requests to change from a weekly cycle during the second week of the month to a monthly cycle, which ends at the end of the month. In this case, the cycle change will take effect on the last weekly cycle of that month (the fourth week cycle that ends prior to the new monthly cycle close date) . This change affects the A&F, Rating, and Billing components.

A change cycle request is always made for a customer, and cannot be made for a specific billing arrangement or subscriber.

Note that:

- In addition to changing the cycle close date, the cycle frequency and bill production frequency can also be changed.
- When the cycle period (frequency and multiplier) of the new cycle and the old cycle differ, it indirectly impacts the bill production frequency, although it is not changed explicitly.

  For example:

  The bill production frequency is 3 (i.e., a bill is produced at the end of every third cycle). Then, if the cycle frequency is changed from Monthly to Bi-monthly, a bill would be produced only once every six months (where it was previously produced once every three months).
- The first bill after a cycle change may be a portion of the new cycle period.

## Request and Change Dates

A cycle change request for a customer is made using CM. CM publishes a transaction, and Billing subscribes to the request.

The new cycle is effective for the customer at the end of the current cycle. I.e., at the end of the current cycle, the charges are produced for all the customers, and a new bill for the new cycle is produced on the close date of the new cycle.

## Change Cycle/BA Frequency Operation

Change Cycle and Change BA Frequency are two different operations that impact the bill date. In addition, since the BA frequency depends on the customer cycle frequency and close day, a change in the customer cycle affects the bill frequency and production day.

**Example:**

A change cycle operation from a monthly cycle to a bi-monthly cycle has the following impacts:

- A BA with a frequency of 1 changes from a monthly bill to a bi-monthly bill
- A BA with a frequency of 3 changes from a quarterly bill to a 6 month bill.

The Change Cycle operation remains the same. The new cycle does not become effective immediately; the current cycle remains effective until it is closed. The new cycle becomes effective for rating when the old one is closed.

Change Production Frequency is a new operation, which is performed in CM. It becomes effective immediately.

Subsequent sections describe the effect of the Change Cycle and Change Bill Production Frequency on the system.

## Change Cycle

Generally, a BA collects its charges when the bill cycle interval multiplied by the bill production frequency is due. This section describes the various situations that may occur when a cycle is changed, and the rule used by the core product to decide whether to select a BA for billing. If the cycle is Monthly, and the bill frequency is 1, the regular cycle change occurs. The BA is billed at the end of the old cycle, and then moved to its new cycle; its next billing period is usually less than one month.

The following rules define whether a BA is selected for bill production in the current cycle run:

- A BA with a frequency of 1 is always selected.
- For a BA with a frequency greater than 1:
  - If no Change Cycle operation occurred since the last cycle run – The BA is selected if the calendar period from the last production of the BA equals the customer cycle frequency multiplied by the BA frequency.
  - If a Change Cycle operation occurred since the last cycle run – The BA is selected if the calendar period from the last production of the BA is greater than the customer cycle frequency multiplied by (BA frequency – 1).

A special rule for Change Cycle case conforms to the current behavior, where the first bill after the Change Cycle operation is usually shorter than a full cycle. This rule generalizes the current behavior by stating that the first bill

production after a Change Cycle operation may be shorter than a full period, but is longer than the period defined by the BA frequency minus one.

*This rule can be used only if BA frequency is 1, and in case no Change Cycle operation occurred since the last cycle run.*

The following example shows that the customer will be billed in its new cycle for a partial period, from 03/03 to 15/03.

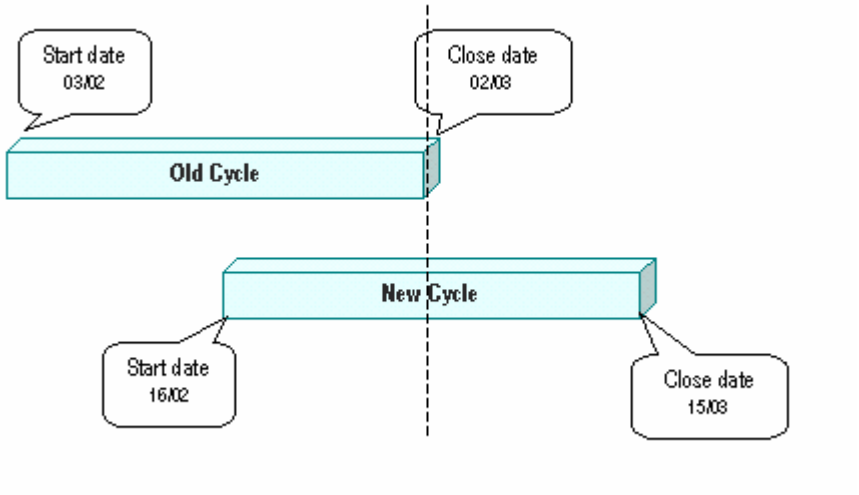


**Figure 7-1: Billing for part of a cycle – Example**

Generalization of this behavior produces the following rule, which defines whether a BA is run or not:

**0 < (Bill Production Frequency – 1) * Cycle Multiplier < Number of months since last bill run**

*Calculations should use the exact month and date, not the rounded values. For example, when the last production date = 15/1, and the current bill run is on 02/03, the Number of months since last bill run = ~1.5).*

*The calculations are based on the current situation of the customer (customer cycle and customer frequency), regardless of the history of changes made to these two parameters.*

## Change Bill Production Frequency

Change Bill Production Frequency takes effect immediately. When the frequency is changed to a higher value, the bill is delayed; when the frequency is changed to a lower value, the bill is produced in the nearest relevant cycle.

The frequency relevant for the current bill run is the frequency selected by Bill Day Initiator. If the frequency is changed in CM during the bill run and before confirmation, the previous frequency is used (unless Bill Day Initiator was rerun).

### Change Cycle Confirmation

The Change Cycle Confirmation process confirms change cycle requests for customers where all their subscribers, agreements, and billing arrangements are confirmed. The confirmation is published to Customer Management via the TRB.

Billing cannot accept a change cycle request for a customer with a pending (not confirmed) change cycle request after the cycle close date of the old cycle.

A change cycle request for a customer with a pending request can be accepted up to the previous cycle's close day, though from this point until bill confirmation no additional change cycle request or change cycle request cancellation can be accepted.

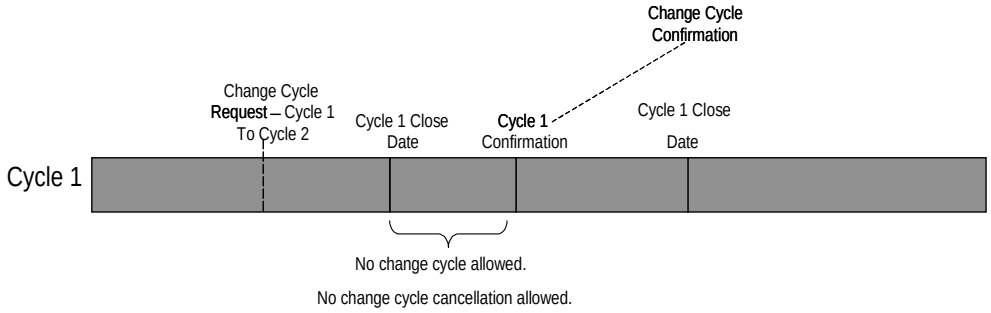


**Figure 7-2: Change Cycle confirmation**

## Change Cycle Impact on Customers

Because Change Cycle requests are issued at the customer level, they impact both the payer (related to the customer) and service receiver (billing arrangements and subscribers).

In most cases where there is no cross-cycle distribution, both the billing arrangement and subscriber change their cycle at the same time. In case of a cross-cycle distribution, the impact on each entity is described as follows:

- Service Payer – The Invoicing module of the service payer collects all charges from the database or from Charge Creation. The change impacts on the cycle close day, which is the day on which the Bill Day processes run for the billing arrangement.
- Service Receiver – Charge Creation collects usage charges on different days. On the first bill in the new cycle, the charges are calculated in a different way:
  - Usage charges are prorated for a shorter period of the new cycle – From the previous cycle close date to the new cycle start date
  - Recurring charges are prorated for the shorter period of the new cycle

# Change Cycle Impact on Recurring Charge Calculation

RCs are prorated appropriately as a result of any Change Cycle operation. RCs are automatically marked for rerating whenever a change occurs in the cycle frequency or multiplier (the number of months, weeks, or days in the cycle).

The cycle change impact on the RC calculation depends on whether the recurring charge calculation type is In Advance or In Arrears.

## In Advance

If the RC is charged in advance, the RC for the time between the previous cycle close date + 1 and the new cycle close date should appear in the last previous cycle's bill.

The Change Cycle Impact process updates the RC Rates table as follows:

- In case there is a rate change, the RC Rate record with the previous cycle code is expired; the expiration date is the old cycle close date. In case that there is no rate change, the old RC is expired with the new cycle close date.
- In case there is a rate change, a new RC Rate record is added with the old cycle code and the new rate:
  - Effective Date – Old cycle close date + 1
  - Expiration Date – New cycle close date
- The proration process that runs on bill day for the previous cycle prorates the advance charge based on the updated RC Rates table.
- A new RC Rate record is added with the new cycle code and the new rate with the following dates:
  - Effective Date – New cycle close date + 1
  - Expiration Date – Empty

## In Arrears

The RC in arrears is charged for the period between the previous cycle close date + 1 and the new cycle close date appears on the first new cycle bill.

The Change Cycle Impact process updates the RC Rates table as follows:

- The RC Rate record with the previous cycle code is expired; the expiration date is the previous cycle close date.
- A new RC Rate record is added with the new cycle code and the new rates with the following dates:
  - Effective date – Previous cycle close date + 1
  - Expiration date – Empty

After these updates, the RC proration of the new cycle prorates the recurring charge.

# Cross-cycle Distribution

Charges incurred by a service receiver are generally billed in the same cycle. In some cases, charges are distributed to a billing arrangement in a different cycle. In these cases, the subscriber charges are collected and calculated in the subscriber's cycle, and are entered into the database to wait for the target billing arrangement's cycle.

In the billing arrangement's cycle, charges from service receivers in other cycles are collected from the database by the Invoicing process.

For example, a private customer has a subscriber and billing arrangement in monthly cycle 20. The subscriber distributes part of its charges to the corporate billing arrangement in monthly cycle 5.

All the private subscriber charges are generated at the end of cycle 20. The charges that are distributed to the private billing arrangement are processed immediately, while charges distributed to the corporate billing arrangement are written to the database.

On cycle 5 bill day, the corporate charges are generated and invoiced immediately, while the private charges from cycle 20 are collected from the database and invoiced.

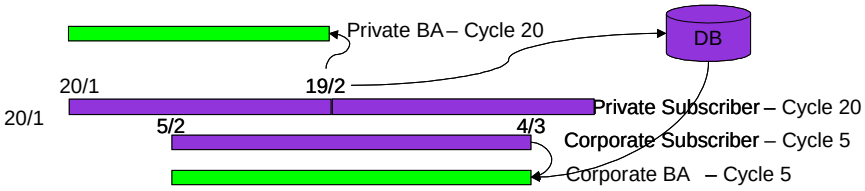


**Figure 7-3: Cross-Cycle distribution – Example**

# 8. CHARGES

In order to improve performance on bill day, Amdocs Billing handles charges in two different process flows:

- Charge Preparation processes run constantly as daemons controlled by the CSP. They use Pricing Engine to calculate recurring and one-time charges and save them as charge requests (pending charges), which are generated as charges on bill day by the Bill Preparation processes.
- Charge Creation processes collect all input charge requests for service receivers, and generate charges for them for billing arrangements. In addition, Charge Creation calculates discount credits based on service receiver performance.

Charges can be generated for the following types of charge requests:

- Accumulated Usage Charge Requests – From event processing component
- Recurring Charge Requests – Inserted into the Billing database as recurring charge rates on the effective date, and collected and prorated by Charge Creation on bill day
- One-Time Charge Requests – Inserted into the Billing database as pending charge requests on the charge creation date, and collected by Charge Creation on bill day

Charges can also be generated upon period calculations. For example, for discount plans that are based on period performance, if the service receiver achieves the requirements for the discount plan, a credit is generated.

## Charge Definitions

The implementation team can define charge codes and charge entity types using Product Catalog. These definitions are exported to Billing Configurator. The billing attributes are then set in Billing Configurator for charge codes imported from Product Catalog.

### Charge Entity Type

Charge Entity Type (CET) defines a set of dynamic attributes that are relevant for all charges related to that CET.

Each attribute can acquire a different value per charge. The value should be defined on the charge request for each charge of this type. The attribute can be defined as mandatory (meaning all charge requests with charge codes that relate to this CET provide a value for this charge) or non-mandatory.

