# Contents

Notices.................................................................................................................................................................
Preface.................................................................................................................................................................

List of Tables......................................................................................................................................................... xi

## Chapter 1: Documentation Standards....................................................................................................................... 13
  Interface Terms...................................................................................................................................................... 14
  Typographic Conventions................................................................................................................................... 14

## Chapter 2: Getting Started................................................................................................................................. 15
  Complete BillMax Paperwork.................................................................................................................................. 16
  Install CentOS 7.x® for BillMax.......................................................................................................................... 16
  Install BillMax..................................................................................................................................................... 17
  Additional Tasks.................................................................................................................................................. 18
  Configure BillMax System Parameters............................................................................................................... 18
    Add Authorized Users......................................................................................................................................... 18
    Configure Global Options.................................................................................................................................. 18
    Update Lists....................................................................................................................................................... 19
    Configure Virtual Company.............................................................................................................................. 19
    Update Account Profiles.................................................................................................................................... 19
    Configure Taxes............................................................................................................................................. 20
    Configure Products......................................................................................................................................... 20
    Configure Payment Processing......................................................................................................................... 20
  Work Flows......................................................................................................................................................... 21
    Add a Prospective Customer/Bill Before Site Visit.......................................................................................... 21
    Add a Prospective Customer/Bill After Site Visit.......................................................................................... 21

## Part I: Fundamentals........................................................................................................................................... 23

## Chapter 3: Automated Processing.......................................................................................................................... 25
  Concepts................................................................................................................................................................. 26
    Automated Processing Overview.......................................................................................................................... 26
    Batch Process..................................................................................................................................................... 26
    Batch Process Group......................................................................................................................................... 26
  How To................................................................................................................................................................. 27
    Modify Email Recipients for a Batch Process Group........................................................................................ 27
    Stop Batch Processing....................................................................................................................................... 27

## Chapter 4: Billing.................................................................................................................................................. 29
  Concepts................................................................................................................................................................. 30
    Allocations.......................................................................................................................................................... 30
    Automated Complimentary Service.................................................................................................................. 30
    Billing/Prorate Day of Month............................................................................................................................. 31
    Commissions................................................................................................................................................... 31
    Discounts......................................................................................................................................................... 32
Prepare an Authorized User for Scheduling ................................................................. 209
Setup Scheduling and Appointments ........................................................................ 209
Add Customer Appointment from the Account ......................................................... 209
Generate Appointment Documents ............................................................................. 210

Part V: Ticketing ....................................................................................................... 211

Chapter 23: Concepts ............................................................................................... 213

BillMax Ticketing Overview ..................................................................................... 214
Messages and Notes ................................................................................................. 214
Service Level Agreements (SLAs) ............................................................................ 215
Issues and Resolutions .............................................................................................. 215
Team Members and Ticket Subscribers ................................................................. 216
Tickets ..................................................................................................................... 217
Queues ..................................................................................................................... 217

Chapter 24: How To ................................................................................................. 221

Enabling Inbound Email ......................................................................................... 222
Enabling Outbound Email Messages ....................................................................... 222
Enabling Ticket Notifications ................................................................................. 223
Jump Start Ticketing ............................................................................................... 223
Jump Start SLA Use ................................................................................................ 223
Stop Assignment of Tickets to Certain Authorized Users ....................................... 224

Part VI: Salesforce Integration ................................................................................ 225

Chapter 25: Concepts ............................................................................................... 227

Integration Overview ............................................................................................... 228

Chapter 26: How To ................................................................................................. 229

Update Salesforce Schema ..................................................................................... 230
Add Database Trigger to Populate Account.Name ............................................... 230
Setup Outbound Messages ....................................................................................... 231
Setup Workforce Actions and Workflow Rules ...................................................... 231
Edit BillMax Customer Portal Configuration File ............................................... 231
Create Salesforce CRM User ................................................................................ 232
Configure BillMax Hooks ....................................................................................... 232

Chapter 27: Reference .............................................................................................. 233

Salesforce Field Mappings ..................................................................................... 234
Salesforce Synchronization Log Files and Error Messages .................................. 235

Part VII: Service Availability Determination ........................................................ 237

Chapter 28: Fundamentals ...................................................................................... 239

Concepts ................................................................................................................ 240
Part VIII: VOIP/Telecommunication

Chapter 29: Concepts

Chapter 30: How To

Chapter 31: Reference
List of Tables

Table 1: Payment Processing Settings............................................................................................ 70
Table 2: IPPay e-check List entries................................................................................................75
Table 3:............................................................................................................................................93
Table 4:............................................................................................................................................94
Table 5:..........................................................................................................................................161
Table 6: Error Code Explanations................................................................................................ 166
Table 7: Database Conventions.....................................................................................................166
Table 8: Accounts..........................................................................................................................234
Table 9: Contacts...........................................................................................................................234
Chapter 1

Documentation Standards

Topics:

- Interface Terms
- Typographic Conventions
Interface Terms

Context Menu
The menu displayed on the left hand of a page.

Main Menu
The menu displayed at the top of the page.

View
To access and display a BillMax Entity or Report.

Account Overview
When viewing an Account, the static block of information about the Account at the top of the tabs.

Typographic Conventions

- Selection of a menu item is Menu Item or Menu Item > Menu Item
- Clicking or activating a button is "Click Button Name"
- Reference to a block of data on a page is Title where Title is the title of the block of data.
- General references to input elements on a page will be Element Name.
Chapter 2

Getting Started

Topics:

• Complete BillMax Paperwork
• Install CentOS 7.x for BillMax
• Install BillMax
• Additional Tasks
• Configure BillMax System Parameters
• Work Flows
**Complete BillMax Paperwork**

A quote and lease agreement will be supplied by BillMax Sales staff. To begin the implementation both documents must be returned.

1. Review the quote and sign at the bottom.
2. Review the lease agreement and initial each bottom right corner
4. Fill in company contacts on p. 12 and payment information on p. 13. Payment information may also be taken over the phone if preferred for security reasons.
5. Return by email to support@billmax.com or by fax at (817)446-7773.

**Install CentOS 7.x® for BillMax**

Describes a basic installation of CentOS for BillMax. Specifics such as network interface name, host name, IP address, etc. will differ. 64 bit architecture is required.

1. Retrieve from [http://www.centos.org](http://www.centos.org) the latest version of the Minimal ISO 64 bit for CentOS 7.
2. Start the installation
3. Choose *Test this media & install CentOS 7*.
4. Click *Continue*.
5. Click *Next*.
6. Click *Date & Time* to set the Time Zone.
7. Scroll down and Click **Installation Destination** to set disk partitioning.

8. Click **Network & Host Name** to set networking parameters. Set **hostname** to an entry that is or will be in DNS.

9. Click **Begin Installation**. Installation will begin.

10. Click **Root Password** to set the password. Not User need be created at this time.

11. Click **Reboot**.

---

**DANGER:** Do not install any other software or packages until after the install and configure steps for BillMax. Doing so may cause the BillMax installation or configuration steps to fail.

---

**Install BillMax**

1. Download the BillMax RPM.
   a) Using a web browser, go to [http://www.billmax.com](http://www.billmax.com).
   b) Click **LOGIN**. Contact support@billmax.com for the user name and password if you don't know it.
   c) Download the Cent OS 7 RPM.
   d) Copy the RPM to the Cent OS machine.

2. Install BillMax.
   a) Login to the Cent OS machine as **root**.
   b) Execute `yum -y groupinstall base`.
   c) Execute `rpm -ivh nameOfRPM`.
   d) Execute `cd /opt/billmax/install`.
   e) Execute `./BxInstall.pl`.
Note: Accepting the defaults for the prompts is recommended.

DANGER: When specifying the root password for the database, do not use the "$" or the "#" character.

f) Execute `yum -y update`.

g) Make note of the URL displayed which includes `BxConfigure.cgi`.

3. Execute `/usr/local/billmax/bin/machid`. Send the output to license@billmax.com to request a license.

4. After receiving the license, using a web browser, go to the URL displayed at the end of step 2 which includes `BxConfigure.cgi` and continue with configuring BillMax.

Additional Tasks

1. Set up printing for PDF files if automated printing desired. This is typically done using CUPS.

2. Set up DNS records. If the server will not be using an email relay, make sure an SPF record for the server is in DNS.

3. Set up email. If the server is sending email directly, then correct DNS entries are all that is needed. If using an existing email relay, then set up the relay. The easiest way to do this is using PostFix. There are many examples on the web of how to do this. Test email by sending an email from the server to a G mail account. If the email is accepted, check the headers to make sure SPF passes.

4. Make arrangements with a Credit Card processor that is supported by BillMax. See https://www.billmax.com/partners/. Other processors in Canada may be available.

5. Set up backups. See Backups, Redundancy and Archiving on page 119.

Configure BillMax System Parameters

Add Authorized Users

Each staff member at your company must have their own unique login id and password. Because BillMax tracks activity by login id, only one person can use the login at any given time.

1. Login to the BillMax Staff Portal using an ID with Administrator privileges.

2. Select System Administration > Authorized Users.

3. Click the New button on the Context Menu.

4. Enter Login Id*.

5. Fill in any other data.

6. Click Save.

Configure Global Options

To set basic settings for your BillMax instance:

1. Login to the BillMax Staff Portal using an ID with Administrator privileges.

2. Select System Administration > Global Options.

3. Enter the name of your company for Site Name.

4. If mapping module was purchased with your lease, enter provider supplied in Geocode Function.

5. To enable address verification service by the USPS, see Setup USPS Address Verification on page 126.

6. If you wish the Account, User, or Service numbers to begin with something other than one, enter Account Holes, User Holes, Service Holes as described in the tool tip.

7. Browse and locate an appropriate Logo Upload File for internal documents. It should be a .JPG, .PNG, or .BMP file.
8. Choose **File Name of Logo**.
9. Select **Save** from the Context Menu.

Other fields are defaulted and should only be changed with the input of support@billmax.com.

**Update Lists**

We recommend updating this lists before going live with BillMax.

<table>
<thead>
<tr>
<th>acodes</th>
<th>List of advertising codes - reasons people found your company</th>
</tr>
</thead>
<tbody>
<tr>
<td>bankaccounts</td>
<td>List of bank accounts that money will be deposited</td>
</tr>
<tr>
<td>cancelreasons</td>
<td>List of reasons that customers list your service</td>
</tr>
<tr>
<td>chargetypes</td>
<td>List of categories that revenue is earned or QuickBooks income accounts</td>
</tr>
<tr>
<td>nosuspendoverduedates</td>
<td>List of holidays not to suspend accounts</td>
</tr>
<tr>
<td>accountclasses</td>
<td>List of Account Classes. Used for financial reports. Also used for product selection.</td>
</tr>
</tbody>
</table>

Review and update each list. There are others that you might want to review as you continue using BillMax.

**Related tasks**

*Update Lists* on page 19

**Configure Virtual Company**

Create and set parameters for each company that will bill and report from BillMax.

1. Select **Billing Administration > Virtual Company**.
2. If a data import was completed for your company, select the highlighted underlined number of the Virtual Company. Otherwise, select **New**.
3. Enter the address fields. The highlighted fields are required.
4. Enter an Administrative contact. The **Email** will be contacted for critical errors and BillMax generated system notices.
5. Enter a Billing contact. The bills will be sent from this **Email**. Who do you want your customers to reply to?
6. Choose a logo file in **Upload New Logo**. This logo will appear on the documents that are sent to your customers.
7. Other settings are defaulted to common selections. Review and update as desired.
8. Select **Save**.

**Related information**

*Virtual Company* on page 50  
Virtual Company

**Update Account Profiles**

We recommend reviewing all profiles before going live with BillMax.

1. If a data import was completed for your company, select the highlighted underlined number of an Account Profile. Otherwise, select **New**.
2. Rename the Account Profiles if a data import was done to something meaningful about each group of customers.
3. Update settings. See **Update Account Profiles**.

**Related information**

*Account Profile* on page 50
Configure Taxes
BillMax has several steps to set up taxes.
1. Set up a tax regions. See Add A Tax Region on page 133
2. Add all tax items and fees to the Tax Item Index. See Add A Tax Item or Fee on page 133
3. Set up all tax groups necessary for your clients' locations. See Add A Tax Group on page 132
4. To check for completeness, select Billing Administration > Check Tax Setup.

Configure Products
Create and set parameters for Products in BillMax. Below are the steps for creating a Product Definition for a Recurring Service
1. Select Billing Administration > Recurring Service.
2. Select highlighted underlined number or New.
3. Enter the Name that will be used in the drop-down list of Services.
4. Enter the Billing Display for the text that the customer will see on a bill.
5. Choose the Charge Type to indicate the category for financial reporting.
6. Review and enter any other field on the General tab. There are many settings for services available including pricing, setup fees, FCC report settings and tax information. The tool tips give detailed information about a field.
7. Select the Provisioning tab to enter provisioning field information. Documentation for Radius provisioning is coming. Contact sales@billmax.com for further information on provisioning.
8. Select the Additional Usage Billing tab if desired. For example, this tab allows you to connect a VOIP Plan to a service definition for VOIP services.
9. Select Save.

Related information
Product Overview on page 82

Configure Payment Processing
Set parameters for Payment Processing in BillMax. Remember there is a list for credit cards and one for e-check for each processor.
1. Select Billing Administration > Lists.
2. Find the processor's lists by clicking Next as needed from the Context Menu.
The name of the processor is in the List name.
3. Click the List Number.
4. Change the following minimum settings. Contact support@billmax.com if interested in other settings.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultsEmail</td>
<td>Admin email address</td>
</tr>
<tr>
<td>State</td>
<td>Active</td>
</tr>
<tr>
<td>Test</td>
<td>0</td>
</tr>
<tr>
<td>UserName</td>
<td>Processor Terminal Id</td>
</tr>
</tbody>
</table>

Related information
Payment Processing Overview on page 70
**Work Flows**

### Add a Prospective Customer/Bill Before Site Visit

Describes a methodology for speedily adding a Prospective customer and collecting money before going for a site visit.

1. **Setup**
   a) Make sure all Package/Service Definitions have the following set to **Yes**:
      - Bill First Price in Prospective Status
      - Bill Setup Fee in Prospective Status
      - Bill Deposit in Prospective Status
   b) Make sure all Package/Service Definitions have Initial Service State set to Prospective and Initial Service Reason set to Wait for Qualification or Wait for Activation.
   c) If there are Recurring Package Elements that should be automatically added and billed, set the Minimum Quantity to 1.
   d) If there are One Time Package Elements that should be billed, make sure the Quantity is set to at least 1.

2. **Creation**
   a) Add an Account in the usual way.
   b) Add the Services to the User in the usual way. Leave the Services in Prospective Status. Pending Transactions will have been automatically created.
   c) Invoice the Customer and collect payment.

3. **Opening Packages/Services after Activation**
   a) **Important**: If this is the first Open Package/Service and anniversary billing is implemented, this day part of the date will set the **Billing/Prorate Day of Month** on the Account. It is very important to set this correctly to avoid billing cleanup.
      - Fill in Start Date if Activation is in the past.
   b) Change Status to Open and Save.

### Add a Prospective Customer/Bill After Site Visit

Describes a methodology for speedily adding a Prospective customer and collecting money after going for a site visit.

1. **Setup**
   a) Make sure all Package/Service Definitions have the following set to **Yes**:
      - Bill First Price in Prospective Status
      - Bill Setup Fee in Prospective Status
      - Bill Deposit in Prospective Status
   b) Make sure all Package/Service Definitions have Initial Service State set to Prospective and Initial Service Reason set to Wait for Qualification or Wait for Activation.
   c) If there are Recurring Package Elements that should be automatically added and billed, set the Minimum Quantity to 1.
   d) If there are One Time Package Elements that should be billed, make sure the Quantity is set to at least 1.

2. **Creation**
   a) Add an Account in the usual way.
   b) Add the Services to the User in the usual way. Leave the Services in Prospective Status. Pending Transactions will have been automatically created.
   c) Set each Pending Transaction to Manual Bill Only. If using the Scheduling module, this can be done when creating the Appointment.

3. **Opening Packages/Services after Activation**
a) **Important:** If this is the first Open Package/Service and anniversary billing is implemented, the day part of the date will set the **Billing/Prorate Day of Month** on the Account. It is very important to set this correctly to avoid billing cleanup.

   Fill in Start Date if Activation is in the past.

b) Change Status to Open and Save.

4. Invoice the Customer and collect Payment.
Part I

Fundamentals

Topics:

- Automated Processing
- Billing
- Bookkeeping
- Customer Organization
- Customer Communications
- Financials
- Payments and Refunds
- Products
- Provisioning
- Remote Applications
- Staff Portal
- System Administration
- Taxes
- Reference
Chapter 3

Automated Processing

Topics:
- Concepts
- How To
Automated Processing Overview

Automated processing is how recurring tasks are accomplished by BillMax. Examples of tasks performed on a recurring basis include:

• Checking the data in the database for logical integrity
• Billing customers the recurring sales and possibly usage sales
• Sending billing documents
• Sending customer notices such as credit card expiration and overdue notices

These processes are executed using the underlying operating systems' cron facility.

Recurring tasks are referred to as a Batch Process and are part of a Batch Process Group. From the top menu, select System Administration > Batch Processing to see what Batch Process Groups are available. When BillMax is installed, no Batch Process Groups are enabled.

The current list of Batch Process Groups installed with BillMax are:

• nightly - enable this when putting BillMax into production. See nightly Typical Batch Processes on page 179.
• cardupdater - enable this to use the IPPay® credit card updating service. This must also be enabled at IPPay®
• mikromatedata - enable this to collect data from Mikrotik® routers

BillMax customers may add additional Batch Process items to existing Batch Process Groups and may also add their own Batch Process Groups.

Batch Process

A Batch Process is a task that is part of a Batch Process Group. The task is a program that is executed. Only a Batch Process where Status is Active and is part of a Batch Process Group that has a Status of Enabled is executed.

The output of every Batch Process is logged to a file in /usr/local/billmax/logs. The log file will be emailed to the email specified if applicable.

A Batch Process may be configured such that if the Batch Process exits with a system code indicating an error, process of the Batch Process Group will cease.

If needed, a single Batch Process may be executed from the Staff Portal.

Batch Process Group

A Batch Process Group is a group of Batch Process entries that are executed sequentially at a specified time. The time at which a Batch Process Group is executed is specified us cron notation.

CAUTION: Due to the date/time nature of the processing it is recommended that a long running process not cross 2:00 A.M. and possible not be scheduled earlier than 2:15 A.M. due to Daylight Savings Time issues. Although most operating systems will account correctly for Daylight Savings Time, this has not always been the case.

All the output files created by each Batch Process are grouped into one file and will be emailed if an email address is specified on the Batch Process Group.

The intent for a Batch Process Group is that it be executed using the cron facility. However, if need be, it may be executed using the Staff Portal.
How To

Modify Email Recipients for a Batch Process Group

1. Select System Administration > Batch Processing > .
2. Choose the Batch Process Group.
3. Enter into Email the email addresses.
4. Select Save from the Context Menu

Stop Batch Processing

1. From the Main Menu, select System Administration > Batch Processing.
2. Select BillMax Nightly Processing.
3. Choose Disabled from Status.
4. Select Save from the Context Menu.
Chapter 4

Billing

Topics:

* Concepts
* How To
Concepts

Allocations

Allocations are internal records that are used in BillMax to tie credits to debits. Allocations are the basis for the Cash based Sales Reports. Allocation records consist of:

1. Allocations
2. Allocation Reversals

Allocations describe both the date and time of the Allocation as well as the amount of the credit allocated to the debit. Allocation Reversals reverse the original Allocation, either partially or in total. Allocation Reversals may be created under the following circumstances:

1. When a credit is reversed.
2. When a debit is reversed.
3. When the Allocations are manually adjusted through the BillMax Staff Portal.

Under normal processing, BillMax automatically creates Allocations and Allocation Reversals. However, BillMax does allow for interrupting this normal processing. Transactions may be flagged to have manual Allocations only. Under this circumstance, someone will have to use the BillMax Staff Portal link the credits to debits.

Flagging transactions for manual Allocations may lead to situations where the Account's balance is zero, but the Account is considered overdue. This occurs because unused credits are not linked to unpaid debits.

Flagging a transaction for manual Allocations is very rare. Sometimes it is set temporarily to manipulate the Allocations to the transaction, but in general should not be left in that condition. It should be reverted to accepting automated Allocations.

Automated Complimentary Service

There are four ways to delivery complimentary service. These are listed below from most desirable to least desirable.

Note: Regardless of the technique chosen, if the customer is an Account in BillMax and is to never receive an Invoice or a Billing Statement, make sure to de-select all Account Automated Billing Methods.

Discounting

Give complimentary Service by using Discounts. Discounts may be given by either specifying a percentage discount or a discount amount. In addition, an expiration date for the Discounts may be entered.

The effect of specifying a discount is that both a Sale and a Store Credit are booked to the Account. This enables tracking the amount of complimentary service.

The discount may be specified on the following Customer Management Entities:

- Account
  1. Percentage based only.
  2. Applies to every billing entered for the Account.
- User
  1. Percentage based only.
  2. Applies to every automated billing entered for the Packages and Services associated with the User.
- Package
  1. Percentage based or Amount based.
  2. Applies to automated billings for the Package.
  3. Does not apply to Services or Fees associated with the Package
- Service
1. Percentage based or Amount based.
2. Applies to automated billings for the Service.
3. Does not apply to the Fees associated with the Service.

**Zero Dollar Pricing**

When setting up a Package or a Service, the pricing may be set to $0.00. However, if fees are being assessed they will continue to be assessed.

Zero dollar sales are entered in the customer's financial record. Whether they are displayed to the customer or not is controlled by the Package or Service Definition setup. Packages and Services that are setup with any pricing that deviates from the Package or Service Definition will show up in the Billing Exceptions report.

The disadvantage to using Zero Dollar Pricing is there is no record of the financial amount of complimentary service.

**Free Pricing**

When setting up a Package or a Service, the Package or Service may be marked as Free. This may be useful when allowing a customer to "try before buying".

The disadvantage to using Free Pricing is that there is no financial record of the package or service.

**Provision Services outside of BillMax**

This last method is not recommended but may be desirable under special circumstances. Provisioning services outside of BillMax means that there is no record in BillMax of the services. This business practice can lead to missed billings and fraud without a thorough audit procedure in place. Exception lists will have to be maintained.

**Billing/Prorate Day of Month**

The **Billing/Prorate Day of Month** is the day of month to which all automated monthly billing is anchored. It is also the day of month to which monthly billings are prorated if the day of a Package or Service **Next Bill Date** is not considered equal to the **Billing/Prorate Day of Month**.

The default for a new Account is derived from the **Billing Day of Month/Prorate Day** specified on the Account Profile. The default will be one of the following:

1. The day of month the Account is created in BillMax.
2. The day set on the Account Profile.

If the Account Profile setting for **Billing Day of Month/Prorate Day** is **Day of Account Creation**, the **Billing/Prorate Day of Month** will be changed automatically by BillMax when the first Package or Service for the Account is Opened. This avoids unnecessary prorating when the first Package or Service is Opened on a date after Account creation.

**CAUTION:** Changing **Billing/Prorate Day of Month** for an Account after billing for Packages and Services has commenced may cause prorating the next billing cycle.

The date that monthly billing actually occurs is derived from the **Billing/Prorate Day of Month** and the **Bill Timing** settings on the Account Profile.

**Commissions**

BillMax keeps track of Sales and Store Credit amounts for the purpose of computed commissions. These amounts may be associated with one or more Agents. The amounts may be on a billed (Accrual) or collected (Cash) basis.

BillMax does not compute commission amounts. Each company has its own way of computing commissions. BillMax provides the data necessary to compute commissions.

An Agent is an **Authorized Users** on page 122 with the Agent field selected.
An Agent may be attached to a recurring billing product. A start date and end date may be specified such that all transactions between and including the dates may be associated with the Agent. An Agent may also be specified on a one time sale or store credit.

**Discounts**

Discounts, for usage or non-usage billing items, may be specified on the following Customer Management Entities:

- **Account**
  1. Percentage based only.
  2. Applies to every billing entered for the Account.
- **User**
  1. Percentage based only.
  2. Applies to every automated billing entered for the Packages and Services associated with the User.
- **Package**
  1. Percentage based or Amount based.
  2. Applies to automated billings for the Package.
  3. Does not apply to Services or Fees associated with the Package
- **Service**
  1. Percentage based or Amount based.
  2. Applies to automated billings for the Service.
  3. Does not apply to the Fees associated with the Service.

**Financial Transactions**

The basic financial transactions in BillMax are:

1. **Debits**
   a. Sale
   b. Refund
   c. Deposit
2. **Credits**
   a. Payment
   b. Store Credit

In addition, BillMax is designed such that a transaction does not need to be deleted to make an adjustment to the account. BillMax supports the following transaction reversals:

1. **Credits**
   a. Sales Return
   b. Refund Reversal
   c. Deposit Reversal
2. **Debits**
   a. Payment Reversal
   b. Store Credit Reversal

Reversals are used to track the reversal event at the time of the reversal and do not affect either the date or the amount of the original transaction thus preserving financial history.

**Sales and Sales Returns**

Sales are used to bill a customer for Services, Items and Fees. Sales may be tagged as "Taxable", "non-Taxable" and "Tax Not Applicable". Sales may be paid off by a customer Payment or a Store Credit.
Sales Returns are the preferred method for making a Sales adjustment. Use a Sales Return instead of a Store Credit. Unlike Store Credits, Sales Returns manage the tax portion of the Sales Return to the penny.

Use Sales Returns to make tax adjustments. Create a Sales Return to return Sales that have the incorrect tax and create new Sales with the correct tax.

Sales Returns may be used to reverse either a complete Sale or a partial Sale. A partial Sales Return may be a partial quantity Sales Return, a Sales Return based on a date range or both. If multiple Sales Returns are created against one Sale, BillMax will keep track to make sure that the sum of the Sales Returns do not exceed the original Sale.

If a Sale has been paid off, the Sales Return will cause the Payments and Store Credits used to payoff the Sale to be available for other Debit transactions.

Sales Returns may be automatically created under the following conditions:

1. When executing a Service Change. Future Sales of the existing Service may have an offsetting Sales Return created.
2. When re-opening a Service. Sales for the period the Service was not opened may have an offsetting Sales Return created.
3. When closing a Service. Future Sales of the existing Service may have an offsetting Sales Return created.

**Store Credit and Store Credit Reversals**

Store Credits are used to give credit to an account for reasons other than a Sales Return. Reasons may include:

1. Good Will
2. Referral Credit
3. Service Outage
4. Discounts
5. etc.

Store Credits may be used to pay off Sales only. **Store Credits may not be used to pay off Refunds or Deposits.**

Store Credit Reversals may be used to reverse either a complete Store Credit or a partial Store Credit. A partial Store Credit Reversal may a partial quantity Store Credit Reversal, a Store Credit Reversal based on a date range or both. If multiple Store Credit Reversals are created against one Store Credit, BillMax will keep track to make sure that the sum of Store Credit Reversals do not exceed the original Store Credit.