**Example:**

Dynamic Attributes:

- Duration – For voice-based usage charges, the duration of a call to which the charge relates
- Period Name – For usage charges, the period in which the event occurs (peak or off-peak)
- Provider – For third-party charges, the provider that creates the charge

Each charge in the system has a charge code that relates to the CET. The following table provides examples of CETs.

| Charge Entity Type | Attributes |
|---|---|
| Voice | Period Name, Duration, First Event Date, Last Event Date, Number of Events |
| Data | Volume, Free Volume, Number of Events |
| One-time Activity Charges | Activity, Reason |
| Recurring Charges | Period Start Date, Period End Date |

## Charge Code

Charge codes are defined in Product Catalog and synchronized with Billing. Each charge request received has a charge code that refers to the charge.

Each charge code has a set of rules that determine the Billing treatment for the charge:

- Tax Code – Reference to one of the tax codes defined in the system
- Revenue Code – Recurring, one-time, or usage charge.
- Tax Included – Indicates whether the charge amount includes tax or the charge request includes the Tax Amount dynamic field, which is already calculated
- Create Zero Charge Policy – Indicates whether to create a charge when the amount is zero
- Charge Entity Type – The CET to which the charge relates

Each charge is affected by the rules that are set for its charge code.

The charge code can also be used to set rules for other components to which Billing extracts information, such as Accounts Receivable and the bill layout utility. These rules can be defined in Billing Configurator as dynamic attributes for the charge code, and extracted by the Billing Extract processes.

Examples of these rules include:

- Eligible Late Payment Indicator – Indicates whether the charge is eligible for a late payment calculation by Accounts Receivable.
- Service Group – Usage charges may be divided into categories in order to group them on the bill and display a title for each group, e.g., Voice, Data, and Roaming. The bill layout utility uses this rule.
- Formatting rules for the bill layout utility.

The implementation team uses Billing Configurator to set the rules for the charge codes. A rule can apply to a specific charge code or to all charge codes.

When a new charge code is introduced, the implementation team can set values for the charge code rules.

The charge code is related to a CET, which defines the charge attributes. Different charge codes can refer to the same CET.

The charge code, CET, and the relations between them are defined in Product Catalog.

# Generating Usage Charges

Accumulated usage charges are extracted from the Usage Rater on bill day by a process that is synchronized by AMC as part of the Bill Preparation processes.

Before extracting the accumulated charges, Billing sends an event to Rater to lock the cycle for rates for the bill cycle, which means that late events for this cycle are guided to the next cycle.

Before the extract, Rater performs the last rerate for the cycle. From this point on, only Billing can initiate rerating.

Usage charges for prepaid accounts are extracted from Rater on bill day in the same way as postpaid usage charges. Billing considers prepaid usage charges as any other charges, and reports them to Accounts Receivable during the Confirmation process.


*Prepaid charges are not considered for invoicing. Billing creates these charges in the database and uses them for the prepaid statement preparation.*

# Preparing Recurring and One-time Charges

Actual charge preparation is performed using the same features and algorithms that are used for Quotation Server, via the same APIs and transactions. For details, see the *Quotation Server* section.

## Recurring and One-time Charge Preparation

The BTL-SOR daemon process runs constantly. It listens for transactions (sent from Customer Management, OMS, OMSE, AR, etc.), and creates charge requests (pending charges) to be collected on the bill day of the cycle.

Billing calls Pricing Engine (in stateless mode) to calculate the charge amounts, and prepares the various events for Pricing Engine for the different types of recurring and one-time charges.

# Generating One-time Charges

## Collecting One-Time Charge Requests

Billing supports different types of one-time charges that can be defined dynamically by PC:

- Customer Management activity charges.
- Offer-level charges that are determined based on Offer activities defined in PC. For example core implementation includes offer setup charges and early termination charges.

### One-time Charges Based on Customer Management Activities

Billing subscribes to preconfigured transactions (defined in Billing Configurator) on the TRB. Billing translates each Customer Management transaction into a Pricing Engine Customer Management Activity event, and sends it to Pricing Engine with the activity, reason code, customer parameters, and the list of offers owned by the customer.

### One-time Charges for Offer Setup

A CSP may charge a customer for setting up an offer. This is usually done for offers that require additional resources, such as an additional line.

Billing subscribes to pre-configured transactions on the TRB (which may be different from the set of pre-configured activities for Customer Management activity-based charges, as described above). Billing translates each Customer Management transaction into a Pricing Engine Offer Setup event, and sends it to Pricing Engine with the new offer and the customer parameters.

### Early Termination Penalty Charges

A CSP may create a penalty charge for early termination of a specific offer (or price plan) with minimum commitment.

Billing subscribes to pre-configured transactions on the TRB (usually, Remove Offer or Cancel). Billing translates each transaction into a Pricing Engine Termination Penalty event, and sends it to Pricing Engine with the removed offers and the customer parameters.


*The OC rate may depend on one of the customer parameters, and on the offers that the customer owns. The rating scheme for the OC is defined in Product Catalog.*

All charge requests are maintained in Billing as pending charges and collected on bill day for Bill Preparation. Prepaid charges are reported to Replenishment Management at charge creation time.

### Other Sources of Charges

There are two additional ways of receiving charges in the system:

- Receiving charge requests on any day (e.g., m-commerce charges). The charges are treated as pending charges and collected on bill day.
- Receiving charges on bill day (e.g., late payments from Accounts Receivable or charges from a different billing system). The process that creates the charges, and the format of the charge file, can be defined in Billing Configurator. Once they are defined, Billing can receive the file and create charges based on it.
- Receiving charge requests via an API, which saves the charge as a charge request.

The file format of the charges can be defined in Billing Configurator and mapped to charges in the system. Billing supports the following file formats:

- Delimited – Fields are separated by a specific delimiter
- Fix – Each field starts at a specific location
- Name & Value – The file is composed of attributes in the "name=value" format

### Charges with Tax Included

Charges can be received by Billing with tax already calculated. This may occur when charges are sent from third-party system, or for roaming events.

A charge with tax can be received in the following ways:

- Charge amount and tax amount in different fields – A special CET must be defined for this purpose.
- Charge amount with tax included – The Tax Reversal function is used to calculate the net amount of the charge and the tax amount. This functionality is supported only if the tax calculation method is internal.

## Recurring Charges

### RC Rates Table

Recurring charges are calculated by Pricing Engine, using definitions made in Product Catalog. Billing extracts from Pricing Engine all the recurring charges per subscriber, unit, or customer and maintains this information in the RC Rates table. The RCs might differ from one subscriber to another if they are dependent on the customer parameters. For example, the bandwidth of the data services may affect the RC of the service.

The Recurring and One-time Charge Preparation process prepares a Billing table with RC rates and dates (effective and expiration). Based on this information, Bill Preparation extracts the charges and prorates the recurring charges (if necessary).

The RC rates are maintained in Billing with the following main attributes:

- Subscriber/Agreement Number
- Customer Receiver
- Amount
- Manual Override Amount – Used if the original amount is overridden for the subscriber or agreement.
- Proration Indicator – Indicates whether proration needs to be calculated for the rate. This indicator is received from PC, and is used to determine the effective and expiration dates for the rate. The proration is then calculated based on these dates.
- Price Plan Indicator – Indicates if the current offer is a price plan.
- Dynamic Attributes – The charge code dynamic attributes. The charge code determines the CET that determines the charge dynamic attributes.
- First Cycle Effective Date – The close date of the first billing cycle in which the charge will be considered on Bill Day.
- Last Cycle Expiration Date – The close date of the first billing cycle that will consider the expiration of the RC by Bill Day.
- Offer – The offer used for the charge calculation.
- Offer Item – The offer item used for the charge calculation.
- Offer Instance – The offer assigned to a subscriber or agreement. The subscriber or agreement may have the same offer effective in different periods of the cycle, or the same offer effective more than once for allowance offers (free events).
- Currency
- Effective Date
- Expiration Date
- Pay Channel – The pay channel responsible for paying for the RC.
- Charge Code – Defined in Product Catalog and received from Pricing Engine.
- RC Type – In Advance, In Arrears, or Effective at Close Date (defined in Product Catalog and received from Pricing Engine).
- Generation Frequency – Indicates the number of cycle periods for which the RC is charged.
- Cycle Code – The cycle code of the service receiver.
- Insert Activity Sequence – The activity that inserted the RC, where the sequence is extracted from the Activity History table.
- Expired Activity Sequence – The activity that expired the RC, where the sequence is extracted from the Activity History table.
- Multiple Charge Indicator – Indicates whether to create multiple charges or a single combined charge when the cycle frequency contains multiple instances of the RC frequency.

- Frequency – The RC frequency (Daily, Weekly, or Monthly).
- Frequency Multiplier – A positive number used as a multiplier of the frequency.

The RC Calculation process flow maintains RC rates with their effective and expiration dates for each RC. Bill Preparation uses this information for the RC proration.

### Maintaining the RC Rates Table

Billing subscribes to pre-configured transactions on the TRB (defined in Billing Configurator) that may affect RC rates. Some examples of these transactions are:

- Change Parameter – CM transaction. When one of the parameters of a subscriber is changed, the recurring charge rate for this subscriber may change.
- Suspend/Restore Subscriber – CM transaction. The subscriber status may be a parameter of the recurring charge rate (to support a reduced recurring charge rate for suspended subscribers). In such cases, Billing treats the subscriber status as a parameter, and this activity – as a Change Parameter activity.
- New Subscriber – CM transaction. New recurring charge rates should be calculated for the subscriber.
- New Offer – CM transaction. New recurring charge rates should be calculated for the subscriber or agreement.
- Override RC – Billing transaction. Change the recurring charge rate for a specific service receiver.

These transactions are defined in Product Catalog. If, in a specific implementation, the Suspend/Restore subscriber does not affect the recurring charge rate, it can be indicated to be not rated. If other activities affect the rate, they can be easily defined in Product Catalog, including the offer activities to be rated.

When one of the transactions is identified, Billing sends only the offers of the subscriber or agreement that were marked for rerate to Pricing Engine to rerate the recurring charge.

Charge Analyzer receives the results from Pricing Engine and, based on the RC policies, decides what action to perform.

Due to performance reasons, different types of pre-configured transactions may be sent directly to Charge Analyzer, while omitting the Pricing Engine step. These transactions affect recurring charge distribution or expiration date, but not the rate. They include:

- Cancel Subscriber – The expiration date for all the subscriber's RC rates is set to the activity date.
- Change Charge Distribution – The record with the RC rate for the old pay channel is set with the expiration date of the activity date, and a new RC rate record is inserted with the activity date and the new pay channel.

These activities are defined in Billing Configurator, as well as other activities that may affect the RC or OC rate, and can be modified to include or exclude other activities.

### New Rates Received from Pricing Engine

Each rate received from Pricing Engine contains the following information:

| Field | Type | Description |
|---|---|---|
| Service Receiver Type | String | Billing Arrangement/ Unit/ Subscriber. |
| CET Name | String | Defines the CET according to the charge code. |
| Service Receiver ID | String | The ID of the service receiver. |
| Service Receiver Customer | Numeric | The ID of the customer that receives the offer. |
| Billing Arrangement | Numeric | Billing arrangement identifier. |
| PCN | String | Pay Channel identifier. |
| Offer ID | Numeric | Offer identifier. |
| Pricing Item ID | Numeric | Pricing Item identifier. |
| Offer Instance | Numeric | Sequence number of the offer ID. |
| Amount Currency | Currency Code | The code of the currency used for the amount. |
| Amount | Numeric | The RC amount. |
| Override Amount | Numeric | Copied from the input event. |
| Charge Code | String | The charge code associated with the pricing item. This field is taken from the Charge Code PIT parameter. |
| Proration Indicator | Numeric | Indicates the proration method used for the PIT. It is taken from the Proration Method PIT parameter. Valid values are:<br>1 – Prorated<br>2 – Not prorated |
| RC Type | Numeric | Indicates the payment timing setting for the PIT, taken from the Payment Timing PIT parameter. Valid values are:<br>In Advance<br>In Arrears<br>Cycle Close Day – Paying for a service only if it was effective on the cycle close day |
| Generation Frequency | Numeric | The recurring charge frequency related to the billing frequency. |
| Price Plan Indicator | Yes/No Indicator | Indicates if the offer is a basic offer. |
| Multiple Charge Indicator | Yes/No Indicator | Indicates whether to create multiple charges or a single combined charge when the cycle frequency contains multiple instances of the RC frequency. |

| Field | Type | Description |
|---|---|---|
| Frequency | Char | The RC frequency (Daily, Weekly, or Monthly) |
| Frequency Multiplier | Numeric | A positive number used as a multiplier of the frequency. |

RC Rate Analyzer checks if an effective RC Rate record already exists with the same key attributes:

- Service Receiver Type
- Service Receiver ID
- Offer
- Offer Instance
- Pricing Item

Charge Analyzer handles each transaction according to a set of Billing activities and policies defined in Billing Configurator. Each transaction can be mapped to one or more RC activities.

For example, the activation of a new subscriber is usually mapped to the addition of a new offer and a new basic offer.

If, as a result of an RC activity, a record is inserted or expired (e.g., an addition or removal of an offer), the effective date of the new record and the expiration date of the old record depend on the Proration Old New policy and the In Advance policy.