**Payments and Payment Reversals**

A Payment is booked in BillMax when funds or legal tender is received from the customer. Methods of Payment include:

1. Cash
2. Check
3. Credit Card
4. Electronic Check

A Payment may be used to pay off a Sale, a Refund or a Deposit. If a Payment has been used to pay off any debit, it may not be refunded. **Only Payments that are not being used to pay off a debit may be refunded.**

Payment Reversals are used to reverse a Payment. **Payment Reversals are not a Refund.** Payment Reversals reverse the entire Payment. Payment Reversals are used to debit a customers account for the following reasons:

1. An NSF check.
2. A Credit Card chargeback
3. A payment booked to an incorrect account
4. Any other reason that a customer should not be credited the original payment.
**Refunds and Refund Reversals**

Refunds are booked in BillMax when funds or legal tender originally received as Payments are voluntarily given back to the customer. Methods of Refund include:

1. Cash
2. Check
3. Credit Card
4. Electronic Check

Refunds are paid off by Payments only. **Store Credits may not be used to pay off Refunds.**

Refund Reversals are used to reverse a Refund. Refund Reversals reverse the entire Refund.

**Deposit and Deposit Reversals**

Deposits are used to debit a customer a deposit amount. A Deposit is not considered a sale and is considered a liability. Deposits are paid off by Payments only. **Store Credits may not be used to pay off Refunds.**

A Deposit is typically collected in exchange for a piece of equipment.

Deposit Reversals are used to reverse a Deposit. Deposit Reversals reverse the entire Deposit.

**Financial Transaction Dates**

There are three dates associated with every financial transaction:

- **Enter Date** - The date the transactions was entered into BillMax.
- **Post Date** - The date the transaction was posted to an Account.
- **Effective Date** - The date the transaction should be considered effective.

The following lists date ramifications for the various transactions.

- Payments, Payment Reversals, Refunds and Refund Reversals
  - Enter Date and Post Date are the Same
  - Effective Date may be current date or in the past.
- Sales, Sales Returns, Store Credits, Store Credit Reversals, Deposits and Deposit Reversals
  - Enter Date, Post Date and Effective Date all may be different.
  - Post Date is the date the transaction is invoiced.
  - Effective Date is the Effective Date of the invoice.

**Late Fees**

Late fee assessment is controlled by settings on the **Account Profile**. Timing and amount of the late fee will depend on the **Late Fee Processing** settings.
Late Fee By Monthly Balance

When Late Fee is set to By Monthly Balance, late fees are assessed once a month. What day during the month is controlled by User Late Fee Grace Days:

No

Late fees are assessed the day the monthly Billing Statement is created. This is true regardless of whether the customer receives Invoices or Billing Statements.

Example

Assume a customer has their automated monthly Billing Statement created on the first of the month. On the first of the month the late fee is assessed for overdue amounts. No consideration is given when the amounts became overdue.

Yes

Late fees are assessed on the day after the monthly Billing Statement due date plus the number of Late Fee Grace Days based on the customers payment method.

Example

Assume a customer has their automated monthly Billing Statement created on the first of the month with a due date the 15th of the month. Also assume the customer pays by credit card and Late Fee Grace Days for Automated Payment Customers is 10. On the 26 of the month (day after the due date of 15th + 10 days) the late fee is assessed for overdue amounts.

Late Fee By Invoice

When Late Fee is set to By Invoice, late fee are assessed once a month per Invoice. What day during the month is controlled by User Late Fee Grace Days:

No

Late fees are assessed the day after the Invoice due date and every month thereafter. This is true regardless
of whether the customer receives Invoices or Billing Statements.

Example
Assume a customer has an Invoice created on the first of the month with a due date of the 25th. On the 26th of the month the late fee is assessed for overdue amounts belonging to the Invoice.

Yes
Late fees are assessed on the day after the Invoice due date plus the number of Late Fee Grace Days based on the customers payment method.

Example
Assume a customer has an Invoice created on the first of the month with a due date the 15th of the month. Also assume the customer pays by credit card and Late Fee Grace Days for Automated Payment Customers is 10. On the 26 of the month (day after the due date of 15th + 10 days) the late fee is assessed for overdue amounts.

Fee Amount
The amount of a Late Fee depends on the following settings:

- Late Fees setting on the Account. If set to No, no automated Late Fees will be assessed.
- Overdue Minimum Amount specified in the Overdue Processing section of the Account Profile. If the total overdue amount for the Account under consideration is less than the Overdue Minimum Amount, no automated Late Fees will be assessed.
- Late Fee and Late Percentage in the Late Fee Processing section of the Account Profile. The overdue amount will be multiplied by the Late Percentage and the greater of the result and Late Fee will be used as the late fee.

Other
Whether or not taxes are assessed on late fees is configured through the Virtual Company setting Taxable under Fees.

Whether or not late fees are assessed on late fees being late is configured through the Virtual Company setting Assess Late Fees on Late Fees under Fees.

Pending Financial Transactions
Pending Financial Transactions are financial transactions that have been added to an Account but are not yet posted as part of an Invoice. As they are not posted the following is true:

1. Taxes have not been computed/assessed
2. Amounts do not show up on financial reports such as Sales, Book Keeping, Store Credit Activity, etc.
3. The customer balance is not affected. The customer is neither debited nor credited the Pending Transaction amount.

Pending Transactions consist of the following:

1. Pending Sales
2. Pending Store Credits
3. Pending Deposits
4. Pending Sales Returns
5. Pending Store Credit Reversals
6. Pending Deposit Reversals
When a Pending Financial Transactions are invoiced, the amounts are then reflected in the Account's balance and show up on financial reports. Until the Pending Financial Transactions are invoiced, the Account is not responsible for the debits or given credit represented by the Pending Financial Transaction.

BillMax nightly processing may be configured to automatically invoice Pending Financial Transactions daily. Otherwise, automated invoicing of Pending Financial Transactions will be will occur the next time an automated Invoice or Billing Statement is generated for the Account.

Pending Sales and Pending Store Credits may have a future date specified to delay automated invoicing.

**Refundable Amount**

The Refundable Amount for an Account consists of all Payment amounts that are not being used to payoff a debit. The total amount that may be refunded to a customer is limited to the total of the payments ever received from a customer. Store Credits may never be refunded to a Customer.

In order to book or process a refund for an Account, the **Refundable Amount** in the Account Overview must be negative.

In general a Refund should be made in the same way the payment was received. A Refund by check will be used for a Payment by check or cash. A Refund by credit card will be used for a Payment by Credit Card and so on. In cases where the Refund cannot be made by credit card because the transaction is too old or the credit card account has been closed, a Refund by check will need to be made.

There is a special case in which the **Refundable Amount** in the Account Overview may be positive. In this case, the customer owes the amount shown. One scenario in which this may occur is the following:

1. Customer makes a credit card payment paying off a Sale.
2. Customer discontinues Service and wants a Refund.
3. A Sales Return is created creating a negative Refundable Amount.
4. A check is written to the customer. The Refundable Amount is now 0.
5. The customer contacts their credit card company and a credit card charge back is granted.
6. The charge back is book in BillMax using a Payment Reversal on the original payment.

The end result is that the customer received the original payment back twice. The customer owes the original payment amount and the **Refundable Amount** is positive.

**How To**

**Billing Scenarios**

**Bill a Non-Standard Duration**

Bill a customer for one duration when almost everyone else is another duration.

Use these steps to setup billing a non-standard duration for a customer. An example scenario is that everyone is billed $35.00 monthly for a service, but a customer wants to be billed for 3 months at a time. This describes an alternate method to creating a Service Definition for that one customer.

1. Choose the new Service using the one month Service Definition.
2. Choose the duration and enter the price to reflect the new billing paradigm.
   
a) From:

   ![Billing Form]

   First Price: 35.00
   First Duration: 1 Month
   Number of Times to Bill First Price: 1
   Second Price: 0.00
   Second Duration: 2 Weeks
   Number of Times to Bill Second Price: 0
   Third Price: 0.00
   Third Duration: 2 Weeks
   Number of Times to Bill Third Price: 0

   ![Billing Form]

   b) To:
3. Enter other data needed for a new Service.
4. Click **Save**.

**Bill First Month/Collect for Last Month**

BillMax does not have this feature built in, but it can be accomplished using the Service Definition Deposit feature.

1. View the Service Definition.
2. Enter the **Deposit** in *Other Pricing* to match the last month's pricing.
3. Click **Save**.
4. Add the new Service to the Account. Unless overridden the Deposit will be automatically billed.

**Release Funds Collected for Last Month**

Release Funds collected under the Bill First Month/Collect Last Month scenario

1. Locate the Deposit that was used to collect the funds.
2. Create a Deposit reversal
3. The funds are now available to be used for other purposes.

**Bill a Customer Once a Month with Prorating**

What to do when billing a customer multiple Services that are not added on the same day of month.

1. **Warning:** Changing these setting after being in use will affect all customers subscribed the Packages and Services and may generate prorating across the board.

When setting up a Service or Package Definition, make sure that the **Prorating** setting are correct.

a) Set **Prorate Date/Amount** to **Yes** if both the time of service and the amount billed are to be prorated.
b) Set **Prorate Date Only** to **Yes** if the time of service is to be prorated, but the full amount should be billed. This is useful for government fees where the full amount is assessed monthly no matter how much of a month is used.

2. If dealing with Services and Package Definitions where **Prorate Date/Amount** and **Prorate Date Only** are set to **No**, then:
   
a) When adding a Service or Package to a customer, the Service or Package **Next Bill Date** will need to be set manually such that the day of the **Next Bill Date** matches the Account **Billing/Prorate Day of Month**.

b) Manually enter any billing adjustments needed for any prorating.

**Bill a Customer Once a Month without Prorating**

What to do when billing a customer multiple Services that are not added on the same day of month without prorating.

1. Edit the Account Profile to which the customer is associated.

2. **Warning**: This change will affect all customers associated with the Account Profile.

   Under **Processing Date Modifiers** select **Offset Day of Month**. When the customer is monthly billed based on the Account **Billing/Prorate Day of Month** and the other Account Profile **Bill Timing** parameters, all Services where the Next Invoice Date fall in the next 27-30 days will be billed.
Add An Agent

1. From the Main Menu, select System Administration > Authorized Users
2. Select New from the Context Menu or choose and existing Authorized User.
3. If a new Authorized User, fill in the form data and select Agent. If existing, select Agent.
4. Select Save from the Context Menu.

Add Agent to Package or Service

1. Bring up an existing Package or Service
2. Click the Sales Agent tab.
3. Click New Agent
   A Service Agent Dialog will appear.
4. Select the Agent
5. Enter the Start Date and End Date if necessary.
6. Click Save.

Report Amounts for Commissions

1. Select from the Main Menu Reports > Sales By Agent.
2. Enter the data for the report.
3. Tip:
   • Accrual No Deferred Revenue Excluded = Billed,
   • Cash/Credit = Collected.
   • Accrual Exclude Deferred Revenue = Exclude Billed but not delivered.
   Select the Type of Sales Report.
4. Select the type of output from the Context Menu.

Close an Account on Specific Date

1. View the Account.
2. In the Status box, set End Date to the specific date plus one so the customer will get service through the specific date. Overnight batch processing will then suspend the Account.
3. Set the **Reason** to **Expired**.
   If the **Reason** is **Customer Request**, then the **Account** is **Suspended** only and remains that way until manual intervention. **Customer Request** allows for customers such as snowbirds who schedule in advance of leaving town for months.

4. The Account will automatically close based on the **Days to Wait Before Closing** on the Account Profile.
   There is a delay because many times a customer gets suspended and change their mind. The delay saves on having to rebuild the Services if Provisioning is involved.

### Create Promotions

Promotions are Referral Credits that are offered on particular Packages or Services for a period of time. The Packages or Services must offer a Referral Credit for an amount other than zero.

1. Setup:
   a) Edit a **Package** or **Service** Definition.
   b) Specify the **Referral Amount** in the **General** section on the **General** tab.
   c) Specify the **Referral Wait Days**. These are the number of days to wait after a **Service** is **Open** before issuing the **Referral Amount**.
   d) **Save** the edits.
   e) Select **Billing Administration** > **Promotions**.
   f) Click **New** and specify the **Name, Dates, and Amount** for each Promotion.

2. Implementation:
   a) Edit or add a new **Package** or **Service**.
   b) Specify the **Referral Account** in the **General** section on the **General** tab. The magnifying glass will give you Search options available to location the Account.
   c) Specify the **Referral Amount** from the drop-down choices.

3. Results:
   a) Nightly processing will create the **Store Credit** transaction for the **Referral Account** if the **Service** is **Open** and the number of **Referral Wait Days** has passed.
   b) **Referral Amounts** are credited only to **Open** or **Suspended Accounts**.
   c) The description for the transaction is **Referral of UserFirstName UserLastName, ('NameOnService')**
   d) The **Charge Type** of the **Store Credit** for Sales Report purposes will be **Referral Credit**.
   e) The **Service Definition** of the **Store Credit** for Sales Report Purposes will be the **Package** or **Service** Definition of the referred product.

### Create Invoice Fees

Invoice Fees are percentage-based recurring fees that are added to each invoice one time based on the positive sum of transactions. Example: State Cost Recovery Fee

1. Select **Billing Administration** > **Invoice Fee**.
2. Click **New** and specify the **Name, Rate and Charge Type** for each Invoice Fee.
3. Specify the other fields as desired and click **Save**.
4. If the **Invoice Fee** is not to be passed on to an Account, bring up the Account and under **Taxes/Fees** set the **Tax/Fee No Pass Through** for **Invoice Fee**. Otherwise, all Accounts will be assessed the fee.