If, as a result of an RC activity, an attribute on the record is updated (i.e., old record is expired and new record is inserted), the effective date of the new record depends on the Proration Old New policy and Advance policy. The expiration date is set one day prior to the calculated effective date.


*More detailed information about the New RC Rate policy can be found in the next section.*

If the rate is changed for an In Advance recurring charge, the effective and expiration dates depend on the In Advance policy, which is determined in Billing Configurator:

- Immediate – The change date is the current date:
  - New Rate at Change Date – New rate effective at change:
    - Expiration date for the old rate is the change date – 1
    - Effective date for the new rate is the change date
  - Old Rate at Change Date –Old rate effective at change:
    - Expiration date for the old rate is the change date
    - Effective date for the new rate is the change date + 1
- Next Cycle – The change date is the next cycle start date.
  - Expiration date for the old rate is the next cycle start date – 1.
  - Effective date for the new rate is the next cycle start date.

### New RC Rate Policy

The New RC Rate policy is used to determine the rate used on the change date. For example:

- Cycle Start Date – 1/4
- Cycle End Date – 30/4
- The price plan was changed on 15/4:
  - Recurring charge for the old price plan – 100
  - Recurring charge for the new price plan – 150
- New Rate at Change Date – RC = 14/30*100 + 16/30*150 = 126.66
- Old Rate at Change Date – RC = 15/30*100 + 15/30*150 = 125
- For New Rate at Change Date, the effective date for the new RC Rate is the change date, and the expiration date for the old rate is change date – 1.
- For Old Rate at Change Date, the effective date for the new RC Rate is the change date + 1, and the expiration date for the old rate is change date.

### RC Policies

The implementation team can use Billing Configurator to determine the RC activities and policies for every transaction that is marked as RC-affected in Billing Configurator.

Whenever a new record is received from Pricing Engine, Billing checks the RC policies defined for the transaction.

Each transaction is mapped to one or more of the following RC activities:

- Addition of a new offer – A new record is added to the RC Rates table if the processed offer does not exist in this table, and is not a basic offer.
- Addition of a new basic offer (price plan) – A new record is added to the RC Rates table if the processed offer does not exist in this table, and is a basic offer.
- Expiration of an offer – Expires an offer if the processed offer exists in the RC Rates table, and it is not a basic offer.
- Expiration of a basic offer (price plan) – Expires an offer if the processed offer exists in the RC Rates table, and it is a basic offer.
- Change pay channel of an offer – If the processed offer exists in the RC Rates table, and the offer's pay channel differs from the current one, the current offer is expired, and a new record with the new pay channel is inserted into the RC Rates table.
- Change rate of an offer – If the processed offer exists in the RC Rates table, and its amount differs from the current rate, the current offer is expired and a new record with the new recurring charge amount is inserted into the RC Rates table.
- Change Cycle – If a customer changes its cycle code, the old RC is expired, and a new RC is inserted with the new cycle code.
- Resume Subscriber – When a cancelled subscriber is resumed, the RC records become effective.

The implementation team can set different values for RC policies for each of these activities as follows:

- Consider Proration Old/New policy – If this policy is set to No, Billing ignores the New RC Rate policy for this activity and always sets the expiration date or effective date of the RC to the effective date of the transaction. For example, for the Remove Offer and Expiration of New Offer RC activities, this policy is always No. When an offer is removed, the service receiver should always be charged up to the removal date, even if the New RC Rate policy is set to New.
- Override In Advance policy – This policy is used only for In Advance RC Offers. If a rate is changed or expires, the change may take effect on the change date and cause a proration for a charge that was already paid, or may take effect only for the next cycle. For example, if a service receiver is charged $10, and the RC rate is changed mid-cycle to $12, if the policy is Immediate, in the next cycle the payer has to pay the $2 for part of the cycle, and if the policy is Next Cycle, the subscriber has to pay $12 only for the next cycle. If this policy does not exist, the policy on the activity overrides the market policy defined in Billing Configurator.
- Force Proration policy – RC rates that are marked as not prorated in Product Catalog may need to be prorated in some cases. For example, if the RC for offer X is not prorated, and a new subscriber purchases the offer in mid-cycle, the subscriber has to pay the full-cycle RC. However, if the subscriber changes a price plan, and the RC is not prorated in both the old and new price plan, the subscriber should not pay the full RC for both price plans. In this case, the Adding New Offer and Removing New Offer activities for the Change Price Plan transaction have the Force Proration indicator set, and although the RC is not prorated, it is prorated if the price plan is changed.
- Billing Services Expiration policy – This policy is for RC rates for billing services, such as call details. The RCs are of cycle close day type. If an offer with such RCs is removed, the expiration date for the offer is the offer removal date, so it is not considered for the next bill. To be considered for the next bill, the policy should be expired on the expiration date. If, on the other hand, a subscriber is cancelled, the offer should be removed on the cycle close day because the subscriber receives the billing services for the final bill. In this case, the policy should be expired on the cycle close day.

## Error Handling

Pricing Engine is part of the Charge Preparation process. This helps the error handling display the exact problem encountered during charge calculation. Usually, errors occur during charge preparation due to a mistake in the implementation definitions created in PC. Enabling Pricing Engine error management and displaying the exact errors created by PE is a great improvement of the error handling in the charge preparation.

Following is the list of error categories handled during charge preparation:

- Customer data missing
- Invalid input record

- Missing information in Product Catalog definitions
- Internal Rater errors

Pricing Engine errors contain detailed information regarding the exact calculation problem.

Any error created in the processing flow of the charge preparation is saved in two tables:

- BL1_REJECTED_TRX – Holds information about the rejected transactions. Its main task is to keep track of the rejected customers, so that Bill Day Initiator can ignore these customers in the bill day run until the errors are fixed.
- TRB1_SUB_ERRS – This table is part of the TRB component. Keeps all the rejected TRB transactions. Each of the rejected charge preparation transactions is displayed in this table including the details of the problem it has encountered during charge preparation.

## Proration and Charge Creation

Recurring charges are created during Bill Preparation. The process scans the RC Rates table and creates a charge for each RC rate that is identified by Pricing Engine as requiring prorating. The process also scans the recurring charge frequency and, if the frequency is not 1, checks the last date that the subscriber was charged for the RC and decides if a charge should be created for the subscriber.

The Proration process considers the following date ranges:

- Period Start and End Date – If the RC frequency is 1, the cycle start and close date are used. If the frequency is greater than 1, the range may cover two or more cycle periods.
- Effective and Expiration Date – The date range for which the RC should be prorated.

Proration is performed for each RC Rate record that has an effective date after the cycle start date or an expiration date before the cycle close date.

The charge rate may be prorated by the effective and expiration dates of the RC rates, according to the following rules:

| Effective Date | Expiration Date | In Advance/ In Arrears | Charge | Explanation |
|---|---|---|---|---|
| Before cycle start date | On or after cycle close date | In advance | A charge for the full amount is created for the next cycle, unless the expiration date is on the cycle close date. | This is the most common situation, with no special activities requiring proration. |
| Before cycle start date | On or after cycle close date | In arrears | A charge for the full amount is created for the previous cycle. | |
| After cycle start date | On or after cycle close date | In advance | A prorated charge is created for the previous cycle.<br>A charge for the full amount is created for the next cycle, unless the expiration date is on the cycle close date. | This situation occurs for the following activities:<br>New Subscriber<br>New Offer<br>Parameter Change<br>Rate Change<br>Change Charge Distribution |
| After cycle start date | On or after cycle close date | In arrears | A prorated charge is created for the previous cycle. | |
| Before cycle start date | Before cycle close date | In advance | A prorated credit is created for the previous cycle. | This situation occurs for the following activities:<br>Cancel Subscriber<br>Remove Offer<br>Parameter Change<br>Rate Change<br>Change Charge Distribution |
| Before cycle start date | Before cycle close date | In arrears | A prorated charge is created for the previous cycle. | |
| After cycle start date | Before cycle close date | In advance | A prorated charge is created for the previous cycle. | This situation occurs when:<br>Subscriber is activated and cancelled in the same period.<br>Offer is added and removed in the same period.<br>Parameter is changed twice in the same period.<br>Rate is changed twice in the same period. |
| After cycle start date | Before cycle close date | In arrears | A prorated charge is created for the previous cycle. | |


*The proration ratio is calculated using the effective and expiration dates. In case a credit is created, the proration amount is based not on the amount in the RC Rates table, but on the actual amount that the subscriber is charged for that RC in the last cycle, considering credits for that charge. Billing subscribes to Accounts Receivable adjustments in order to keep records for adjustments to RCs.*

The following table describes the charges created during Bill Preparation, based on activities that occur during the cycle period.

| Activity | In Advance/ In Arrears | Charge | Proration Period |
|---|---|---|---|
| New Offer<br>New Subscriber | In advance | A prorated charge is created for the previous cycle period. | Activity date<br>Period close date |
| | | A charge for the full amount is created for the next cycle period. | None |
| | In arrears | A prorated charge is created for the previous cycle period. | Activity date<br>Period close date |
| Cancel Subscriber<br>Remove Offer | In advance | A prorated credit is created for the previous cycle period. | Activity date + 1<br>Period close date |
| | In arrears | A prorated charge is created for the previous cycle period. | Period start date<br>Activity date |
| Change Parameter<br>Rate Change | In advance | A prorated credit is created for the previous amount for the previous cycle period. | {Activity date + 1 or activity date}<br>Period close date |
| | | A prorated charge is created for the new amount for the previous cycle period. | {Activity date + 1 or activity date}<br>Period close date or expiration date (the earliest) |
| | | A charge for the full amount is created for the next cycle period. | None |
| | In arrears | A prorated charge is created for the previous amount for the previous cycle period. | Period start date<br>{Activity date or activity date – 1} |
| | | A prorated charge is created for the new amount for the previous cycle period. | {Activity date + 1 or activity date}<br>Period close date |
| Change Charge Distribution | In advance | A prorated credit is created for the previous pay channel for the previous cycle period. | {Activity date + 1 or activity date}<br>Period close date or expiration date (the earliest) |
| | | A prorated charge is created for the new pay channel for the previous cycle period. | {Activity date + 1 or activity date}<br>Period close date or expiration date (the earliest) |
| | | A charge for the full amount is created for the new pay channel for the next cycle period. | None |

| Activity | In Advance/ In Arrears | Charge | Proration Period |
|---|---|---|---|
| | In arrears | A prorated charge is created for the previous pay channel for the previous cycle period. | Period start date {Activity date or activity date – 1} |
| | | A prorated charge is created for the new pay channel for the previous cycle period. | {Activity date + 1 or activity date} Period close date |
| None | In advance | A charge for the full amount is created for the next cycle period. | None |
| | In arrears | A charge for the full amount is created for the previous cycle period. | None |


*A combination of activities can occur during one cycle period:*

- *Activation and Change Parameter*
- *Change Parameter and Cancellation*

*Each of these cases can result in the creation of charge(s) as described above, for each period. All these cases are covered in the Proration Rule table.*

## Proration Formula

The proration formula is determined using Billing Configurator, and can be one of the following:

- Cycle Days Proration – The number of days between the Proration Start Date and Proration End Date is divided by the number of days between the Cycle Start Date and Cycle Close Date.
- Fixed Days Cycle Proration – The number of days between the Proration Start Date and Proration End Date is divided by a predefined numerator, such as the number of days in a month (28, 30, 31). In case the rate covers the entire cycle, no proration is performed.
- Tiered RC – According to the number of effective days in the cycle, for example:
  - Up to 10 days – No RC
  - 11-20 days – 50%
  - 20-up – 100%
- External – The proration formula is external, and is handled in the customization layer. In this case, the core module invokes a call to a predefined method that receives the following major parameters:
  - Cycle Start Date
  - Cycle End Date
  - Proration Start Date
  - Proration End Date
  - Cycle Frequency

- Cycle Multiplier
- RC Frequency
- RC Multiplier

The method returns the proration factor, which is then multiplied by the full RC rate in the core module.

### Prepaid Recurring Charges

Prepaid recurring charges are reported to Accounts Receivable and Replenishment Management by publishing a prepaid charge transaction as part of the Bill Confirmation flow.

## Overriding Recurring Charge Rates

Billing provides the option to override RC amounts using an API. The user can override the RC amount for a specific subscriber and offer. The overridden amount is entered into the RC Rate table as a new record with the effective date of the transaction. It is possible to indicate if the overridden amount should be prorated on the next bill day, i.e., whether the next bill day process takes the overridden amount as the rate for the all cycle period, or prorates the recurring charge from its effective date, and the previous recurring charge up to the override date.

# 9. BILLING DISCOUNTS

Billing discounts are calculated using Discount Engine. Discounts are calculated on bill day for the total performance of a subscriber in a cycle period, or a group of subscribers that belong to the same hierarchy unit.

Billing discounts refer to total charges of the various revenue types, including:

- Usage charges
- Recurring charges
- One-time charges

Rater Discounts refer to usage charges only.

## Basic Discount Terms

This section describes the commonly used discount-related terms.