### Do a Sales Return or Store Credit Reversal

1. View the Account.
2. Click on the **Billing** tab.
3. If needed, expand the Invoice(s) that have the Sales to be Returned or Store Credits to be Reversed.
4. For each Sale or Store Credit
   a) Click the **number** for the transaction.
A dialog will appear.
b) Click the Sales Return or Store Credit Reversal tab.
c) Click Save Sales Return or Save Credit Reversal.

5. Select either Billing Statement or Invoice from the Context Menu and Post the Sales Return and/or Store Credit Reversal Invoice(s).

Return Sales Tax for a Tax Exempt Account

These steps are followed when a customer is billed for tax before discovery that the customer should be billed on a tax exempt basis.

1. View the Account.
2. Click on the Billing tab.
3. Expand the Invoice(s) that have the taxable sales.
4. For each taxable sale, create a Pending Sales Return.
5. For each taxable sale, create a comparable non-taxable Pending Sale

6. Select either Billing Statement > Post or Invoice > Post. The Account will now have the Sales Tax credited back to the Account and the Sales book correctly as non taxable Sales
Chapter 5

Bookkeeping

Topics:
- Concepts
- How To
Concepts

Bank Deposits

Bank Deposits are a means of grouping Payments, Refunds, Payment Reversals and Refund Reversals such that the sum of the transactions match an amount on an external statement such as a bank account statement or a credit card transaction report. These may be used for auditing or reconciling purposes.

For the purposes of this discussion, there are two types of bank accounts:

1. A BillMax tender bank account. This is an account that collects funds by type of tender (cash, check, Visa®, MasterCard®, electronic check, etc.).

2. The actual bank account to which funds are deposited. An example would be a business checking account at a local bank.

When an IIF file is created, Bank Deposits are used to create a QuickBooks® deposit transaction into the bank account specified on the BillMax Bank Deposit. The funds for the QuickBooks® deposit transaction come from the BillMax tender bank accounts created and funds populated in QuickBooks® by the IIF file.

Bookkeeping Overview

Bookkeeping is the process of entering transactions into an accounting system. For completeness, the following transactions should always be entered into the accounting system:

- Sales
- Sales Returns
- Store Credits
- Store Credit Reversals
- Deposits
- Deposit Reversals
- Payments
- Payment Reversals
- Refunds
- Refund Reversals

In addition, Payments, Payment Reversals, Refunds and Refund Reversals may be grouped into Bank Deposits. Entering these in the accounting system may also be desirable.

The recommended report for generating bookkeeping entries is the Bookkeeping Report. It generates all the necessary entries including tax amounts. If the software used for accounting is QuickBooks®, then an IIF file may be downloaded for import.

Bookkeeping Report

The Bookkeeping report is a report that combines multiple financial reports into a single report. These reports include:

- Sales - includes Sales, Sales Reversals, Store Credits and Store Credit Reversals
- Payments - include Payments and Payment Reversals
- Refunds - includes Refunds and Refund Reversals
- Write Offs

These used for updating an accounting system.

In addition to these transactions, Payments, Payment Reversals, Refunds and Refund Reversals may be grouped into Bank Deposits. If this is done, Bank Deposit information is also included in the Bookkeeping Report.

The data displayed by the Book Keeping report is the basis for creating and Intuit® IIF file suitable for importing into QuickBooks®.
• Sales Options map to Items in QuickBooks.
• IIF Income Account Options map to Income Accounts in QuickBooks.
• IIF Class Options map to Class in QuickBooks.

The default output from BillMax for Items, Income Accounts and Class may not be correct for the BillMax customer. BillMax provide a mechanism for mapping the Item, Income Account and Class values to those desired in QuickBooks.

How To

Default the Financial Behavior for Virtual Company and Account Profile Moves

How to set the default behavior when moving an Account between Virtual Companies and Account Profiles.

1. Select Billing Administration > Lists.
2. Choose systemstate list.
3. Choose movepayhistconsumerof.
4. Make edits.
5. Click Save.
6. Choose movepayhistrofile.
7. Make edits.
8. Click Save.
Chapter 6

Customer Organization

Topics:

- Concepts
- How To
Concepts

Customer Organization Overview

Customers are organized in the following manner:

1. A customer is represented by an Account.
2. An Account belongs to a Virtual Company.
3. An Account's billing rules are primarily stored in an Account Profile.
4. An Account Profile may be limited to one Account or be shared by many Accounts.
5. An Account may have multiple Users.
6. A User is used to represent an additional contact for an Account, an additional service location for an Account or both.
7. An Account may have a Guarantor Account.
8. A Guarantor Account is a customer that will pay the billings on behalf of other Accounts.
9. A User may have one or more Packages/Services/Fees which represent products sold/provisioned on a recurring basis to the customer. See Products on page 81.

Virtual Company

A Virtual Company is primarily used to collect Accounts under a single brand. The attributes, such as name, address, logo, of the Virtual Company are printed on the documents sent to the customer.

Although the primary intent to the Virtual Company is for branding, BillMax customers have used it for other means of segregating Accounts as all financial reports may be run on a Virtual Company basis. Other uses for Virtual Companies have been

- Segregate Accounts by geography such as state of residence of the customer.
- Segregate Accounts by reseller.
- Segregate Accounts by customer type (residential, commercial, etc.)

The Virtual Company also specifies some defaults setting for new customers added to BillMax.

All BillMax Authorized Users have access to all Virtual Companies.

Account Profile

Every Account is associated with an Account Profile. The Account Profile contains the bulk of the business rules that are applied to a customer. The business rule categories are

- Bill Timing
- Overdue Processing
- Late Fee Processing
- Electronic Payment minimums and maximums

The Account Profile also controls the automated document delivery format and presentation. Customers receive either Billing Statements or Invoices.

The Account Profile may be specific to a single Virtual Company or may be applicable across all Virtual Company.

Use an Account Profile for a group of customers that share the same business rules and document presentation. Unique customers may have their own Account Profile not available to other customers.

Bill Timing

Bill Timing refers to following:

- Scheduling when recurring transactions are added to an Account as Pending Transactions.
- Scheduling when Pending Transactions are posted to an Account.
Late Fees

Late fee assessment is controlled by settings on the Account Profile. Timing and amount of the late fee will depend on the Late Fee Processing settings.

![Late Fee Processing](image)

**Late Fee By Monthly Balance**

When Late Fee is set to By Monthly Balance, late fees are assessed once a month. What day during the month is controlled by User Late Fee Grace Days:

**No**

Late fees are assessed the day the monthly Billing Statement is created. This is true regardless of whether the customer receives Invoices or Billing Statements.

**Example**

Assume a customer has their automated monthly Billing Statement created on the first of the month. On the first of the month the late fee is assessed for overdue amounts. No consideration is given when the amounts became overdue.

**Yes**

Late fees are assessed on the day after the monthly Billing Statement due date plus the number of Late Fee Grace Days based on the customers payment method.

**Example**

Assume a customer has their automated monthly Billing Statement created on the first of the month with a due date the 15th of the month. Also assume the customer pays by credit card and Late Fee Grace Days for Automated Payment Customers is 10. On the 26 of the month (day after the due date of 15th + 10 days) the late fee is assessed for overdue amounts.
Late Fee By Invoice

When Late Fee is set to By Invoice, late fee are assessed once a month per Invoice. What day during the month is controlled by User Late Fee Grace Days:

No  Late fees are assessed the day after the Invoice due date and every month thereafter. This is true regardless of whether the customer receives Invoices or Billing Statements.

Example
Assume a customer has an Invoice created on the first of the month with a due date of the 25th. On the 26th of the month the late fee is assessed for overdue amounts belonging to the Invoice.

Yes  Late fees are assessed on the day after the Invoice due date plus the number of Late Fee Grace Days based on the customers payment method.

Example
Assume a customer has an Invoice created on the first of the month with a due date the 15th of the month. Also assume the customer pays by credit card and Late Fee Grace Days for Automated Payment Customers is 10. On the 26 of the month (day after the due date of 15th + 10 days) the late fee is assessed for overdue amounts.

Fee Amount

The amount of a Late Fee depends on the following settings:

- **Late Fees** setting on the Account. If set to No, no automated Late Fees will be assessed.
- **Overdue Minimum Amount** specified in the Overdue Processing section of the Account Profile. If the total overdue amount for the Account under consideration is less than the Overdue Minimum Amount, no automated Late Fees will be assessed.
- **Late Fee** and **Late Percentage** in the Late Fee Processing section of the Account Profile. The overdue amount will be multiplied by the Late Percentage and the greater of the result and Late Fee will be used as the late fee.

Other

Whether or not taxes are assessed on late fees is configured through the Virtual Company setting Taxable under Fees.

Whether or not late fees are assessed on late fees being late is configured through the Virtual Company setting Assess Late Fees on Late Fees under Fees.

Account

The Account in BillMax represents a customer or subscriber. The following are settings for an Account:

1. Addresses used for delivery of all billing documents.
2. All payment methods belong to the Account.
3. May be a Guarantor for Beneficiary Accounts.
4. All non Service related billings are associated primarily with the Account.
5. The Account is associated with both a Virtual Company (for branding purposes) and an Account Profile (for billing rules).

The Account may represent from an individual purchasing one or more Services in one location all the way to a corporation purchasing services in multiple locations and possibly on the behalf of its employees.
Guarantor/Beneficiary Accounts

Guarantor Accounts are Accounts that have agreed to pay the bills for another Account.

Beneficiary Accounts are Accounts associated with a Guarantor Account.

Guarantor Accounts have the following characteristics:

1. When entering a Payment, all the outstanding debits of the Beneficiary Accounts are present.
2. Automated Payment processing for the Guarantor Account includes debits from the Beneficiary Accounts.
3. Overdue amounts for Beneficiary Accounts do not affect the Guarantor Account. In other words, Beneficiary and Guarantor Accounts may be separately overdue.

Unlike Accounts where multiple locations and multiple services are billed and sent to the Account holder, billings for beneficiaries are not sent to the Guarantor but are sent to the Beneficiary Accounts. This allows for an approval process by the Beneficiary.

User

The User in BillMax represents the Service location of Services and/or additional contacts for the Account. It also may represent a different location for the Services. In general there will be one User per Account. Reasons to have multiple Users per Account are:

1. Multiple locations for Services where the Services are taxed differently.
2. Multiple locations for Services where the location of the Service needs to be reported such as on the FCC 477.
3. Multiple Users of Services such as employees in a corporation.
4. Multiple contacts for an Account. If integrating the Ticketing System with email, Tickets that come in will automatically be associated with the User that has the same email address as the From email on the Ticket.

The User holds the credentials for the Customer Portal login.

The User determines what Tax Region is in use when Packages and Services associated with the User are billed.

Administrative User

The Administrative User is the User that is able to perform administrative functions through the Customer Portal. This includes but is not limited to:

1. Making Payments
2. Adding Payment Methods

Which User is the Administrative User is specified on the Account.

How To

Flag an Account as Tax Exempt

How to flag an Account as Tax Exempt using the Staff Portal.

1. View the Account
2. Choose No for Taxable in Taxes.
3. Enter the Tax Exempt ID if applicable
4. Click Save in the Context Menu.

Flag Account as Tax Exempt for Specific Tax

This should be done when an Account is normally subject to taxation, but by law may be exempt from a specific tax for a specific type of sale classified in BillMax by Charge Type.
1. View the Account
2. Click on the **Tax Exemptions** tab
3. Click **Add Tax Exemption**
4. Choose the appropriate **Tax Group, Charge Type** and **Reason** for the tax exemption
5. Choose one or more **Tax Items** that the Account is exempt from paying
6. Click **Save**

**Specify Business Rules for a Unique Account**

How to create a custom Account Profile for an Account

This will used when a customer has a need for different billing rules than normal such as terms for payment.

1. From the Main Menu, select **Billing > Account Profiles**
2. Create a new profile
   - If you have an Account Profile that closely resembles the Profile to be created, select and clone the existing Account Profile
   - Otherwise, from the left hand menu click New.
3. Configure the new Account Profile making sure the Availability setting is "Available".
4. Click Save on the left menu.
5. Bring up the unique Account.
6. Change the Account Profile setting to the new Account Profile.
7. Click Save on the left menu.
8. From the Main Menu, select Billing -> Account Profiles
9. Select the new Account Profile.
10. Change the Availability setting to "No New Accounts".
11. Click Save on the left menu.
Chapter 7

Customer Communications

Topics:
- Concepts
- How To
Concepts

Billing Documents

Billing Documents Overview

Billing Documents consist of the following:

- Billing Statements
- Invoices
- Receipts
- Statements

The type of documents that are sent to an Account during Automated Processing is determined by the Account Profile. Choices are:

- Send a Billing Statement
- Send an Invoice
- Send an Invoice along with a Statement show overdue balances.

Billing Documents are rendered as HTML for viewing in a browser or PDF for sending via email as an attachment or printed and mailed.

Billing Documents are created by transforming XML data into HTML for web browsers or XLS:FO which is subsequently transformed into PDF.

The XML and subsequent rendered output is stored in /usr/local/billmax/documents. Logs files of the transformations are stored in /usr/local/billmax/logs and may contain valuable debugging information.

The default XSL transformation files are /usr/local/billmax.cfg/xsltfiles/html/documents.xsl for HTML and /usr/local/billmax.cfg/xsltfiles/fo/documents.xsl for XLS:FO and subsequently PDF.