### Discount Package

Discount Package comprises one or more discount items. Discount packages and discount items are defined in Product Catalog, and are attached to an offer. When a user purchases a discount offer, the user is entitled to the Discount Package terms.

### Discount Items

Discount items are sets of rules that define the discounts. Each discount item comprises the following eligibility and contributing rules:

- Contributing Rules – Customers that comply with the contributing rules are entitled to the discount. Examples of contributing rules are:
  - Spending more than $500 on international calls – Contributing rule is that the charges are for international calls; the accumulative attribute is the monetary amount.
  - Downloading more than 800 MB in GPRS sessions – Contributing rule is that the charges are for GPRS, the accumulative attribute is the download volume.
  - Number of a customer's subscribers is greater than 15 – Contributing rule refers to subscribers for customers; the accumulative attribute is 1.

- Eligibility Rules – Charges that comply with the eligibility rules are the charges the discount refers to when crediting the customer. For example:
  - 15% discount on all international call charges – Eligibility rule is that the charges are for international calls.
  - 10% percent discount on recurring charges – Eligibility rule is that the charge is a recurring charge.

In some cases, the contributing rules and eligibility rules are the same:

- 15% discount on GPRS charges – If the total charge on GPRS sessions is greater than $100, then the contributing and eligibility rules refer to the GPRS charges.
- 10% discount on international call charges – If the total charges for international calls are greater than $400, then the contributing and eligible rules refer to the charges on international calls.

In other cases, the contributing rules are different from the eligibility rules:

- 10% discount on GPRS charges – If the customer spends more than $500 on international calls, then the contributing rule refers to the international call charges, while eligibility rule refers to the GPRS charges.
- 15% discount on recurring charges – If the total volume for GPRS usage is greater than 300 MB, then the contributing rule refers to charges for GPRS, and the eligibility rule refers to the recurring charges.

Contributing and eligibility rules are sets of conditions on predefined attributes. The predefined attributes are defined as discount entity types in Product.

## Discount Entity Types

Discount entity types (DETs) can be based on a charge entity – a subset of charge attributes, or on other attributes. The core system supports only charge DETs. However, it is possible to define additional DETs in Product Catalog and support them in the customization layer.

Eligibility rules can be based only on CETs because the discount percent should refer to amounts, which are charge attributes.

The charge DET is a set of attributes derived from charge attributes, which the user can define using Product Catalog as part of the charge entity definitions. Charge DET attributes can be selected from the following charge attributes:

- Charge Code
- Charge Entity Type
- Offer
- Revenue Code
- Any dynamic attribute defined for a specific CET.

The above attributes are defined as Qualification attributes.

The charge DET also has accumulated attributes, which can be any accumulative charge attribute, e.g., amount, volume, or duration. These attributes are defined as Accumulation attributes.

# Discounting Process

## Discount Engine

Discount Engine is a standalone component that maintains the Discount Plan definitions in the system, knows how to retrieve Discount entities, and returns credit to customers. Discount Engine is invoked by the Charge Creation process during Bill Preparation.

Discount Engine returns one credit for each eligible Discount entity. I.e., if Billing sends to Discount Engine 40 charges that entitle the subscriber to a 10% discount, a credit of 10% of the value of the charge is returned for each eligible charge.

Discount Engine receives the discount package definitions from Product Catalog via the Reference Table Synchronization process. Billing provides Discount Engine with the Customer Hierarchy snapshot of the customer for the relevant cycle period. Customer Management assigns discount plans (as part of an offer) to customers in the customer hierarchy model. Bill Preparation is responsible for feeding Discount Engine with all the discount entities, so that Discount Engine can calculate the credits and return them to Charge Creation, which generates credit charges for the customer.

### Discount Entity Type

Billing receives from Product Catalog the DET definitions, and the CETs. The mapping of a CET to charge DET uses XML mapper defined in the core global (static) attributes. Dynamic attributes do not require mapping (i.e., same definition in Discount Engine and Charge Entity).

Billing knows how to translate each charge in the system to a discount entity, and sends it to Discount Engine.

Discount charges can include Discount Engine (dynamic) attributes if they are defined on the CET by Product Catalog.

### Additional Discount Entity

Additional discount entities are not generated by the core Billing process, but can be received from a customized process prior to Charge Creation. The customized process provides a file with discount entities. The file format, mapper, and process are defined in Billing Configurator.

Charge Creation serves as a proxy. It passes all the discount entities to Discount Engine for calculation. The credits that Discount Engine returns also take into consideration the additional discount entities.

### Examples of Discount Entity Types

- Subscriber Offers – The attributes are Offer Type (Basic or Additional), Offer and Subscriber Status. The accumulated field is No of Occurrences. This DET can be used for discounts based on the number of subscribers (counts basic offers of active subscribers).
- Subscriber Loyalty – The attribute is the Customer Type, and the accumulated field is the number of months since the subscriber was activated. In this case, each subscriber is mapped to a Subscriber Loyalty

discount entity. This DET can be used to qualify loyal subscribers for an additional discount.

The following diagram demonstrates the flow of discount entities and credits in the system:

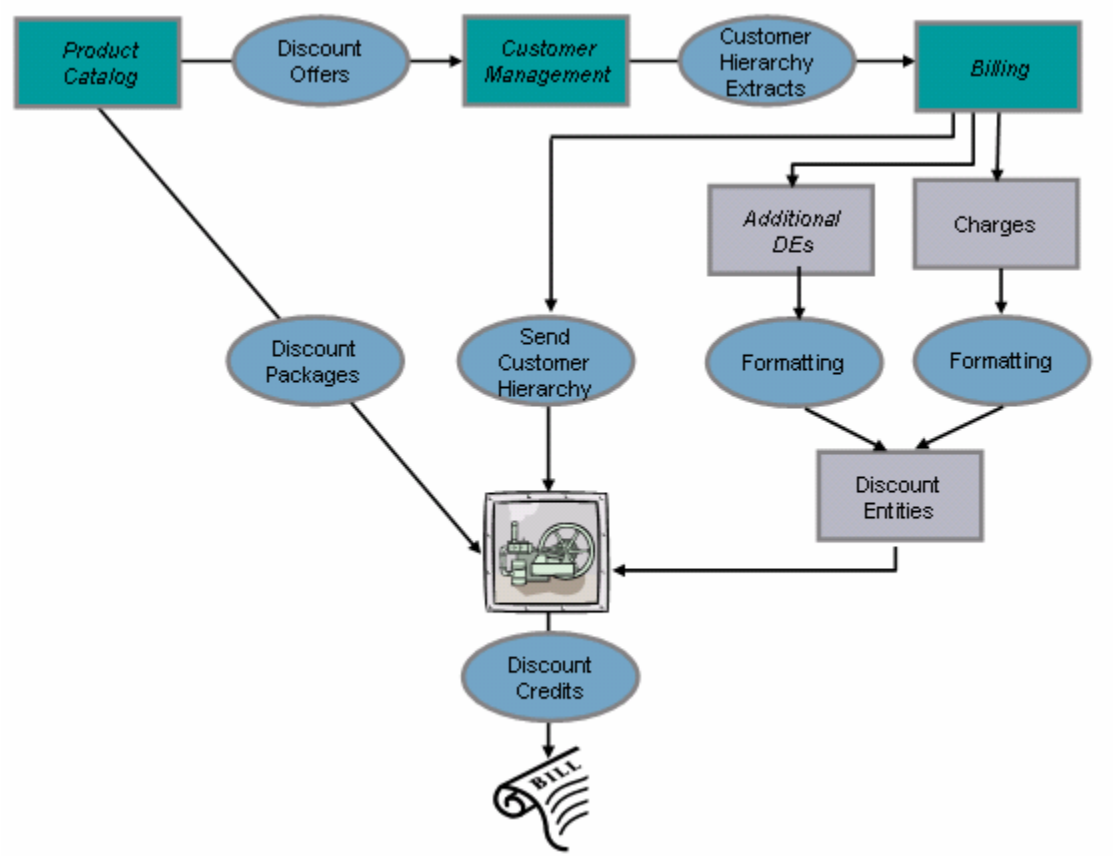


**Figure 9-1: Flow of Discount Entities and Credits**

Each discount entity is checked against all the customer's discount offers, to see if the charge qualifies as an eligible or contributing entity.

The following example is for a discount based on charge dynamic attributes:

If the total amount of charges for the provider, *Amazon*, is greater than $100, give a 20% discount for recurring charges.

The contributing entity is:

Provider ID – Amazon

Since the Provider ID is a dynamic attribute for an *External* charge entity type, the contributing entity has to define a condition on the charge entity type as well:

*Charge entity type* – External

In this way, charges with other CETs do not qualify for this discount. The discount is not applicable to them since they have no Provider ID attribute.

## Prorated Bill Day Discounts

Bill day discounts are prorated to reflect the portion of the billing cycle during which they were effective. The following cases are covered:

- Discount offer becomes active in the middle of the cycle
- Discount offer is discontinued during the cycle but the subscriber is still active at bill day
- The subscriber is cancelled before bill day.

Proration can be performed for the following types of discounts:

- Flat monetary amount – The discount amount is prorated
- Tier/step intervals – The tiers are prorated

  For example:

  A 10% discount is applied once the subscriber has reached $100. Then, if the discount applies to half the cycle for a given subscriber, the discount would be 10% once the subscriber reaches $50.

## Discounts for Prepaid Pay Channels

Discount credits that should be paid by prepaid pay channels are reported to Accounts Receivable and Replenishment Management as part of the Billing Confirmation process.

# Discount Definitions

## Proration Definitions

### Proration Type

Proration type is defined per component as one of the following:

- Prorated
- Not Prorated – Apply only if effective at DE run date
- Not Prorated – Apply if appears during the DE given time range
- Take from Market

If not defined explicitly, proration type is taken from the market default (defined at DE runtime per market). The default proration type at the market level is Not Prorated – apply only if effective on the DE run date.

For each component, the engine allows the user to specify: “Do not apply if plan was attached less than X days” (where X is a positive number).

### Proration Formula

The Proration Factor is calculated based on the time the Plan was attached to the entity (ignoring the exact time of the day) according to one of the following criteria:

- Real days
- Fixed number of days (flat) – The number of days for this formula is defined at the market level. The default is 30 days in a cycle.
- User-defined – Exit point

**Example:**

Plan XYZ was attached on 5/3/00. Cycle dates are 2/3/00 – 2/4/00.

The above formulas will calculate the factor as follows (March has 31 days, April has 30 days):

- According to the Real Days formula: (27 + 2) / 31 = 0.935483
- According to Fixed Number of Days formula: (27 + 2) / 30 = 0.966666 (if the fixed number of days is 30)

Proration formulas are defined per prorated component. If not defined explicitly, the proration formula is taken from the market default (defined at the DE run per market).

### Proratable Properties in Discount Plans

Proration is defined per component for each of the following properties (any combination of the properties may be used):

- Contribute amount
- Eligible amount
- Tiers/Steps From/To
- Amount (Fixed or Percent)
- Maximum

The following section describes all the properties that can be prorated or factorized (i.e., cycle factor) in Discount Plans, and provides an example for each property.

### Proration Factor Examples

In all the examples in this section, the Discount Plan is attached for a quarter of the month, and the Proration factor is 0.25.

#### *Contributing Amount*

The contributing amount is multiplied by the Proration factor.

**Example:**

Original Tiers list:

0 – 100: 10%

100 – 200: 20%

200 – Infinity: 50%

Contributing charges: $160

Prorated contributing charges: $160*0.25 = $40

The first tier is selected, and a 10% discount is given on the eligible charges.

If proration is not used, the second tier is chosen, and a 20% discount is applied.

### *Eligible Amount*

The eligible amount is multiplied by the Proration factor.

**Example:**

Original Flat definition:

Give 20% discount on eligible charges.

Eligible charges: $160

Prorated eligible charges: $160*0.25 = $40

The discount of 40*20% = $8 is given.

If proration is not used, a discount of $32 is applied.

### *Tiers/Steps From/To*

Values of From/To in the Tiers or Steps list are multiplied by the Proration factor.

**Example**:

Original Steps list:

0 – 100: 0%

100 – 200: 10%

200 – Infinity: 20%

Prorated Steps list:

0 – 25: 0%

25 – 50: 10%

50 – Infinity: 20%

For eligible charges of $100, the discount is 25*0% + 25*10% + 50*20% = $12.5.

If proration is not used, the discount is $100 * 0% = $0

### *Amount (Fixed or Percent)*

The amount (either in Tiers/Steps lists or Flat) is multiplied by the Proration factor.

**Example**:

Original Tiers list:

0 – 100: $10

100 – 200: $40

200 – Infinity: $100

Prorated Tiers list:

0 – 100: $2.5

100 – 200: $10

200 – Infinity: $25

Contributing charges: $160

Eligible charges: $50

A discount of $10 is given on the eligible charges.

If proration is not used, the discount is $40.

### *Maximum*

The maximum amount is multiplied by the Proration factor.

**Example**:

Original Discount Plan:

Give a 20% discount on eligible charges with a maximum of $40.

Eligible charges: $100

The discount of 100*20% = $20.

The prorated maximum is $40*0.25 = $10.

The discount is $10.

If proration is not used, the maximum is not applied, and the discount is $20.

## Cycle Factor Examples

In all the examples in this section, the Discount Plan is attached for the whole period of the cycle, and the cycle is bi-monthly, that is, the Cycle Factor is 2.

### *Contributing Amount*

The contributing amount is multiplied by the Cycle factor.

**Example:**

Original Tiers list:

0 – 100: 10%

100 – 200: 20%

200 – Infinity: 50%

Contributing charges: $160

Factorized contributing charges: $160*2 = $320

The third tier is selected, and a 50% discount is given on the eligible charges.

If factor is not used, the second tier is chosen, and a 20% discount is applied.

### *Eligible Amount*

The eligible amount is multiplied by the Cycle factor.

**Example:**

Original Flat definition:

Give 20% discount on eligible charges.

Eligible charges: $160

Cycle factor eligible charges: $160*2 = $320

The discount of $320*20% = $64 is given.

If the Cycle factor is not used, a discount of $32 is applied.

### *Tiers/Steps From/To*

Values of From/To in the Tiers or Steps list is multiplied by the Cycle factor.

**Example**:

Original Steps list:

0 – 100: 0%

100 – 200: 10%

200 – Infinity: 20%

Prorated Steps list:

0 – 200: 0%

200 – 400: 10%

400 – Infinity: 20%

For eligible charges of $300, the discount is 200*0% + 100*10% = $10.

If the Cycle factor is not used, the discount is 100*0% + 100*10% + 100*20% = $30.

### *Amount (Fixed or Percent)*

The amount (either in Tiers/Steps lists or Flat) is multiplied by the Cycle factor.

**Example**:

Original Tiers list:

0 – 100: $10

100 – 200: $40

200 – Infinity: $100

Prorated Tiers list:

0 – 100: $20

100 – 200: $80

200 – Infinity: $200

Contributing charges: $160

A discount of $80 is given on the eligible charges.

If the Cycle factor is not used, the discount is $40.

### *Maximum*

The maximum amount is multiplied by the Cycle factor.

**Example**:

Original Discount Plan:

Give a 20% discount to eligible charges with maximum of $40.

Eligible charges: $300

The discount of $300*20% = $60.

Cycle factor maximum is 40*2 = $80

The discount is $60.

If the Cycle factor is not used, the maximum is applied, and the discount is $40.

## Discount Item Methods

The Discount component is defined by:

- Contributing Discount Entities
- Eligible Discount Entities
- Threshold – Based on the accumulated attributes of Contributing Discount Entities
- Discount Calculation Method – The way the calculation is processed

Subsequent sections describe the supported Discount Calculation methods.

### Flat Discount Method

Only one discount entity is defined. This entity is used as both the eligible and the contributing discount entity. The same discount qualification is applied to both the qualification for threshold accumulation and eligibility for discount.

The accumulated field is always the amount field (mapped to the Static Amount field in the Charge entity).

The discount can be a percentage of the total amount of the eligible items, or a fixed amount.

### Stepped Discount Method

Only one discount entity is defined, which serves as the eligible and contributing Discount element. The same discount qualification is applied to both the qualification for threshold accumulation and eligibility for discount.

The accumulated field is always the amount field (mapped to the Static Amount field in the Charge entity).

For example, a stepped discount defines a percentage to be discounted for each amount range:

| Step | From | To | Percent |
|---|---|---|---|
| 1. | 0 | 300 | 0% |
| 2. | 300 | 450 | 15% |
| 3. | 450 | --- | 20% |

The first 300 amount units yield no discount, the next 150 receive a 15% discount, and the remaining amount (above 450) receives a 20% discount.

For example, if a subscriber spends $500, the discount calculation is:

(450-300)*15% + 50*20% = $32.50

The calculated amount is distributed to the eligible charges based on the relative contribution of each charge.

A stepped discount always defines a percentage for each step, and cannot define fixed amount discount.

### Tiered Discount Method

This method allows separate definition of eligible and contributing elements.

The accumulated field for the eligible element discount qualification criterion) is always an amount field and a DET that must be charged, while the accumulated field for the contributing element can be amount or any other dynamic attribute of the charge (e.g., duration or volume), or any other DET defined in the system.

For example, if the contributing element's accumulated field is GPRS volume, the following table can be defined:

| Tier | From | To | Percent |
|---|---|---|---|
| 1. | 0 MB | 300 MB | 0% |
| 2. | 300 MB | 450 MB | 15% |
| 3. | 450 MB | --- | 20% |

If the total volume is 400 MB, the subscriber receives a 15% discount for all eligible charges. If the total volume is 460 MB, the subscriber receives a 20% discount for all eligible charges.

Tiered discounts can define fixed amounts or percentage for each tier.

All discount methods can define minimum and maximum amounts for the discount. If the calculated discount is smaller than the minimum amount, no discount is given. If the calculated discount is greater than the maximum amount, the discount is changed to the maximum amount.

## Conditional Discount Packages

A conditional discount package is a discount package with multiple contributing rules, where each contributing rule has a separate accumulation. In terms of functionality, it is equivalent to a regular discount package with a single discount item and multiple contributing rules.

Conditional discount packages are composed of one or more discount conditions, where each discount condition defines a separate contributing rule.

**Example:**

If the total usage charge is greater than $300, and the GRPS usage volume is greater than 150 MB, a 10% discount is given on all recurring charges. This is implemented as a conditional discount package with the following definitions:

- Eligible rule: Revenue Type = RC
- Discount amount: 10% with no minimum/maximum
- Two discount conditions defined with OR as the conditional relation
- DET is Charge
- Accumulation attribute is Amount:

- Qualification is Revenue code = UC
- Condition is “from ($) 300”

- Accumulation attribute is Volume
  - Qualification is Revenue code = UC
  - Offer = GRPS, condition is “from 150 (MB)”

## Hierarchy/Subscriber Discount

A discount offer can be attached to a subscriber or unit in the hierarchy.

If the discount offer is attached to a subscriber, the discount is calculated based on the subscriber performance (discount elements related to the subscriber), and the subscriber receives discount credits, one credit per charge, proportional to the charge amount related to the amount of all eligible qualifying charges.

If the discount offer is attached to a unit in the hierarchy, all the charges and other discount entities of the subscribers and units under the unit to which the discount offer is attached, are accumulated as contributing and eligible entities, and the threshold is checked against the performance of charges that relates to the unit’s sub-hierarchy.

In this method too, a credit is created for each charge. Both contribution and eligibility can be restricted to the unit’s own discount entity, or the entire unit sub-hierarchy discount entities.

### Hierarchy Discount Example

Unit A has the following discount offer:

If the total usage charges are greater than $500, a 10% discount is calculated for the recurring charges.

The hierarchy structure is:

- Unit A
  - Unit B
    - Subscriber 1
    - Subscriber 2
    - Subscriber 3

The hierarchy discount charges are:

| | Usage Charges ($) | Recurring Charges ($) | Pay Channel |
|---|---|---|---|
| Subscriber 1 | 100 | 10 | 1 |
| Subscriber 2 | 145 | 30 | 2 |
| Unit B | --- | 40 | 1 |
| Subscriber 3 | 265 | 10 | 2 |
| Unit A | --- | 50 | 3 |

The total usage charges are $100 + 145 + 265 = $510. Therefore, the unit is entitled to a 10% discount on the recurring charges, and the following credits are created:

- Subscriber 1 – $1 credit for Pay Channel 1
- Subscriber 2 – $3 credit for Pay Channel 2
- Subscriber 3 – $1 credit for Pay Channel 2
- Unit B – $4 credit for Pay Channel 1
- Unit A – $5 credit for Pay Channel 3

Each credit is related to an eligible charge. Therefore, each credit is distributed to the pay channel to which the original charge relates.

# 10. INVOICING

Invoicing scans all billing arrangement charges received from Charge Creation. For postpaid charges, it invoices the charges, calculates taxes, and creates invoices. For prepaid charges, it relates the charge to the prepaid route, calculates the tax based on the system policy, and generates the statements.

Invoicing has an accumulation utility that enables it to accumulate charges based on the configuration defined in Billing Configurator. These accumulations can be used by the Bill Layout extract, reports, etc.

When a Charge's currency is different from the BA currency, Invoicing converts the charge amounts to the amount in BA currency.

The Rate Effective Date for currency conversion is configurable through the Billing Configurator. The Rate Effective Date can be set to one of the following values:

- **Charge Effective Date**: The exchange rate is applied on the day that the charge is made.
- **Cycle Close Date**: The exchange rate is applied on the day that the cycle is closed.
- **Cycle Run Date**: The exchange rate is applied on the day that Billing is run. If Billing is re-run after a Billing Undo, the cycle run date is the same as the date on which Billing was run for the first time for that cycle instance.

The original charge in the offer currency is maintained in the Charge table (with internal digit precision). The converted charges in the BA currency are maintained in the Invoice Charge Relation table (with external digit precision). The accumulation utility can be configured to use either one of the above currencies (i.e., offer currency or BA currency).

## Tax Calculation

This section provides information regarding the various types of tax calculation.

### Tax Code

Each charge code has a Tax Code attribute. Each tax code uses different policies regarding rates (for Internal Tax method) and for exemptions.

### Tax Type

Tax Type determines the type of tax applied in the country. While the tax code is a Billing definition, the tax type refers to the tax outside the system. For example, in Europe, VAT is the tax type; in America, federal tax and state tax are both tax types.

Each tax code refers to one tax type, and many tax codes can refer to the same tax type.

## Tax Calculation Method

Billing supports two tax calculation modes:

- Internal Tax Calculation – Based on tax rate tables defined in Billing Configurator. This mode is used for simple tax rules (for example, European CSPs).
- External Tax Calculation – The calculation is executed using an external tax package. This mode is used for complex tax rules (for example, American CSPs).

It is possible to define the Internal Tax Calculation rules for specific tax codes, and External Tax Calculation for other tax codes. Each tax code may use separate external package.

### Internal Tax

The internal tax calculation is based on rules defined in Billing Configurator.

The tax rate can depend on customer attributes, e.g., on the district of the subscriber residence. The tax rate is defined in the following way:

| Tax Code | Tax Rule | Tax Authority | Tax Type | Tax Rate |
|---|---|---|---|---|
| TX1 | Zone (Service Receiver) = 1 | Government | VAT | 8% |
| TX1 | * | Government | VAT | 10% |
| TX1 | * | Government | Internet Tax | 15% |
| TX1 | Customer Type (Payer) = Senior | Government | Internet Tax | 2% |
| TX2 | * | Government | VAT | 10% |

TX1 is the tax code for Internet usage charges. Each Internet usage charge is subject to two types of tax – special Internet tax and VAT.

Subscribers who live in Zone 1 pay only 8% VAT, while all others pay 10%. Zone 1 rules refer to the service receiver, i.e., if a subscriber lives in Zone 2 and pays bills via a billing arrangement that resides in Zone 1, the higher tax (10%) is paid.

Internet tax is 15%. However, senior citizens should pay only 2%. This rule refers to the payer, so if a senior citizen pays the Internet usage for a grandchild, only 2% tax is paid. TX2 is the tax for recurring charges, and there are no special rules for it.

If the customer (subscriber or pay channel) for a specific tax code does not match one of the rules, the default rule is always "*".

The tax code is an attribute of the charge code.

For cases where tax is very simple, with no special rules, the tax rate is defined in the following way:

| Tax Code | Tax Rule | Tax Authority | Tax Type | Tax Rate |
|---|---|---|---|---|
| All | * | Government | VAT | 8% |

All charges have the same tax rate.

Each tax rate has an effective and expiration date, and each tax is calculated considering the charge date.

### Itemized Tax Indicator

Tax can be calculated for each charge separately, or for all the charges with the same tax type, tax rate, and invoice type.

**Example of itemized and non-itemized tax:**

Itemized Tax

| Charge Amount | Tax Type | Tax Rate | Tax Amount |
|---|---|---|---|
| 21 | VAT | 8% | 1.68 |
| 33 | VAT | 8% | 2.64 |

Non-itemized Tax

| Charge Amount | Tax Type | Tax Rate | Tax Amount |
|---|---|---|---|
| 21 | VAT | 8% | --- |
| 33 | VAT | 8% | --- |
| 54 | VAT | 8% | 4.32 |

### External Tax Packages

External tax calculation is based on tax packages, usually for the American market.

It is possible to define a different tax package for each tax code.

The interface with the tax package is defined in the customization layer. The core module prepares the relevant information for the tax package, and provides an exit point to the customization layer for the tax package interface.

The exit point receives the following parameters:

- The name of the tax package to interface with
- Amount of charge
- Tax code
- Charge effective date
- Customer and service tax exemptions
- Customer attributes required by the tax package (usually, address entities); the attributes are defined as explained in the *Using Customer Attributes* section.