The optional logo is specified on the Virtual Company.

The transformation occurs only once for a Posted document for performance reasons. As a results, changes to the logo or to the XSL transformation files will not show up on Posted documents that have already been rendered.

Customizing Billing Documents may be done in one of two ways:

2. Creating new XSL Files and changing the Account Profile Document Class to refer to the new XSL file for major changes.

! CAUTION: New files will not be updated by the BillMax upgrade process. This will cause new display functionality to be omitted until the new file is manually updated with the new changes.

Billing Statement

A Billing Statement is a BillMax document that shows activity from the last Billing Statement to the current one determined by the transactions' Post Date. They are most likely used for retail customers. They may contain one or more of the following:

1. Invoice displaying all line items
2. Payment
3. Payment Reversal
4. Refund
5. Refund Reversal

In addition the Billing Statement will include:
1. A beginning balance
2. An ending balance
3. Notes
4. Information about Credit Card or Electronic Check declines

For Accounts associated with an Account Profile marked to send Billing Statements, all sales transactions will have the same due date and the same electronic collection date.

Note: To maintain parallel data processing, Billing Statements are created for customers associated with Account Profiles marked to send Invoices. However, these are not sent automatically to the customer. In addition, if the Billing Statement contains data from more than one Invoice, sales transactions may have different due dates and electronic collection dates. Therefore, although allowable, sending a Billing Statement to an Invoice customer is not advised.

Invoice

An Invoice is a BillMax document that may contain one or more of the following transactions:

1. Sale
2. Store Credit
3. Sales Return
4. Store Credit Reversal
5. Deposit
6. Deposit Reversal

All transactions on an Invoice share the same Effective Date, Post Date and Electronic Collection Date.

Invoices containing Sales and Store Credits

Sales and Store Credits may be grouped on a single Invoice. This is true even if the Sales and Store Credits fall under different Tax Groups.

Within an Invoice, Sales and Store Credits are Allocated to each other until the credits are completely used (if possible) to pay off the debits

Taxes for an Invoice may be assessed per line item on the invoice or after summing all the transactions. The behavior is subject to each Tax Group setting.

Invoices containing Sales Returns and Store Credit Reversals

Sales Returns and Store Credit Reversals are grouped both by type of transaction and original invoice. If there are Sales Returns and Store Credit Reversals from a single invoice, the Sales Returns will be grouped in one Invoice and the Store Credit Reversals will be grouped on another Invoice. This is because the Sales Return Credit is linked to the original Sale and the Store Credit Reversal is linked to the original Store Credit. If there are Sales Returns from multiple Invoices, they will be grouped on multiple Sales Return Invoices. This also occurs for Store Credit Reversals. This helps the end customer track debits and credits to the original Invoice.

Invoices containing Deposits and Deposit Reversals

Deposits are grouped on their own Invoice. This is because Store Credits may not be used to pay off a Deposit. Allowing Deposits and Store Credits on the same Invoice would lead to a misleading display.

Similarly, Deposit Reversals are grouped on their own Invoice and grouped by the original Invoice.

Receipt

A Receipt is a receipt for Payment. Receipts are generated and sent using the BillMax Staff Portal. For customers that receive Billing Statements, a receipt may be redundant.
Statement

A BillMax Statement comes in one of two forms:

1. A list of items that occurred between two dates.
   a. Invoices
   b. Payments
   c. Payment Reversals
   d. Refund
   e. Refund Reversals

2. A list of "Open" items. Items that have not been paid off or used up.
   a. Invoices
   b. Payments
   c. Refunds

Account Profiles may be configured such that Automated Processing send Statements of "Open" items to Invoice customers if an "Open" item is overdue. Statement may be generated and sent using the BillMax Staff Portal.

Email

Email Overview

BillMax uses email in many ways to communicate the following to customers:

- Billing Statements, Invoices, etc. - billing related documents.
- Ticket messages and notifications - see Ticketing on page 211 for further information.
- System Email Templates - emails that are automatically system generated such as overdue notices.
- Customer Notices - emails created by a BillMax Authorized User and sent to a group of customers.

Billing Documents

Billing documents by default are emailed as PDF attachments. These include:

- Billing Statements
- Invoices
- Statements
- Receipts

If they exist, the body of the email will be created using the files /usr/local/billmax/cfg/xsltfiles/emailbody/html.xsl for an HTML representation and /usr/local/billmax/cfg/xsltfiles/emailbody/text.xsl for a text representation.

Customer Notices

Notices may be emailed to customers using both text and HTML. There are two parts to doing this:

1. Identifying the customers that will receive the notice. This is accomplished using SQL to create a selection.
2. Creating the email template of the message that also uses data from the SQL.

System Email Templates

System Email Templates are templates for emails that are used by BillMax for pre-programmed purposes. These include but are not limited to:

- Overdue notices.
- Credit card decline notice
- Customer Portal password reset notice

Click on Correspondence > System Email Templates to see a complete list.
How To

Add Additional Late Notices

BillMax supports up to 10 late notices named "overdue1", "overdue2", overdue3", etc. Sequential numbers are required.

1. Select **CorrespondenceSystem Email Templates** from the Top Menu.
2. Choose the template **overdue1**.
3. Select **Other > Clone** from the Context Menu.
4. Change the **Name** "overdue1_CLONE" to the desired "overdue#".
5. Change the **Template**.
6. Select **Save** from the Context Menu.

Add HTML View to Body of Email for Invoices and Billing Statements

1. Login into the BillMax server as root.
2. `su` to the BillMax system user, typically billmax.
3. `cd /usr/local/billmax/cfg/xsltfiles/emailbody`
4. `cp html.xsl-dist html.xsl`
5. Verify results by using BillMax to send an Invoice and Billing Statement.
6. Using the email client, check the HTML only view of the email.

Add Text View to Body of Email for Invoices and Billing Statements

1. Login into the BillMax server as root.
2. `su` to the BillMax system user, typically billmax.
3. `cd /usr/local/billmax/cfg/xsltfiles/emailbody`
4. `cp text.xsl-dist text.xsl`
5. Verify results by using BillMax to send an Invoice and Billing Statement.
6. Using the email client, check the text only view of the email.

Create/Edit an Email Template

Email Templates are entered via an HTML **textarea** widget.

1. Access the **textarea** for the email template.
2. Add or edit the email header lines. These must include:
   - **To**:
   - **From**:
   - **Subject**:
3. Other email headers such as **CC**: may be specified
4. Enter a blank line after the last email header line.
5. Enter the text body of the email message. The text body is required.
6. **Warning**: HTML must be well formed in XHTML format. The text entered is validated as to well-formed XML, but the XML elements are not checked to see if HTML.

**Tip**: When using an **<img>** tag, if the **src** attribute begins with **http:** or **https:**, then the **img** tag is not modified. If the **src** attribute is anything else, it is considered an image file located somewhere in the /usr/local/billmax directory and is added as an embedded image.
Optionally enter the HTML body of the email message. This is done by entering the `<html>` tag on a line after the text body plus the rest of the HTML.

```
To: ${TO}
From: ${FROM}
Subject: Outage

We are having an outage

Billing Department
<html>
<body>
  <img src="https://www.example.com/images/logo.jpg"/> 
  <img src="/images/logos/logo.jpg"/> 
  <h1>We are having an outage</h1>
  <p>Billing Department</p>
</body>
</html>
```

**Customize a Billing Document (PDF)**

XML for documents are stored in `/usr/local/billmax/documents` by date. Choose a file containing the document type you wish to modify by examining the contents of the file.

3. Run the commands "xsltproc -o xmlfile.fo documents.xsl xmlfile; /usr/local/fop/fop -c fop.conf -q -fo xmlfile.fo -pdf xmlfile.pdf" where `xmlfile` is the input XML file copied from `/usr/local/billmax/documents`.
4. View the resulting PDF document (assuming no errors).
5. When satisfied with PDF, do one of the following:
   - copy edited `documents.xsl` to `/usr/local/billmax/cfg/xsltfiles/fo/documents.xsl`
   - implement a new `documentclass`
     1. Copy edited `documents.xsl` to `/usr/local/billmax/cfg/xsltfiles/fo/newnamedocuments.xsl`
     2. Add `newname` to the BillMax List `documentclass`.
     3. For each Account Profile that will use the new `documentclass`, set the Document Class to `newname`.
6. **Warning:** PDF files are stored for performance reasons. When testing a new XSL file, you must either have pending transactions and be generating trial documents, or be changing some data like due date in a posted document to see the results of the new XSL file. Changing the document class will also generate a new HTML and PDF.

   Test PDF generation from interface.

**Customize a Billing Document (HTML)**

XML for documents are stored in `/usr/local/billmax/documents` by date. Choose a file containing the document type you wish to modify by examining the contents of the file.
2. Run the commands "xsltproc --output xmlfile.html --xinclude documents.xsl xmlfile" where xmlfile is the input XML file copied from /usr/local/billmax/documents.
3. View the resulting HTML document (assuming no errors).
4. When satisfied with HTML, do one of the following:
   • Implement a new documentclass
     2. Add newname to the BillMax List documentclass.
     3. For each Account Profile that will use the new documentclass entry, set the Document Class to newname.
5. Warning: HTML files are stored for performance reasons. When testing a new XSL file, you must either have pending transactions and be generating trial documents, or be changing some data like due date in a posted document to see the results of the new XSL file. Changing the document class will also generate a new HTML view.

Test HTML generation from interface.

Customize the Credit Card Decline Notice

1. Select Correspondence > System Email Templates.
2. Choose the template ccdecline.
3. Note: The following variable are available. Use either "$variable" or "${variable}" notation.

   TO The Account Email Address.
   FROM The Billing Email Address of the Virtual Company associated with the Account.
   SUBJECT The phrase "Company Account ##:w!" where Company is the name of the Virtual Company and ## is replaced by the Account number.
   LAST4 Last 4 digits of the credit card number.
   AMOUNT The amount that was to be collected.
   REASON Always "DECLINED".
   config.* fields from the Virtual Company record to which the Account belongs. See config Table (Virtual Company) on page 177

Modify the Template.
4. Select Save from the Context Menu

Customize the Default Late Notice

1. Select Correspondence > System Email Templates.
2. Choose the template overdue1.
3. Note: The following variable are available. Use either "$variable" or "${variable}" notation.

   TO The Account Email Address.
FROM

The Billing Email Address of the Virtual Company associated with the Account.

SUBJECT

The phrase "account ## is past due" where ## is replaced by the Account number.

DAYS

The minimum number or days overdue that will trigger the last notice.

MAXDAYSOVERDUE

The maximum number of days overdue for any overdue amount.

AMOUNT

The amount overdue.

BALANCE

The Account balance.

NOTICE

The total number of late notices sent including the current one.

Modify the Template.

4. Select Save from the Context Menu

Send A Mass Email from BillMax

1. Create a Memorized Report to generate the Email addresses:
   a) Select Correspondence > Memorized Report.
   b) Select New.
   c) Enter Description.
   d) Enter Query. The email field to which the email will be sent must be aliased as 'TO'. A 'FROM' field is required, but its use in the email template is optional. The following selects unique emails from Users that are not Closed:

   ```sql
   SELECT DISTINCT email AS 'TO', 'billing@example.com' AS 'FROM'
   FROM user WHERE email != '' AND user.state!=1
   ```
   e) Select Save from the Context Menu.

2. Create the Email template:
   a) Select Correspondence > Customer Notices.
   b) Select New.
   c) Enter Name.
   d) Enter Description.
   e) Choose Memorized Report created in step 1.
   f) Enter Template. The following is an example for sending an outage notice.

   ```
   Note: Fields from the query may be inserted into the template using the "${variablename}" notation.
   
   Note: HTML presentation may be sent as well as text presentation.
   
   To: ${TO}
   From: ${FROM}
   Subject: Outage
   
   We are having an outage
   
   Billing Department
   <html>
   <body>
   ```
We are having an outage
Billing Department

Processing in TEST mode...

Using Template [
To: ${TO}
From: ${FROM}
Subject: Outage

We are having an outage
Billing Department

Number of rows in result: 4

------ processing Troy.Beasley@example.com [
To: Troy.Beasley@example.com
From: billing@example.com
Subject: Outage

We are having an outage
Billing Department

We are having an outage
Billing Department

------ processing Shari.Acosta@example.com [
To: Shari.Acosta@example.com
From: billing@example.com
Subject: Outage

We are having an outage
Billing Department
We are having an outage
Billing Department

]       processing Julia.Mercer@example.com [ 
To: Julia.Mercer@example.com
From: billing@example.com
Subject: Outage
We are having an outage
Billing Department

We are having an outage
Billing Department

]       processing Matilda.Merrill@example.com [ 
To: Matilda.Merrill@example.com
From: billing@example.com
Subject: Outage
We are having an outage
Billing Department

We are having an outage
Billing Department

Tested 4 rows

4. Select **Process** to immediately send the emails.
Chapter 8

Financials

Topics:

- Concepts
Concepts

Transactions

BillMax supports the following financial transactions:

- Sale
- Payment
- Store Credit
- Refund
- Deposit
- Sales Return
- Payment Reversal
- Store Credit Reversal
- Refund Reversal
- Deposit Reversal
- Sales Write Off
- Payment Write Off
- Store Credit Write Off
- Refund Write Off
- Sales Write Off Reversal
- Payment Write Off Reversal
- Store Credit Write Off Reversal
- Refund Write Off Reversal
- Beginning Balance

The scope of transactions enable customer accounts to be continually updated in the present and not having to make adjustments in the past. This preserves financial history.