The exit point maps the customer attributes to the tax package attributes (e.g., tax packages may need a special code for each state, while billing sends the state name), invokes a call to the tax package, and returns the results to Billing core.

Tax can be calculated for each charge separately, or for a group of charges. The need to calculate tax for a group of charges is for special rules for specific services. For example, Internet charges in Texas are exempt of tax for the first $25. The core module aggregates the entire Internet charges for the

state of Texas, and sends them as one transaction to the exit point for the tax calculation. The way to define a group of charges for the tax package is explained later in the *Dynamic Accumulation* section.

## Tax Exemption

Tax exemption can be related to a specific customer or to specific services.

### Customer Exemption

Customer tax exemptions are defined using Billing APIs.

A customer or subscriber can be exempted from all taxes or from specific tax types.

Exemptions can be defined at the service receiver level or at the payer level. The Billing policy determines which exemption to check (the payer exemption or the service receiver exemption).

Customer attributes of either the payer customer and BA, or receiver customer and subscriber (as defined by the market policy for tax exemption), which are required by the tax package (usually, address entities), are defined as explained in the *Using Customer Attributes* section.

### Service Exemption

Service tax exemptions are defined in Billing Configurator by exempting a specific tax code.

There are two levels of tax code exemptions:

- Exemption of tax code – All charges with this tax code are not subject to tax
- Exemption of specific tax types – Tax codes can be exempted only from specific tax types, or form taxes for specific tax authorities; for example, Internet services carry no state tax

In some cases, the tax exemption is calculated by the tax package. In such cases, tax exemptions should not be defined in Billing Configurator.

## Tax Qualifiers

Each tax item has tax qualifiers, which are the customer attributes affected by the tax calculation, for example state or county.

In External Tax Calculation mode, the tax qualifiers are determined by the tax package and should be returned by the exit point.

In Internal Tax Calculation mode, the tax qualifiers are determined by the tax rule (such as Zone = 1).

## Tax Rate Changes

A tax rate change affects new charges created after the tax change. However, it has a different effect on the charges created during the cycle period in which the tax change occurred. There is a different approach for external and internal tax calculation.

### External Calculation

One-Time, Accumulated Usage, and Recurring Charges – Each charge is sent to the tax package with the charge effective date. The tax package then calculates the tax using the correct tax rate.

### Internal Calculation

- One-time Charges – Each charge is sent to the tax package with the charge effective date, and the tax is calculated based on the tax rule effective on that date.
- Recurring Charges – Billing acts according to the policy defined in Billing Configurator:
  - If the policy is to prorate the tax rate for recurring charges – The tax rate is prorated according to the tax change date and customer attribute changes.

    For example, if the tax rate was changed from 5% to 10% in the middle of a cycle period, half of the RC amount is calculated with 5%, and the other half – with 10%. Same applies to changes in the customer address (i.e., moving state).
  - If the policy is not to prorate the tax rate for recurring charges –The tax is calculated with the tax rate at the cycle close date.
- Accumulated Usage Charges – Product Catalog is notified about the tax change date manually.

  Rater breaks the accumulated charges into two different accumulations. The first includes accumulated charges that occur up to the change date, and the second includes accumulated charges that occur after the change date.

  Each accumulated usage charge is received with an effective date within the period covered by the accumulated usage. This is the date used for the tax calculation.

## Tax and Charge Rounding

Billing eliminates the discrepancies that can arise when the same tax rate is applied to both a series of itemized charges and the non-itemized sum thereof.

The following algorithm is used to ensure that the sum of itemized tax amounts always equals the non-itemized tax. The rounding error for each charge (i.e., the difference between the actual charge calculated and the rounded charge displayed on the bill) is accumulated, and is added to the next charge. The same is done for each tax item amount (same tax type, authority and rate).

**Example:**

| Actual | | No Correction | | Correction Handling | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Charge | Tax | Display Charge | Display Tax | Adjust Charge | Display Charge | Error Charge | Adjust Tax | Display Tax | Error Tax |
| 6.86980 | 1.1678660 | 6.87 | 1.17 | 6.86980 | 6.87 | -0.0002 | 1.1678660 | 1.17 | -0.002134 |
| 8.37827 | 1.4243059 | 8.38 | 1.42 | 8.37807 | 8.38 | -0.00193 | 1.4221719 | 1.42 | 0.0021719 |
| 5.14652 | 0.8749084 | 5.15 | 0.87 | 5.14459 | 5.14 | 0.00459 | 0.8770803 | 0.88 | -0.0029197 |

| Actual | | No Correction | | Correction Handling | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Charge | Tax | Display Charge | Display Tax | Adjust Charge | Display Charge | Error Charge | Adjust Tax | Display Tax | Error Tax |
| 6.08711 | 1.0348087 | 6.09 | 1.03 | 6.09170 | 6.09 | 0.0017 | 1.0318890 | 1.03 | 0.001889 |
| 3.65350 | 0.6210950 | 3.65 | 0.62 | 3.65520 | 3.66 | -0.0048 | 0.6229840 | 0.62 | 0.002984 |
| 3.17135 | 0.5391295 | 3.17 | 0.54 | 3.16655 | 3.17 | -0.00345 | 0.5421135 | 0.54 | 0.0021135 |
| 7.96042 | 1.3532714 | 7.96 | 1.35 | 7.95697 | 7.96 | -0.00303 | 1.3553849 | 1.36 | -0.0046151 |
| 1.42632 | 0.2424744 | 1.43 | 0.24 | 1.42329 | 1.42 | 0.00329 | 0.2378593 | 0.24 | -0.0021407 |
| 0.00591 | 0.0010047 | 0.01 | 0.00 | 0.00920 | 0.01 | -0.0008 | -0.0011360 | 0.00 | -0.001136 |
| 0.17583 | 0.0298911 | 0.18 | 0.03 | 0.17503 | 0.18 | -0.00497 | 0.2875510 | 0.03 | 0.257551 |
| 8.94662 | 1.5209254 | 8.95 | 1.52 | 8.94165 | 8.94 | -0.0002 | 1.5196805 | 1.52 | -0.002134 |
| TOTALS: | | | | | | | | | |
| 51.82165 | 8.8096805 | 51.84<br>X 0.17<br>8.8128 | 8.79 | | 51.82<br>X 0.17<br>8.8094 | | | 8.81 | |

1. Adjust (1st line) = Actual (1st line)
2. Display = Adjust, rounded
3. Rounding error (not shown) = Adjust– Display
4. Adjust = Actual + Rounding error for the previous line

In the example above, note that when correction handling is not applied, the total of the displayed charges is 2¢ less than the actual total. The total of the displayed tax amounts is also 2¢ less than both the actual tax and the tax calculated on the total of the displayed charges. (In this example, the tax rate is 17%.)

# Dynamic Accumulation

During invoicing, Billing scans the charges, and accumulates them into groups. Postpaid charges are always accumulated into Total Invoice Amount.

## Accumulator Attributes

Billing provides the option to total charges into different groups according to criteria defined in Billing Configurator.

The implementation team determines the attributes by which the charges are accumulated, and the fields to accumulate the charges. Billing groups all charges with the same value of the Accumulated By attribute, and accumulates the charges according to the accumulated attributes. The accumulated attributes can be charge attributes, pay channel attributes, tax attributes that relate to the charge, or customer attributes for customers to which the charge relates (as a service receiver or payer).

It is possible to provide values for a number of Accumulated By fields, while others do not contain values.

### Accumulated by Charge Attribute

Any charge attribute can be selected, including:

- Pay Channel
- Charge Code

- Service Receiver ID
- Offer
- Pricing Item
- Dynamic Attribute – Select a CET, and then select one of the dynamic attributes of that CET

An example of accumulation based on a charge dynamic attribute is:

- Accumulated by – Provider
- Accumulated field – Amount

All charges are accumulated per provider. The result is total charges per provider that charges the billing arrangement. However, the Provider field may be populated only in third-party charges (a special CET); other charges will not have the Provider field. These charges will be accumulated to an accumulator with empty Provider.

It is possible to populate the accumulated field with a value, and accumulate only those charges with the specified value. In the previous example, the Accumulated By field can be determined as "Provider – Amazon". In this case, only charges for Amazon will be accumulated. All other providers will not be accumulated.

### Accumulated by Tax Attribute

The following fields can be defined:

- Tax Type
- Tax Code
- Tax Authority
- Tax Rate

The Accumulated by Tax attribute is used for non-itemized tax calculations (other attributes from the charge are also considered), so all charges with the same tax type and tax rate should be grouped for tax calculation. The tax accumulation is done also by invoice type of the charge, which represents the breakdown of the invoice in Accounts Receivable, to enable reporting of taxes per invoice.

For an external tax package, any accumulation can be configured via Billing Configurator.

### Accumulated by Customer Attribute

Any defined customer attribute can be selected. In addition, the implementation team needs to define if the customer attribute refers to a payer or to service receiver.

If a customer attribute refers to the payer, it must be accumulated at the customer or billing arrangement level. A customer attribute that refers to the service receiver must be accumulated at the customer or subscriber level.

Examples of accumulation based on a customer attribute include:

- Accumulated by – State
- Accumulated field – Amount

The billing arrangement charges are grouped and summarized by state for tax purposes (different tax rules for every state).

### Use of Dynamic Accumulations

Dynamic accumulations can be used for the following purposes:

- Reports:
  - Total duration for all roaming charges
  - Total volume for GPRS charges
- Information for the bill layout utility:
  - Total charge for each subscriber
  - Total charge for each pay channel
- Tax – Accumulate charges to be taxed together, for example charges of a specific group that have rules depending on the total amount for charges.

When the accumulation is defined for tax purposes, the tax module knows that a specific charge is part of a group, and does not send it to the tax module. Instead, it uses the accumulation mechanism, and sends the total amount to the tax module.

## Invoices

All postpaid charges and their taxes relate to an invoice.

### Total Amount Due Rules

Rules can be defined for the total invoice amount in the core layer and in the customization layer.

Currently, the core layer supports Zero Balance Indicator, which is usually set for internal customers. This ensures that if the total amount due on the invoice is not zero, a credit or a debit is created to zero the balance. The information about whether a customer's balance should be zero is extracted from Customer Management (usually, derived from the customer type), and is stored in the BA entity.

Rules can be defined for the total invoice amount in the customization layer. The system provides an exit point to apply rules to the total amount due in the customization layer. The exit point manipulates the invoice total amount, and returns a buffer of charges (debits or credits).

The returned charges are invoiced, taxed, and included in the charge accumulation. The invoice total amount is then updated with these charges.

An example of charges that can be created in the customization layer using this exit point is as follows. Some CSPs encourage their customers to pay by direct debit. These CSPs add a special fee for customers that pay in cash, but only if the amount due is greater than a predefined value. The value and payment method are checked in the customization layer, which decides

whether a charge is created, and determines the charge amount according the specific CSP requirements.

## Prepaid Statements

Invoicing creates a prepaid statement for each prepaid pay channel of the BA, whether or not the charges were generated by the Bill Preparation process.

Prepaid charges are used to prepare a prepaid statement for each pay channel.

# 11. DOCUMENT CREATION

Billing produces a document for each billing arrangement. Each document is composed of one or more of the following statements:

- Invoice statement – For invoice receivers
- Bill statement – For bill receivers
- Prepaid statements – For prepaid pay channels

There are five types of documents:

- Document with one invoice statement
- Document with one bill statement
- Document with one or more prepaid statements
- Document with one invoice statement and one or more prepaid statements
- Document with one bill statement and one or more prepaid statements

The relations between invoice, statement, and document are illustrated in Figure 11-1.

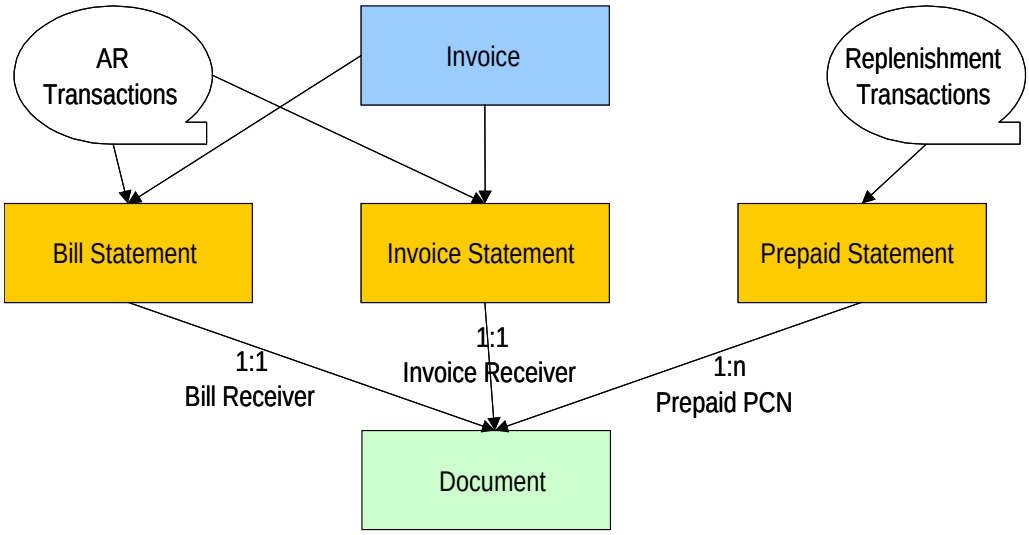


**Figure 11-1: Document Types and Their Relations**

## Document Production Type

The document production type can have one of the following values:

- First document
- Regular document
- Final document – A document for a cancelled billing arrangement active in the last cycle.