Sale

Sales transactions may classified into the following:

- Recurring
- One Time

Sales transactions are used to add debits to a customer for services delivered, items purchased, etc. Sales may have tax consequences. Sales may be paid by Payments or Store Credit.

Payment

Payments are the primary means of adding credit to a customer and represent customer payments for services delivered, items purchased, etc. Payments are considered by definition to be made using legal tender through the use of cash, check, credit card, eCheck, etc. As a method of fraud prevention, partial or full amounts from unused Payments are the only amounts that may be used to issue a Refund.

Store Credit

Store Credits may be used to give a customer non legal tender credit. Store Credits may have tax consequences. The recommended use of Store Credits is to credit a customer for referral, for buying a long term contract, discount code, etc. It is not recommended that they be used for service not delivered or for the return of items purchased. A Sales Return should be used for this purpose.
Refund
Refunds are a debit to a customer and represent the voluntary transfer of legal tender back to the customer. Only unused amounts from Payments may be used to create a Refund.

Tip: Unused Store Credit amounts are not refundable. This is a fraud prevention measure. If an Account has unused Store Credits and Payments that are used for Sales, Store Credits may be used to replace the Payment leaving unused Payment amounts available for a Refund.

Deposit
Deposits are a debit to a customer. Deposits may be paid by Payment only. The intent of the Deposit transaction is to collect money that will eventually be returned to the customer. Typically a Deposit is collected as a surety against a piece of equipment. Before amounts for an Account may be written off, all Deposits must have a corresponding Deposit Reversal.

Sales Return
Sales Returns are a credit to the customer. Sales Returns are linked to Sales. Sales Returns may be partially returned by setting a lower quantity than the original Sale, setting a shorter time span than the original Sale, or both. Sales Returns are tracked to the penny against the original Sale to ensure that multiple Sales Returns against a single Sale will not exceed the original Sale, both in Sale amount and tax amounts. Sales Returns and Store Credit Reversals are the only ways to make tax adjustments. This ensures that tax reporting of taxable versus non taxable amounts is accurate.

Payment Reversal
Payment Reversals are used to book legal tender debits to a customer that are not Refunds. Examples of these are charge-backs for credit cards and NSF (insufficient funds) for checks.

Store Credit Reversal
Store Credit Reversals are a debit to the customer. Store Credit Reversals are linked to Store Credits. A partial Store Credit Reversal may be created by setting a lower quantity than the original Store Credit, setting a shorter time span than the original Store Credit, or both. Store Credit Reversals are tracked to the penny against the original Store Credit to ensure that multiple Store Credit Reversals against a single Store Credit will not exceed the original Store Credit, both in Store Credit amount and tax amounts. Store Credit Reversals and Sales Returns are the only ways to make tax adjustments. This ensures that tax reporting of taxable versus non taxable amounts is accurate.

Refund Reversal
Refund Reversals are a credit to the customer. In general these are used to fix erroneous Refunds. In addition, a customer may have been given two refunds against a single Payment. An example of this may occur when a Refund by check is entered into BillMax against a credit card payment. Later that payment is returned by the credit card company in the form of a charge back. When the Payment is reversed using a Payment Reversal and the check is returned by the customer, a Refund Reversal will be appropriate.

Deposit Reversal
Deposit Reversals are used when monies collected for a Deposit are to be returned to the customer. Deposit Reversals are also necessary when writing off a customer balance.

Write Off
Write Offs are used to write off the balance of a customer.
Write Off Reversal

Write Off Reversals are used to reverse the effect of Write Offs. Generally these are used when a customer returns after the balance has been written off.

Beginning Balance

Beginning Balance transactions are typically used when adding a customer to BillMax and they have a balance being carried over from another billing system. Beginning Balances may represent Sales along with taxes, Payments, Store Credits along with Taxes and Refunds (over refunded amounts). Beginning Balances do not show in any financial transaction reports. However, the effects of subsequent transactions against a Beginning Balance will show up in financial transaction reports.

Beginning Balances may also be used to create an account adjustment. This is not recommended as the adjustment does not show in financial transaction reports, but it is available.
Chapter 9

Payments and Refunds

Topics:

- Concepts
- How To
## Concepts

### Payment Processing Overview

Lists are used to configure merchant accounts in BillMax, specifically credit card processing and eCheck processing. There are a number of settings for credit card and eCheck processing with most of the items having preset values. A specific setup for each Virtual Company can be created. Some settings are critical and must be set. Here is the list of critical values:

**Table 1: Payment Processing Settings**

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Company</td>
<td>All</td>
<td>All will allow the list to be used for All Virtual Companies. Otherwise, specify the number of the Virtual Company the list applies to.</td>
</tr>
<tr>
<td>Test</td>
<td>0</td>
<td>0 indicates the list is active for production. 1 is the test mode where transactions are fake.</td>
</tr>
<tr>
<td>ResultEmail</td>
<td><a href="mailto:admin@example.com">admin@example.com</a></td>
<td>The address where the batch processing log will be emailed.</td>
</tr>
<tr>
<td>State</td>
<td>Active</td>
<td>Active indicates this list is in use for the specified Virtual Company and PayType. Not Active means the list is not in use.</td>
</tr>
<tr>
<td>PayType</td>
<td>Credit</td>
<td>This value must be set to either &quot;Credit&quot; or &quot;Echeck&quot;.</td>
</tr>
<tr>
<td>UserName</td>
<td>ZX1GHTYU1234</td>
<td>The TID or Terminal ID provided by the merchant service provider is entered as the UserName.</td>
</tr>
<tr>
<td>Tokenization</td>
<td>1</td>
<td>0 means that tokenization will not be used. Otherwise, 1 specifies that tokens will be generated as needed and used.</td>
</tr>
<tr>
<td>Module</td>
<td>ippay</td>
<td>This entry indicates the BillMax module to use.</td>
</tr>
</tbody>
</table>

Out of the box, BillMax provides two lists for each processor it supports: one for credit cards and one for e-checks (bank debit). To use a simple setup where only one merchant account is used for all virtual companies, locate the appropriate lists and configure the settings mentioned above. To use multiple merchant accounts, create two new lists for each Virtual Company if credit cards and e-checks are accepted. As each list contains a couple dozen settings, use the `copylist.pl` script from the command line as follows:

```
/usr/local/billmax/bin/copylist.pl <from list name> <new list name>
```

Edit the new list to be used for the new Virtual Company. Terminal IDs can be used for more than one list.

**Note:** If the copied list is Active, there is a brief period where the two lists refer to the same Virtual Company. Subsequent charge/refund attempts will fail. Copy the lists at a time where it will not be used.

**Note:** If the "from" list name is the same as the module name, the `copylist.pl` script will rename the module.
To check the overall configuration, view System Administration > System Status. It is also a good idea to test a single transaction for a customer in the appropriate Virtual Company before the nightly batch runs.

Lastly, BillMax payment processing can be configured to run for only certain Virtual Companies. See Configure Payment Processing by Virtual Company on page 73.

Related tasks
Configure Payment Processing on page 20
Set parameters for Payment Processing in BillMax. Remember there is a list for credit cards and one for e-check for each processor.

Electronic Payment and Refund Transactions

Electronic Payment and Refund Transactions are payments and refunds initiated by BillMax using a third party processor. Transactions type submitted to the third party processor are:

- Credit Card
- Electronic Check
- NACHA files

Credit card processing of course involves debit or credit of a customer's credit account whereas Electronic Check achieves the same with a customer's bank account generally through the Automated Clearing House network (ACH). A successful Credit Card payment transaction (barring a charge back) guarantees the receipt of funds at the time of the transaction. Electronic Check processing does not because of the asynchronous nature of ACH. It might take several days before a merchant knows whether the payment or refund was actually successful.

Credit Card and Electronic Check Processing Modes

BillMax processes electronic transaction either on a demand basis within the web interface or automatically as part of the nightly batch processing.

Whenever a Credit Card or Electronic Check transaction is attempted, a record of the attempt is created. These records may be viewed on an Account's Billing > Electronic Payment Activity tab.

Credit Card and Electronic Check

When BillMax submits a Credit Card or Electronic transaction request the result will be captured in BillMax as:

- Approved
  
  A Payment or Refund is entered into BillMax.

- Rejected

- Held
  
  Used processing through Authorize.Net. Part of Authorize.Net's fraud detection technology. See Resolve a Credit Card or Electronic Check Transaction Flagged as Held on page 77

- Unknown
  
  The transaction was submitted to the processor, but no valid response, if any, was received. See Resolve a Credit Card or Electronic Check Transaction Flagged as Unknown on page 77

Settled Transactions

When a transaction is submitted to a processor and it is approved, the transaction will be not be "Settled" from the processors perspective. Even though the transaction has been approved, no process has been started by the processor to transfer any funds. During this period, a transaction may be voided at the processor, the the end customer will never see the transaction on their credit card or bank statement. See Correct a non-Settled Credit Card or Electronic Check Payment on page 76.

If a transaction has been settled then either an offsetting Refund or Payment will need to be made. See Correct a Settled Credit Card or Electronic Check Payment on page 77
NACHA files

Using NACHA files operates on the premise that the approval or rejection of a transaction will take place sometime in the future relative to booking a Payment in BillMax for the customer. Therefore, if BillMax is setup for NACHA processing and an Electronic Check transaction is created for a customer the following happens:

1. A Payment or Refund is entered into BillMax for the Account. The Payment or Refund is approved by definition.
2. A record is entered into the NACHA file for future upload.

Transactions in a NACHA file that are subsequently rejected will have to go through the Reversal process. See Book a Payment Reversal on page 76.

Processing Check Files

Checks can be entered (posted) in a number of ways. When the number of checks to process is low, the customer service representative may locate the customer’s account and post the checks by data entry on each of the respective account pages. See Book a Payment while Viewing an Account on page 76.

When the number of checks to process is numerous, the customer service representative may instead use a special batch payment entry interface. This interface is designed to limit the navigation and entry so the process is easier and faster to perform. See Book Payments for Multiple Accounts on page 76.

BillMax can also load checks from payment files that are either provided by the external vendors (a bank's "lock box" service for instance) or derived from a check scanning process.

Scanned Checks

BillMax also offers integration with check scanners. When using a check scanner, a payment file is produced by the scanning process then uploaded to BillMax for processing. Presently BillMax supports file formats produced by Magtek®. In particular, see http://www.magtek.com/product/micrsafe.

The scanned data may be processed in one of two ways:

- Check data is entered and payments posted as if manually entered. Checks are then deposited in a bank account.
- Check data is converted to electronic checks and payment is collected via electronic check processing. This may result in speedier deposits and avoid trips to the bank.

Important: To use the check conversion process, your customers must be informed that their checks may be converted to electronic checks.

Any electronic check payment method entered using the conversion process will be marked as such in the interface. The electronic payment entry will show the original check number for reference on the Account's Billing tab. These records should not be used for Auto Pay unless prior approval has been given from the customer.

The scanning process captures the check number, the bank account, and the bank routing number. The payment amount and the account number must be entered manually. To aide this entry, when using the check conversion option, BillMax will save the scanned and entered data as an electronic check payment method record if no record exists for that bank account. These records are then used to process the checks electronically and provide a lookup of account and balance for subsequent check scans.

Related tasks

Book Payments from Scanned Checks on page 76

Vendor Supplied Payment Files

Presently BillMax supports payment files from Pollux Systems® (www.polluxsystems.com) and ProfitStars® (www.profitstars.com). Other vendor support is possible with minor programming. These companies receive checks on your behalf, make the bank deposits, and provide payment files of the processed checks.

BillMax provides the ‘loadcheckfile.pl’ script which can be used to receive check files (via FTP) from these vendors. This script is typically installed as a BillMax batch process where it is run once a day.

Warning: The script must be edited so the site, username, password, and vendor are specified.
Once files are loaded via this process, the check payments are posted via the ‘Upload Checks’ menu item under Billing Administration.

**How To**

**Add a Credit Card to an Account**
How to add a Credit Card to an BillMax Account using the Staff Portal

1. View the Account.
2. Select **Add > Credit Card** from the Context Menu.
3. Enter the data.
4. Click **Save**.

**Configure Payment Processing by Virtual Company**
The default is for both BillMax EFP Credit Card processing and BillMax EFP E-check processing are to be ACTIVE steps in the nightly Batch Processing for all Virtual Companies. The steps below will limit the processing to certain Virtual Companies.

1. Select **System Administration > Batch Processing**.
2. Choose the number for **nightly**.
3. Choose the number for **BillMax EFP Credit Card Processing**.
4. Modify the Arguments setting to include `-C x,y,z` where x, y, and z are the numbers for the Virtual Companies to process.
5. Repeat for **BillMax EFP Credit Card Processing** if needed.

**Create a Bank Deposit**

1. Select **Billing Administration > Bank Deposits** from the Main Menu.
2. Select **New** from the Context Menu.
3. Choose the **Bank Account** to which the deposit was or will be made.
4. Enter the **Date of Deposit**. This is the date that will be on the bank account statement.
5. Retrieve all possible transactions that go into the Deposit
   a) Choose the **Company** if bank deposits are separated by Virtual Company. Otherwise choose **ALL**.
   b) Choose the **Tender Types**.
   c) Choose the **Transaction Type**. Use **ALL** unless there is a good reason not to.
   d) Enter **Beginning Post Date** of the transactions for the deposit.
   e) **Note**: If deposits typically include charge backs or electronic check NSF, make sure the ending postdate will include those transactions.
   f) Click **Get Transactions**
6. Choose the Transactions that make up the Bank Deposit. Use Select All and Clear From Row to do mass selection/de-selection.
7. Click **Save** on the Context Menu. The display stays on the Bank Deposit page. Once the top label changes from New Bank Deposit, using the Context Menu you may use the Spreadsheet or Print Friendly options to create a deposit report.