- Revised Final document – A document for a cancelled billing arrangement that already received a final document, but for which additional charges or finance activities were received in the last cycle period.
- Dropped – A document for a cancelled billing arrangement that already received a final document, and for which no charges or financial activities were received in the last cycle period. *Dropped* documents are not extracted to the bill layout utility.

# Statements

## Bill Statement

A bill statement is prepared for all bill receivers who need to receive a statement. A bill statement contains:

- Previous Balance
- Financial Activities – Payments and adjustments
- Invoice Amount
- Total Amount Due – Amount to be paid.

The bill statement is based on:

- The previous balance calculated in the last cycle for the billing arrangement
- The invoice amount calculated by the Invoicing process
- The extract of financial activities from Accounts Receivable

### Financial Activities

Financial activities are extracted from Accounts Receivable; they include:

- Adjustments
- Payments

All the financial activities that affect the balance are defined in Billing Configurator with the impact on the balance (increase or decrease), and with text that describes the activity (the text is extracted to the bill layout utility to be displayed on the bill). Any financial activity received from Accounts Receivable should contain the following information:

- Financial activity
- Date
- Amount

Billing takes the previous balance of the billing arrangement, calculates the financial activity impact, and adds the invoice amount to get the *Total Amount Due* on the bill.

Billing sends to Accounts Receivable the date range for these activities to ensure that no activity is lost or reported twice.

### Balance Reconciliation

Accounts Receivable sends to Billing the Accounts Receivable balance of the account that should be equal to the previous balance plus the financial activities calculated in Billing. If the amounts are different, Billing acts according to the policy defined in Billing Configurator:

- Ignore – No reconciliation is performed
- Warning – AMC displays a warning.
- Reject – Reject the bill if the reconciliation fails


*The Accounts Receivable balance relates to an account, while* Total Amount Due *refers to a billing arrangement. Since each account may have only one Bill Receiver billing arrangement, the billing arrangement balance is the account balance as well (Invoice receivers have no* amount due*).*

## Invoice Statement

An invoice statement contains the following information:

- Invoice total amount
- Invoice total tax

Accounts Receivable extracts overpayment information to Billing if the overpayments are not applied automatically to cover other invoices (according to the policy set in Accounts Receivable).

Overpayments are taken into account by Billing. They decrease the invoice amount due.

## Statement Total Amount Due Rules

Some rules might be defined for the total amount due after applying the financial activities. The system provides an exit point for the customization layer to handle the Total Amount Due rules. The exit point can manipulate the invoice total amount; it returns a buffer of charges.

The exit point acquires the following parameters:

- Statement Type (Invoice or Bill)
- Total Amount Due
- Payment Method

The exit point returns a buffer of charges and an amount due handling indication.

The charges created at this point are not taxed, and they do not participate in the charge accumulations. The charges are associated with the statement and update the total amount due of the statement. In this case, the total amount due of the statement is different from the total invoice amount.

Following are some examples of how the customization layer of this exit point may be utilized:

- If the statement amount due is under a predefined minimum amount, the CSP may decide to mark the statement as Not to Pay.
- If the invoice statement amount is under a predefined minimum amount, the CSP may decide to add the amount to the next invoice period. In thus case, the customization layer should:
  - Create a single credit amount for the current invoice with the invoice total amount, or multiple credit amounts for each charge in the invoice. The credit causes the invoice amount to become zero.
  - Create a single future charge for the next invoice with the total invoice amount, or multiple future charges for each charge of the current invoice.
- If the bill statement amount is below a predefined minimum amount, there is no need to create charges or credits. Instead, the bill statement indicates that the amount due is carried over to the next bill, and Accounts Receivable does not calculate the late payment charge or print a message on the bill. In such a case, the customization layer updates the amount due handling attribute of the statement.

# 12. QUALITY ASSURANCE

Quality Assurance is a dynamic auditing tool used to monitor the billing process. The QA team defines the QA population using Billing Configurator and, at the end of the Bill Preparation flow, produces reports and extracts for the bill layout utility.

The QA selection is for billing arrangements, where the QA-generated product is a document that the billing arrangement eventually received, or for customers, where the QA product is the customer charges report.

The report is used to audit customers, where the service receiver distributes its charges to the billing arrangements of different customers, and even in different cycles.

The report displays all the charges created for the service receivers of the customer. This prevents a situation where the QA team checks a document and finds problems in the charge creation part for the receivers, when the service receiver is already confirmed in the previous cycle. In such a situation, it is impossible to undo the charge creation for the service receiver.

## QA for Billing Arrangement Documents

### Population Selection Definition

The implementation team can define a set of criteria for the QA population by assigning the criteria to the required cycle code, and by setting the required population size for each assignment.

A criterion definition is a set of conditions on the customer, produced document, or invoice. Each billing arrangement that matches all the conditions is a candidate for the QA population.

The QA criteria can apply to a specific cycle code or all cycle codes.

Conditions can be defined on one of the attributes described in subsequent sections.

#### Billing Arrangement Attributes

- Itemized Tax (Y/N)
- Zero Balance Indicator (Y/N)
- Document Type – Available values are:
  - Invoice
  - Bill
  - Bill + Prepaid
  - Invoice + Prepaid
  - Prepaid

### Document Attributes

- Production Type – One of the following document types:
  - Regular document
  - Final document
  - First document
  - Revised final document

### Bill Statement Attributes

Available only for billing arrangements with a bill statement: Total due amount range.

### Invoice Statement Attributes

Available only for billing arrangements with an invoice statement: Total due amount range.

### Invoice Attributes

- Total amount range
- Total tax amount range

### Pay Channel Attributes

- Payment Category – Postpaid or Prepaid
- Payment Method – Available values are:
  - Direct debit
  - Cash
  - ATM
  - Credit card
  - Check

### Tax Attributes

- Tax Type
- Tax Authority
- Exemption Type – Service or Customer

### Customer Attributes

One of the attributes defined in Billing Configurator, for example:

- Price Plan
- Customer Type

An example of QA population selection that can be defined using Billing Configurator is:

- Criteria 1 – 10 billing arrangements:
  - Price Plan PP1 – At least one of the subscribers that distribute its charges to the billing arrangement has the PP1 price plan
  - Customer Type – Residential
  - Document Type – Invoice Receiver
  - Invoice Amount Range – $100-300
- Criteria 2 – 5 billing arrangements:
  - Customer Type – Business
  - Invoice Amount Range – $1000-5000

### Selecting a Specific Billing Arrangement

The QA team can ask to add specific billing arrangements to the QA population by providing the billing arrangement numbers.

This is done for a specific cycle run, and can be defined in addition to the QA population defined for the cycle, or as a replacement to the defined QA population.

## Population Selection Process

The QA population is selected during the Bill Day flow by the following processes:

- Invoicing – Selects population based on the invoice, tax, and customer information
- Document Creation – Selects population based on the statement and document

Each process checks if the billing arrangement matches one of the criteria defined for QA. If yes, the process adds the billing arrangement to a QA population list with the IDs of the criteria that the billing arrangement matched.


*A billing arrangement can match more than one criterion.*

The QA Initiator process checks the QA population list and prepares the final QA population by:

- Removing billing arrangements that were not successfully completed
- Providing an exit point that gets the entire QA population, and that can remove part of the population from the list, e.g. remove customers based on the number of their matched criteria.
- Removing billing arrangements if a specific criterion has more billing arrangements than the required population, so that the number of billing arrangements matches the defined population.

The QA Population flow is described in Figure 12-1.

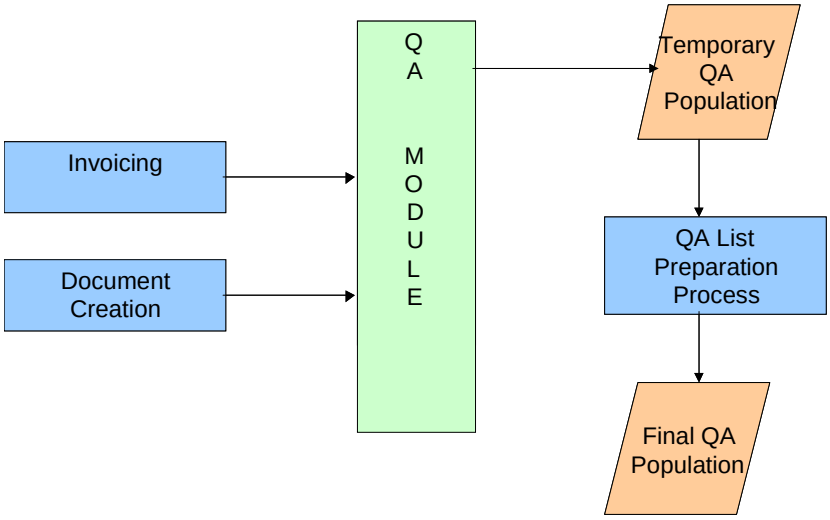


**Figure 12-1: QA Population flow**

## QA Products

The QA products for billing arrangements are the extracts for the bill layout utility, as defined in Billing Configurator. For more information about Bill Layout extracts, see the *Bill Layout Extracts* section.

The process extracts information only for the QA population.

# QA for Customer Documents

## Population Selection Definition

The implementation team can define a set of criteria for QA populations and, for each of the defined criteria set the required population size.

A criterion definition is a set of conditions on the customer. Each customer matching all the conditions is a candidate for the QA population.

Conditions are defined on one of the attributes described in subsequent sections.

### Customer Attributes

- One of the attributes defined in Billing Configurator; for example:
  - Price Plan
  - Customer Type
- Customer attributes maintained by Billing, such as:
  - Change Cycle – Before Change Cycle or After Change Cycle
  - Customer Currency

An example of QA population selection that can be defined using Billing Configurator is:

- Criteria 1 – 10 customers:
  - Price Plan – PP1 – Service receivers with the PP1 price plan
  - Customer Type – Residential
- Criteria 2 – 5 customers
- Price Plan – PP2

### Selecting a Specific Customer

The QA team can ask to add specific customers to the QA population by providing the customer IDs.

This is done for a specific cycle run, and can be defined in addition to the QA population defined for the cycle, or as a replacement to the defined QA population.

## Population Selection Process

The customer QA population is selected by the QA Initiator Population process. The QA Initiator process:

- Selects customers that match the QA criteria related to customers
- Moves over the QA population list and prepares the final QA population
- Removes customers that were not completed successfully
- Provides an exit point that gets the entire QA population, and that can remove part of the population from the list, e.g. remove customers based on the number of the criteria they matched
- Remove customers if a specific criterion has more customers than the required population, so the number of service receivers matches the defined population

## QA Products

The QA product for a customer is the Charges report. This report displays all the charges of a customer created by the Bill Day flow. The data is prepared in ASCII format, which enables importing this data into another application, such as MS Excel, for further treatment.

# 13. BILL LAYOUT EXTRACTS

The bill layout utility is responsible for formatting the layout of the bill and sending it to the print shop for printing.

## Extract Sources

The information extracted for the bill layout utility is usually maintained separately in the following components:

- Customer Management – Customer information, such as names and addresses
- Billing – Charges, taxes, invoices, etc.
- Accounts Receivable – Payments, adjustments, refunds, etc.
- Rating – Event extracts and non-charge PI extracts (allowances)

Billing is responsible for managing and operating the various extracts from the different components. Billing works on the same customer groups, and extracts the information during the same process flow, although each extract belongs to a different component.

## Billing Messages

### For Your Information Messages

For Your Information (FYI) messages are marketing messages CSPs send with documents to customers with specific characteristics. Regular FYI messages appear on the bill document itself, while Page Insert messages appear on a separate sheet.

Examples of FYI messages are:

- A welcome message for billing arrangements receiving their first bill
- An upgrade campaign for billing arrangements whose subscriber(s) carry a specific price plan
- A note on new tax rates for billing arrangements subject to a specific tax type

#### FYI Message Definitions

The marketing team uses Billing Configurator to define FYI messages and the criteria used to identify the billing arrangements that should receive them.

Billing Configurator stores the messages and their qualification criteria in the Billing reference tables, which can be accessed by a customized process and applied to the appropriate bills.

Subsequent sections describe the qualification criteria that can be defined using Billing Configurator.

### Billing Arrangement Attributes

- Zero Balance Indicator (Y/N)
- Document Type – Available values are:
  - Invoice
  - Bill
  - Bill + Prepaid
  - Invoice + Prepaid
  - Prepaid

### Document Attributes

- Production Type – One of the following document types:
  - Regular document
  - Final document
  - First document
  - Revised final document

### Bill Statement Attributes

Available only for billing arrangements with a bill statement: Total due amount range.

### Invoice Statements Attributes

Available only for billing arrangements with an invoice statement: Total due amount range.

### Invoice Attributes

- Total Invoice amount range
- Total tax amount range

### Tax Attributes

- Tax Type
- Tax Authority
- Exemption Type – Service or Customer

### Customer Attributes

- One of the attributes defined in Billing Configurator, for example:
  - Price Plan
  - Customer Type
- Customer attributes maintained by Billing, such as “Change Cycle Request – Before Change Cycle or After Change Cycle”.

Any message can be defined with a set of dynamic attributes and their values, such as message position and electronic code. Since the attributes have no impact on Billing; they are defined during implementation.

## Charge Messages

Charge messages appear next to charges; they are extracted to the bill layout utility with each charge.

The implementation teams define a message for each charge code in Billing Configurator. The message usually describes the meaning of the charge, e.g., “voice calls” for a voice call charge code.

# Extract Definitions

All extract definitions are defined in Billing in two phases described in subsequent sections.

### Complex Entity Definitions

For entities that comprise of one or more tables, extract definitions are defined in XML format in the Billing reference tables, and contain the list of attributes in both the complex entities and the SQL query that creates them (e.g., joined tables). The extract groups are created with Extract Initiator (joined with cycle population, which has an attribute for each customer and billing arrangement that relates them to an extract group). The complex entities and dynamic attributes are parsed, and presented as regular simple attributes. Complex entities can be defined in the same way for Customer Management or Accounts Receivable by their respective implementation teams.

### Extract Definitions

Based on the complex entity definitions, the implementation team can use Billing Configurator to define the attributes to be extracted from each complex entity, and the extract file format.

Billing supports the following record types for the extract output definitions:

- Position – Each attribute is located at a specific position (start position and end position)
- Delimited – A predefined character separates attributes; the attribute order in the file is predefined
- Name and Value – The record is composed of attributes in the *name=value* format, with a predefined character separating the attributes

Billing supports files with single and multiple record structures. In a multiple record structure, a key attribute must be defined. This key attribute determines the record type.


*Complex entity definitions should be handled by an implementation team or professional services with a good knowledge of the application data model. Extract definitions in Billing Configurator can be made by a business implementation team.*

### Reference Data Extract

During complex entity creation, the implementation team can indicate a complex entity as a *reference*, and then base the complex entity on the reference data information.

Extracts that are based on complex entities are run in a separate process, not on a specific customer population, ignoring the customer groups created by Extract Initiator.

An example of reference extracts is a charge code with message codes, and message codes with message text.

### Event Processing Extract Definitions

Event and PI extracts are defined in Product Catalog, not in Billing Configurator. For more information, see the Product Catalog and Event Processing Specification documents.

The Event Processing extract is a separate process, which is controlled by the Bill Layout process extracts, and which works on the same customer groups as all other extracts.

The extracted events and non-charge PIs are for all service receivers that distribute their charges to one of the billing arrangements in the running cycle, whether they are in the same cycle or in a different cycle.

The Event Processing extract gets a list of service receivers and billing arrangements in their cycles, and knows how to extract the information from the various Rater instances (since different cycles may run on different machines).

## Editing Extract Scripts

The extract definitions in Billing Configurator are generated as an XML-based extract script for the Billing Extract process.

An expert user can edit this script to add the following features that cannot be generated by Billing Configurator:

- Additional queries not based on complex entities.
- Calls to external functions – C++-based external functions can be called from the script for additional manipulation of the extracted information. The information returned from the external function is then extracted or used as a base for another query.

## Enhanced Bill Redirection (Printing Categories)

A configurable subset of the generated bills can be redirected to multiple destinations (not only to the CSR). The redirection functionality includes the following features:

- Bills can be assigned (using a set of APIs) to a variety of configurable categories for redirection. They can be assigned manually or automatically (using an exit point).

- A wide variety of destinations are possible, including email addresses, phone numbers (via SMS), network printers, and destinations for special (e.g., international) postal handling.
- Bills can be redirected on a permanent or temporary basis. If temporary, effective and expiration dates can be applied.
- A redirected bill can be marked such that its regular paper bill is suppressed or redirected.

# 14. BILL UNDO

Undo enables the CSP to cancel all or part of the Bill Preparation after it has been run but prior to Confirmation. Undo rolls back the Bill Preparation process to the point prior to where the errors were introduced, and sets the entity and cycle statuses accordingly. It is performed when either a set of specific entities or the entire cycle is rejected – typically as a result of the QA process.

This chapter describes the conditions required, and the results achieved, by the Undo process. For information on the individual processes that make up Undo, see the Undo Flow section.

## Undo Population

Undo can be requested either for specific entities (customers and/or billing arrangements) or for the entire cycle:

- Undo to Customer – The customer has an undo request type, which is set by the business user through the AMC Cycle Management screens. Set for either a specific set of customers or for the entire cycle, the undo request type can be either Full Undo or Full Undo with Rerate, indicating the type of Undo needed for the customer.
- Undo to Billing Arrangement – A billing arrangement has an undo request type, which is set by the business user via the AMC Cycle Management screens. Set for either a specific set of customers or for the entire cycle, the undo request type can be either Undo Document or Undo Invoice, indicating the type of Undo required for the billing arrangement.
- Undo the entire cycle – If the entire cycle is marked for Undo, all customers and billing arrangements in the cycle are marked for Undo. When more complex selection criteria are required, IT professionals can prepare a population list for Undo

When the entity is undone, it is considered as manually rejected, and the error description contains the undo reason that can be used as one of the rerun criteria.

## Undo Types

There are several types of Undo, each rolling back the cycle or set of entities to a different stage of the Bill Preparation process. Following is the list of Undo types (in order of increasing severity):

- Document – The created document is deleted, and the Bill Preparation flow is rolled back to the point immediately after the Tax & Invoicing. This is done for billing arrangements only.
- Invoice – Both document creation and invoice data are deleted, and the Bill Preparation flow is rolled back to immediately after Charge Preparation. This is done for billing arrangements only.

- Full – Everything is undone, including Initiator, Extracts, Change Cycle, and Charge Preparation. If Full Undo is requested for either a customer or its billing arrangement, it is performed for both.
- Full with Rerate – Same as Full Undo, except that the customers in the group are also directed to the Rerate flow.

# Prerequisites for Undo

The following conditions must be met in order for Undo to be performed on the charges of a Customer:

- The customer entity must not yet have been confirmed.
- The associated billing arrangements must not yet have been confirmed, and must be successfully marked for Undo.
- If the customer's charges are distributed to a different billing arrangement (whether in the same cycle or a different one), the status of that billing arrangement must be UC (Unified Charges) or earlier, meaning that the charges have not yet been billed to that billing arrangement.

The following condition must be met in order for undo to be performed on a given billing arrangement:

The billing arrangement must not yet have been confirmed.

# Undo Functions

The Undo process performs the following functions:

- Deletes from the tables the data written after the rollback point
- Resets entity and cycle statuses (to the rollback point)
- Clears the Format Extract Date (date on which extract to Bill Formatter was performed)
- Re-invokes the undone processes (optional)


*When the Undo process is performed, the entities undone are considered as manually rejected, and the error description contains the undo reason.*

Subsequent sections describe the tables and statuses affected.

## Invoice Undo

The following tables are deleted:

- BL1_BILL_FINANCE_ACTIVITY
- BL1_BILL_STATEMENT
- BL1_INVOICING_STATEMENT
- BL1_PREPAID_STATEMENT
- BL1_DOCUMENT
- BL1_TAX_ITEM
- BL1_TAX

- BL1_CHARGE (only those charges that originated in Invoicing)
- BL1_INV_CHARGE_REL
- BL1_CHARGE_ACC
- BL1_INVOICE
- BL1_CYC_QA_POP

In addition, the entity status is set to Unified (to indicate that billing processes have been completed up to, but not including, Invoicing).

## Full Undo or Undo with Rerate

In addition to the billing arrangement tables listed above, the following customer entity tables are deleted:

- BL1_CHARGE
- BL1_RC_FREQ_CREATION

Both the customer and billing arrangement statuses are set to Not Processed.

## Special Processing for Cross-distribution

If a customer whose charges are paid by the billing arrangement of another customer undergoes Undo, the status of the payer's billing arrangement is set to Payer Charges (i.e., it includes its own charges only, not the charges of the other entities it pays for).

# 15. BILL CONFIRMATION

Once the QA process is complete, and the rejected entities (if any) have been marked for Undo, confirmation can be requested for the remaining cycle's customers and billing arrangements. This enables the CSP to release the confirmed results to bill printing or further processing outside the Billing component.

Once confirmed, an entity cannot be undone. Likewise, entities marked for Undo cannot be confirmed. Thus, the Undo and Confirmation processes may be run simultaneously, or in any order, once the rejects are marked for Undo.

This chapter describes the conditions required, and the results achieved, by the Confirmation process. For information on the individual processes that make up confirmation, see the *Confirmation Flow* section.

## Invoking Confirmation

The CSP uses Application Monitoring and Control (AMC) to request confirmation for a cycle. Run Request Listener initiates confirmation.

## Prerequisites for Confirmation of a Given Entity

The following conditions must be met in order for a given customer to be confirmed:

- The customer must have completed the billing process, with its status set to Complete (CM).
- The customer must not be marked for Undo. Its Undo Request Type must be set to No (N).

The following conditions must be met in order for a given billing arrangement to be confirmed:

- The billing arrangement must have completed the billing process, with its status set to Billed (BL).
- The status of customers distributing their charges to the billing arrangement must be set to Complete (CM) if in the same cycle, or to Confirmed (CN) if in a different cycle.
- Neither the billing arrangement nor any of its customers (i.e., service receivers that distribute their charges to that billing arrangement) may be marked for Undo. Their Undo Request Types must all be set to No (N).
- Bill Printing Extract must have been performed on the billing arrangement (requested by the policy set in Billing Configurator).

# Confirmation Functions

The Confirmation process performs the following functions:

- Extracts Accounts Receivable information – Information is extracted from the customer data and passed to Accounts Receivable. The criteria used for extracting information are set in Billing Configurator.
- Publishes Prepaid Bill Day Charges – Prepaid charges created by Bill Preparation are published as transactions, which in turn get picked up by Replenishment Management (RPL). Specifically, if a charge's revenue code indicates RC, or the charge source is Discount, and the pay channel is Prepaid, the charge is sent to RPL by publishing it as a transaction.
- Sets statuses to Confirmed – When confirming a customer, the statuses of the customer entity, its billing arrangement entity, and the billing arrangement's documents are all set to Confirmed. For the document, this also means setting the Last Document ID to the current document number, and the Last Production Date – to the cycle end date. This prohibits these entities from being undone and permits their further processing.
- Confirms cycle changes – If a cycle change has been requested via Customer Management (CM), the current cycle code is set to the newly-requested cycle code, and is published as a transaction to be picked up by CM.
- Checks the status of each entity and sets the cycle status on that basis – If the status of each entity is set to Confirmed, the cycle status is set to Confirmed.

# 16. Reference Table Synchronization

This chapter describes the synchronization process of the Billing reference tables.

## General Description

Reference Table Synchronizer (RTS) is a generic application that coordinates the distribution and synchronization of data stored in tables in the various components' databases. It distributes data from the master reference tables to target reference tables in such a way that the component application is not required to be connected to another database. The following describes how RTS synchronizes between Billing and Product Catalog:

- Implementation Repository – Used to create CET and DET XMLs. It is also sent "as is" to Discount Engine to fill the DE reference tables.
- Full Discount Packages – Sent to Discount Engine to fill the DE reference tables.
- Discount Packages – Sent to Discount Engine to fill the DE reference tables.
- Pricing Packages – Copied "as is" for the PC Analyzer use.
- Offers – Copied "as is" for the PC Analyzer use.
- Business Entities – Used to create a BOH billing XML

## Synchronization Modes

Billing uses the following synchronization modes for reference table synchronization:

- Copy without conversion
- Simple conversion
- Complex conversion (callback function)

## Copy without Conversion

In this mode, the reference table is copied “as is” from the master table. Used for the following tables:

- CUSTOMER_ACT, OFFER_ACT, CUSTOMER_OFFER_ACT are copied from PC tables
- BL1_PARTITION_DEFINITION is copied from the Rater Partition Definition table
- BL1_XML_DISTRIB is populated for PC events from the Offers and Pricing packages tables.

## Simple Conversion

In this mode, the reference table is partially copied using an XSL file.

For example, the Charge Code reference table is updated using the PC table and via an XSL file.

## Complex Conversion (Callback Function)

Callback function mode is used for additional updates. For example, the BL1_XML_DISTRIB table is populated during the callback events called for PC events of Implementation Repository, Full Discount Packages, Discount Packages, or Business Entities.


*Copy and Simple Conversion events are usually triggered and updated by RTS. Complex Conversion events are usually triggered by callback events.*