**Creating Bank Deposits from IPPay® data**
Creating Bank Deposits from IPPay® will be successful if and only if all transactions recorded at IPPay® are also in BillMax. Try to do all Payments and Refunds via credit card and electronic check processing using BillMax. Try not to use the IPPay® portal to create transactions and then record the transaction in BillMax.
Use notifications of charge backs and electronic check NSF transactions to enter Payment Reversals. See *Book a Payment Reversal* on page 76.

**IPPay® Settlement Files**
Use either CSV files received via email from IPPay® or XLS files downloaded from the IPPay® website to create bank deposits.

1. Select **Billing Administration > Bank Deposits** from the Main Menu.
2. Select **New** from the Context Menu.
3. Click **Settlement File**.
4. Select the **Bank Account** for the deposit.
5. Click **Choose Files** to upload. Only one file at a time is allowed.
6. Click **Upload**.

If eligible transactions are in the settlement file (not all settlement files result in a Bank Deposit), a new Bank Deposit will be created with a best effort at the **Date of Deposit**. The Bank Deposit may be viewed and adjusted by clicking **Index** and selecting the newly created Bank Deposit.

**Note:** Not all settlement files will create a Bank Deposit that matches an listing in a bank statement. In some cases the settlement file will create one Bank Deposit while the bank statement may list multiple deposits. It is recommended the result be accepted unless it is trivial to split up the newly create Bank Deposit.

**Note:** Another issue is that sometime the settlement file does not contain all transactions. This can be addressed by editing the newly created Bank Deposit and adding the missing transactions listed in BillMax.

**IPPay® Credit Card Charge Backs**
Notifications of charge backs are transmitted via email and fax. Use the data received to enter charge backs into BillMax.

1. Identify the Account and amount/transaction from the chargeback data.
2. Create Payment Reversals for each chargeback. See *Book a Payment Reversal* on page 76.
3. Using data from a bank statement, create a Bank Deposit using Payment Reversals of which the sum equals that listed on the Bank Statement. The date of the Bank Deposit should be that listed on the bank statement.

**IPPay® Electronic Check NSF transactions**
Notifications of NSF for Electronic Check are transmitted via email and fax. Use the data received to enter NSF transactions into BillMax. Notifications will show transactions that were funded and then found to be NSF, and transactions that were found to be NSF before funding. The distinction is important in step 3.

1. Identify the Account and amount/transaction from the NSF data.
2. Create Payment Reversals for each chargeback. See *Book a Payment Reversal* on page 76.
3. Using data from a bank statement, create a Bank Deposit using Payment Reversals of which the sum equals that listed on the Bank Statement. The date of the Bank Deposit should be that listed on the bank statement. Use only NSF transactions that were funded. Unfunded NSF transactions will be included in a Bank Deposit with the original payment transactions.

**IPPay® Credit Card Chargeback Reversal**
In rare cases, notification of a chargeback reversal is received.

1. Identify the Account and amount/transaction from the notification.
2. If the Account had the amount written off, reverse the Write Off.
3. Enter a new Payment manually selecting the correct tender type. See *Book a Payment while Viewing an Account* on page 76 or *Book Payments for Multiple Accounts* on page 76.
5. Optionally create a Bank Deposit using data from a bank statement showing receipt of chargeback reversal funds. See *Create a Bank Deposit* on page 73.
Work with Payments

Automate E-check Returns

Use these steps to setup E-check Returns. Currently only available for IPPay payment processing.

1. Modify the Correspondence > System Email Templates > achdecline email template if desired.
2. Create a Billing Administration > One Time Item/Fee if a fee will be charged for a e-check return.
3. Create or modify a ticketing queue and create an issue for E-check Returns at Correspondence > Queues if a ticket will be created and tracked for a e-check return.
4. In addition to the normal settings for this list, edit Billing Administration > Lists > IPPay-echeck for E-check Return settings. Below is an example of possibilities.
5. Enable the batch process at System Administration > > Batch Processing > echeckreturn.
6. To test, Run Process and check System Administration > Log Files to view results.

<table>
<thead>
<tr>
<th>Table 2: IPPay e-check List entries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Item</strong></td>
</tr>
<tr>
<td>DisableEcheckAutoPay</td>
</tr>
<tr>
<td>EcheckReturnFeeDefinition</td>
</tr>
<tr>
<td>EcheckTicketQueue</td>
</tr>
<tr>
<td>EcheckTicketProblem</td>
</tr>
<tr>
<td>EcheckRetryDays</td>
</tr>
</tbody>
</table>

Manually Process a Credit Card

How to manually charge a credit card using the BillMax Staff Portal

1. View the Account.
2. Select Payment > Process Credit Card from the Context Menu.
3. Choose the Payment Information To Use.
4. Enter the Amount to charge. The default is the total amount owed by the customer.
5. Fill in any other relevant data. Most of the time the defaults may be accepted.
6. Make sure that the appropriate Debits under Allocations are selected if the Amount to charge was changed.
7. Click one of the Process Credit Card buttons depending on your needs.

Manually Process an Electronic Check

How to manually draft a bank account via an electronic check using the BillMax Staff Portal

1. View the Account.
2. Select Payment > Process Electronic Check from the Context Menu.
3. Choose the Payment Information To Use.
4. Enter the Amount to charge. The default is the total amount owed by the customer.
5. Fill in any other relevant data. Most of the time the defaults may be accepted.
6. Make sure that the appropriate Debits under Allocations are selected if the **Amount to charge** was changed,
7. Click one of the **Process Electronic Check** buttons depending on your needs.

**Book a Payment while Viewing an Account**
How to book a cash, check or other payment received outside of BillMax using the BillMax Staff Portal while viewing an Account.

1. View the Account.
2. Select **Payment > Book Payment Received** from the Context Menu.
3. Choose the **Payment Information To Use**.
4. Enter the **Amount**. The default is the total amount owed by the customer.
5. Fill in any other relevant data. Most of the time the defaults may be accepted.
6. Make sure that the appropriate Debits under Allocations are selected if the **Amount** was changed,
7. Click one of the **Save** buttons depending on your needs.

**Book Payments for Multiple Accounts**
How to rapidly book cash, check or other payments received outside of BillMax using the BillMax Staff Portal.

1. Click **Payment** on the Top Menu.
2. Open a Payment dialog:
   - Select search criteria.
   • Enter the search data and press Enter or Tab.
3. Fill in the form.
4. Save the Payment by pressing the Enter key while in any text field or by clicking a **Save** button.

**Book Payments from Scanned Checks**

1. Scan the checks into a text file
2. Upload the scanned checks using **Billing Administration > Upload Scanned Checks**
3. Complete or edit the payment data in the table.

   **Note:** BillMax receives the data and queries on the bank account and bank routing number. Where matches are found, BillMax fills in the Account number and balance for the payment amount. Otherwise, the customer last name and street address are included so that a CSR can search for the account.
4. Select **Process as eChecks** to convert the checks and process the payments or select **Book Payments** if checks will be deposited manually. The effective date and bank branch are only valid if **Book Payments** is used.

**Book a Payment Reversal**
How to book an NSF Check or Credit Card Chargeback

1. View the Account.
2. Click on the **Billing Tab**.
3. Select the Payment by clicking the number in the Number column.
4. Click on the **Book Reversal** tab in the opened dialog.
5. Fill in the form.
6. Click **Save Reversal**.

**Correct a non-Settled Credit Card or Electronic Check Payment**

1. Verify with the Credit Card/Electronic Check Processor that the Payment has not settled.
2. View the Account.
3. Click on the **Billing Tab**.
4. Select the Payment by clicking the number in the Number column.
5. Click **Processor Void**.
6. Click Yes if acceptable.
7. If necessary, reprocess the Credit Card or Electronic Check for the correct amount.

**Correct a Settled Credit Card or Electronic Check Payment**
The Payment must be settled.

1. View the Account.
2. If need be, follow the steps Create Refundable Amounts on page 78.
3. Follow the step in Manually Process a Credit Card Refund on page 78 or Manually Process an Electronic Check Refund on page 78
4. If necessary, reprocess the Credit Card or Electronic Check for the correct amount.

**Resolve a Credit Card or Electronic Check Transaction Flagged as Held**
Use these steps when an Electronic Payment or Refund Transaction is flagged as Held.

1. View the Account.
2. Select Billing > Electronic Payment Activity
3. Identify the Held transaction looking for "Held" under the Approved? column.
4. Approve or decline the transaction at Authorize.Net®
5. Approve or decline the transaction in BillMax
   a) Click the relevant link under the Number column.
      The Electronic Transaction dialog will be displayed.
   b) Enter Authorization Code if available.
   c) Enter Processor Id if available.
   d) Click Accepted or Declined.
      A dialog asking to continue will be displayed.
   e) Click Yes.
      If accepted, a pre-populated Payment dialog will be displayed.
   f) Enter any extra data in the Payment dialog.
   g) Click Save.

**Resolve a Credit Card or Electronic Check Transaction Flagged as Unknown**
Use these steps when an Electronic Payment or Refund Transaction is flagged as Unknown.

1. View the Account.
2. Select Billing > Electronic Payment Activity
3. Identify the Unknown transaction looking for "Unknown" under the Approved? column.
4. Verify with the processor whether or not the transaction was successful.
5. Approve or decline the transaction in BillMax
   a) Click the relevant link under the Number column.
      The Electronic Transaction dialog will be displayed.
   b) Enter Authorization Code if available.
   c) Enter Processor Id if available.
   d) Click Accepted or Declined.
      A dialog asking to continue will be displayed.
   e) Click Yes.
      If accepted, a pre-populated Payment dialog will be displayed.
   f) Enter any extra data in the Payment dialog.
   g) Click Save.
Work with Refunds

Create Refundable Amounts
Create Refundable Amounts Using Sales Returns
See Refundable Amount on page 37

1. Create Sales Return Invoices using paid off Sales up to the amount to Refund. See Do a Sales Return or Store Credit Reversal on page 42.
2. Verify the Refundable Amount is correct.

Create Refundable Amounts Using Allocations
Sales have been paid off by any means but Store Credit, but no desire to do a Sales Return. Usually done when customer requested to do a partial payment but a full payment was collected. See Refundable Amount on page 37

1. View the Payment to be refunded.
2. Choose No for Auto Allocate.
3. Choose the Allocations to allocate the part to the payment that will not be refunded if a partial refund is to be made
4. Click Save.
5. Verify the Refundable Amount in the Account Overview is correct.

Create Refundable Amounts using Store Credits
The Account has an overall negative balance and the Refundable Amount is $0.00. This indicates that the negative balance is composed of Store Credits which are not Refundable. See Refundable Amount on page 37. When the following steps are completed, whatever Sales were paid off by selected Payments will be paid off by Store Credits.

1. Select Store Credit > Existing Store Credit to Refundable Cash.
2. Enter Amount. The maximum available is the maximum of the Account balance and the sum of Payments that have not been refunded and are not being used to pay off Deposits.
3. Choose the Eligible Payments and enter the Exchange Amount if necessary.
4. Click Save.

Manually Process a Credit Card Refund

1. View the Account.
2. Verify that the Refundable Amount in the Account Overview is negative.
3. Select Refund > Refund via Credit Card from the Context Menu.
4. Choose the payment(s) to refund. The default is to refund all available credit card payments. If no payments are available, then no credit card payment is available to refund.
5. Enter Amount to Refund if a partial refund of the amount is to be made.
6. Click Refund to Credit Card.

Manually Process an Electronic Check Refund

1. View the Account.
2. Verify that the Refundable Amount in the Account Overview is negative.
3. Select Refund > Refund via Electronic Check from the Context Menu.
4. Choose the payment(s) to refund. The default is to refund all available electronic check payments. If no payment is available, then no electronic check payment is available to refund.
5. Enter Amount to Refund if a partial refund is to be made.
6. Click Refund to Credit Card.

Book a Refund while Viewing an Account

1. View the Account.
2. Select Refund > Book Refund Made from the Context Menu.
3. Enter the data.
4. Click Save.

**Correct a Settled Credit Card or Electronic Check Payment**

The Payment must be settled.

1. View the Account.
2. If need be, follow the steps Create Refundable Amounts on page 78.
3. Follow the step in Manually Process a Credit Card Refund on page 78 or Manually Process an Electronic Check Refund on page 78.
4. If necessary, reprocess the Credit Card or Electronic Check for the correct amount.
Chapter 10

Products

Topics:

• Concepts
• How To
Concepts

Product Overview

Products are items and services sold to and/or provisioned for the customer. Products are managed by the following:

1. Product Definitions that specify what Products are available for sale or provisioning. These referred to and defined by:
   - One Time Item/Fee Definition
   - Recurring Service Fee Definition
   - Recurring Service Definition
   - Recurring Package Definition

2. Products that are being sold or provisioned on a recurring basis. These are based on the Product Definitions and are added to each Account at the User level. When added to a User, these are referred to and defined by:
   - Packages
   - Services

In addition, groups of Packages and Services may be added to a User by using a Bundle.

Attention: When adding to a User, the list of available Package, Service and Bundle Definitions presented to the CSR may be filtered by the Account Class and Technology settings. In addition to those Definitions where the Account Class and Technology are not specified, only those that match the Account Account Class setting and the User Technology setting will be available to the CSR. This is presentation only. After creation no checking is done.

In addition to Product Definitions, Products sold may have a usage component that will be billed in addition to the recurring price. Billing for the usage component may be defined by adding a Usage Tier Plan to a Recurring Package or Service Definition. In the case of VOIP billing, a VOIP plan may be added to a Recurring Package or Service Definition. See VOIP/Telecommunication on page 249 for further information on VOIP billing.

The best use of Package and Service Definitions is to use them for both provisioning and billing. Billing is used to bill for the services delivered. Provisioning is used to enable or disable the services. Provisioning may be configured such that as the status of a service changes, for example due to the service ending or payment for the service is overdue, access to the service may be disabled or enabled.

Bundle Definition

A Bundle Definition is a convenient single selection feature that defines a group of Package Definitions, Service Definitions and One Time Item/Fee Definitions to use to add Packages and Services to a User and assess the One Time Item/Fee amounts.

One Time Item/Fee Definition

A One Time Item/Fee Definition is created to define pricing for something that is sold on a non-recurring basis. Examples are:
   - Consulting
   - Equipment

One Time Item/Fee Definitions may be added to Packages, Service or Bundle Definitions. They may also be limited to use in Packages Definitions only.

Package Definition

A Package Definition is the same as a Service Definition with one major difference. A Package Definition may include the following:
   - One Time Item/Fee Definitions
• Recurring Service Definitions

For included Definition, the following may be specified:

• A minimum number of products to be added. When a Package is added to a User, One Time Item/Fee Definitions will be assessed the minimum number of times and the minimum number of Services will be added to the Package with Status of Prospective.
• A maximum number of Services to be added. This is not enforced. However, a CSR is notified when they are about to exceed the maximum number.
• Different pricing than that specified on the Service Definition. This enables Package level pricing for a Package of Services.

Services belonging to a Package are affected by the change in Status of a Package. For example, if a Package is Open and is changed to Closed, the non-closed Services belonging to the Package will be Closed.

Service Definition

A Service Definitions is created to define the following for a Product.

1. Recurring Billing
2. Provisioning only
3. Recurring Billing and Provisioning

For billing purposes, a Service Definition specifies the following:

• Default pricing for the service. Up to three different prices may be specified.
• Default durations for the service associated with the pricing.
• Type of billing cycle:
  • Bill up to the last price specified and stop.
  • Bill up to the last price specified and continue billing the last price.
  • Bill up to the last price specified and repeat the cycle.
• The billing description
• Any associated usage billing settings.

For provisioning purposes, a Service Definition defines the provisioning Information to be collected.

Service Definitions may include Recurring Service Fee Definitions which specify addition billings associated with a Service Definition.

How To

Service Configuration

Use D01 for Service Identification

D01 is by default the Login Name for a Package or Service and can be used for Provisioning. If D01 is not needed for a Login Name, it can be renamed to identify the service as needed. The value of D01 is always passed to the Service Hook if the Service Hook is utilized.

1. Setup:
   a) Edit a Package or Service Definition.
   b) If the identifier will be for internal only, on the General tab in the Billing Display Modifiers section, change Include Service/Package Name Line to No so that D01 will not appear on a customer's bill.
   c) Change Login Name to an identifying attribute of this service on the Provisioning Information tab if a different attribute is desired.
   d) Save the edits.

2. Implementation:
a) Edit or add a new Package or Service to a customer. 
b) Specify the attribute or Login Name in the Provisioning section on the General tab.

3. Results:
   a) The attribute or Login Name will be listed under Label on the User/Package/Service tab with its value under Identifier.
   b) The attribute and its value will not appear on the bill if Include Service/Package Name Line was set to No. Otherwise Service Name: and the attribute will appear on a separate line under the Billing Description on the bill.

**Configure New Service for Automated Future Billing and Provisioning**

Describes how to setup up a Service such that nightly processing will automatically open the Service at a future date. Provisioning may also occur if configured.

1. View a New Service
2. Choose Prospective for Status.
4. Enter a future date for Start Date.
5. Fill in any other data.
6. Click Save.

**Limit Product Selection By Account Class**

Product Selection for a Customer may be limited by specifying the Account Class on the Account Profile and on the Product Definition. This is selection only. No checking is done of data in the database.

1. Populate the accountclasses List.
   a) Select Billing Administration > Lists.
   b) Select the accountclasses List.
   c) Add entries to the accountclasses List.

2. Populate the Account Class field in the Account Profiles.
   a) Select Billing Administration > Lists.
   b) Select the Account Profile.
   c) Set Account Class.
   d) Click Save.

3. Populate the Account Class field in the Product Definition.
   a) Select Billing Administration and then one of the items under Product Definitions. Usage Tier Plan is not applicable for this
   b) Select the Account Class. More than one may be chosen.
   c) Click Save.

When displaying the list of Products to add to a User, only those where the Account Class on the Product is either not set or if set and matches the Account Class on the Account Profile will be displayed. The same process is used when displaying products available in the Customer Portal. The same holds for One Time Items/Fees.

**Create Automated Referral Credits**

1. Setup:
   a) Edit a Package or Service Definition.
   b) Specify the Referral Amount in the General section on the General tab.
   c) Specify the Referral Wait Days. These are the number of days to wait after a Service is Open before issuing the Referral Amount.
   d) Save the edits

2. Implementation:
a) Edit or add a new **Package** or **Service**.
b) Specify the **Referral Account** in the **General** section on the **General** tab.

3. Results:
   a) Nightly processing will create the **Store Credit** transaction for the **Referral Account** if the **Service** is **Open** and the number of **Referral Wait Days** has passed.
   b) **Referral Amounts** are credited only to **Open** or **Suspended Accounts**.
   c) The description for the transaction is **Referral of UserFirstName UserLastName, ('NameOnService')**
   d) The **Charge Type** of the **Store Credit** for Sales Report purposes will be **Referral Credit**.
   e) The **Service Definition** of the **Store Credit** for Sales Report Purposes will be the **Package** or **Service** Definition of the referred product.
Chapter 11

 Provisioning

Topics:

• Concepts
• How To
Concepts

Overview

BillMax is a flexible and extendable Business Support System (BSS) that out-of-the-box handles all billing, payment processing and customer management. BillMax may also be configured to manage Operations Support Systems (OSS) and data in a tightly coupled way so customer billing state and service delivery remain in sync. This ability is known as “provisioning”.

BillMax can instantiate, remove or alter customer services in the OSS. Its architecture is open-ended in the sense that nothing is assumed about the systems it provisions. Generally, if a provisioning API or non-interactive (program or script) is available for the system, BillMax can provision it.

Provisioning is accomplished by the following:

1. The execution of a program known as a "hook" whenever a specific database table is updated. See Provisioning Executable on page 91.
2. A program scheduled to read the database and update an OSS.

A hook may be executed immediately when a database table is updated or set to be run asynchronously.

Customer data, where provisioning is generally applicable, is stored in three tables: account, user and service. Changes to data in any of these tables can trigger provisioning if desired. However, most provisioning is done using data in the service table.

BillMax can provision many different systems or devices. Examples include: RADIUS, domains and email, scheduling, traffic shapers, monitoring, routers, APs, CPEs.

Provisioning errors may occur because the status or data in BillMax and the external system differ. For example, BillMax may think a Service is closed whereas the provisioned system thinks it is open. In such cases, any attempt to Open the service within BillMax will fail. When this occurs, using the Do Not Provision option on the Service can make the data synchronous.

Related reference

account Table (Account) on page 167
The Account table contains information related to the entity being billed.

user Table (User) on page 173
The User table is used for service locations and alternate contacts. Multiple User records may be linked to one Account.

Related information

Account on page 52
User on page 53
Package Definition on page 82
Service Definition on page 83

Provisioning Implementation

When records in designated database tables are created or updated, BillMax looks for the existence of a related executable called a "hook" in the /usr/local/billmax/local directory named table_hook where table is the name of the database table. Presently the following tables are supported:

- account
- user
- service - used for both Packages and Services
- efpsstate
- efpdata
- payhist
• queue
• ticket
• message
• datacall

If the file is present, BillMax will load the environment with values taken from applicable records from one or more database tables and execute the script or program. No program arguments are passed to this process.

The environment variables passed are dependent on the record being processed. At a minimum the record being processed is passed. Its new variable values are passed in the form NEW_field, where field is the name of the applicable database column. For example, if a service is changed or created the variables passed would be NEW_number, NEW_state, NEW_price, etc.

Tip: To determine what variables are available, create an executable that prints out the environment. See Examine the Account Hook Environment on page 93 as an example.

If the environment variable NEW is set to 0, the environment will also contain old values for the record being modified. These variables take the form of OLD_field. In this way, the launched process can determine what column data has changed and act accordingly. Note that the table_hook executable is not executed for every change during BillMax processing. Many changes may have no OSS related consequences (for example, a subscription renewal made during automated processing).

BillMax provides a service_hook-example file that provides a framework for examining variables to determine if provisioning is required and if so launches a Resource specific process to complete the provisioning.

In addition to passing the variables for the primary record being processed, BillMax also passes in the environment, where applicable, related record data. For the service record, the variables for the parent user and account records are also passed. These records take the form of ACCOUNT_field and USER_field respectively. These values may be useful to the provisioned system for identification or other purposes.

When the provisioning executable process is completed, BillMax examines the result code. If non zero, BillMax reverses any changes made to the record and displays command output of the process to the CSR if the activity is interactive or writes the messages to a log file if not.

Related information

Provisioning Glossary

An understanding of the following entities is essential to use Provisioning with BillMax.

Hook
An executable that is run when a specific table is updated. BillMax sets the environment and will run the executable immediately or create an entry in the asyncprov table for asynchronous execution.

Network Access Server
A RADIUS client device that consults a RADIUS server to determine whether or not a connection to a service should be granted and under what conditions. The case of an internet service, the NAS may consult the RADIUS server to determine if the customer should be allowed to use the network and if so, under what conditions. An example of a condition is to what speed is a customer entitled. BillMax uses RADIUS reply groups defined in the RADIUS server to communicate the conditions to the NAS.

Product/Service Definition
See Package Definition on page 82, Service Definition on page 83

Provisioning Variables
Specified in the Product Definition that is used to provision a Service of that type. While all variables in a
Service can be used in provisioning, these variables are typically limited to the fields: d01, d02, d03 … d10.

RADIUS
Remote Authentication Dial-In User Service (RADIUS) refers to protocols, standards, clients and servers used to provide central Authentication, Authorization and Accounting (AAA) services for computer networks. Authentication verifies a user’s identity whereas Authorization determine what rights, actions or service are allowed for a user. In order to utilize RADIUS, devices in a network must support RADIUS (i.e. include a RADIUS client). This is most often done in customer premise equipment (CPE) or in the case of wireless Internet, at the Access Point (AP). See also https://en.wikipedia.org/wiki/RADIUS for more details.

RADIUS Attribute
RADIUS attributes are informational items used to define specific authentication, authorization, and accounting elements stored on a RADIUS server.

RADIUS Reply Groups
Groups of reply attributes with the primary use to specify settings that tell the NAS what speeds or bandwidth the user is entitled to. Groups may be thought of as Plans. To sell different levels of internet access, multiple groups or plans are created in RADIUS server, then the appropriate product definition in BillMax is configured to use that RADIUS plan.

Resources
Settings in the Product Definition and Service used to identify which systems are provisioned by BillMax. Resources are also used in name spaces to ensure uniqueness of Service names. The Resources are managed a bit-wise integer field where every bit specifies a potential system to be provisioned or name space to be checked. Resources is an example of a field whose value is copied to the Service from the Service Definition when the Service is created. See also Resources on page 91

Service Hook
The general name for the executable used for provisioning Services. Its actual path is /usr/local/billmax/local/service_hook

SERVICES Hash
This is a data structure used in /usr/local/billmax/local/service_hook. It is used to specify systems, data, and actions to provision. Typically, it is the only code that needs to customized in /usr/local/billmax/local/service_hook.

BillMax RADIUS Implementation
BillMax distributes a preconfigured FreeRADIUS® (https://freeradius.org) instance. This FreeRADIUS® instance runs on the BillMax server and accesses tables in the BillMax database.

A key feature of BillMax is the ability to tie a subscriber’s status in the billing system with that of the services provided to the subscriber by the ISP. For example, when a customer first subscribes to a service, BillMax can be configured to perform the Operations Support System (OSS) actions needed to provide that service.

Similarly, when a customer doesn’t pay or closes their account, their service can automatically be disabled or terminated. The ability for a Business Support System (BSS) like BillMax to alter OSS systems and data is known as provisioning.
When implementing provisioning by using RADIUS, BillMax transmits subscriber information to the RADIUS database. The information pushed includes, but not limited to, Username, Password, Plan (group name) and Static IP Address (if applicable).

Provisioning manages the authentication and authorization of users via RADIUS. In addition, RADIUS also provides information on customers' use of your network. This accounting information includes, but not limited to, bytes transferred, time online and packets transferred.

For usage-based billing on either time or bandwidth, access to the Radius accounting records are needed. In BillMax this is accomplished via a Perl script that receives the accounting packets and writes them to a BillMax datacall table. BillMax optionally performs usage billing upon receipt of this data. The BillMax Remote Application process (edge) is used by the Perl script in the submission of data. Therefore, the setup and use of this daemon is a prerequisite for populating the datacall table and optionally performing billing on this data.

BillMax along with FreeRADIUS® includes logic to manage prepaid network time. When access time is purchased for prepaid services, BillMax stores that time on the Service record. That time is subsequently managed via the post authorization step in the BillMax Perl script. Specifically, the used time is determined by querying the datacall table. Then that amount is subtracted from the purchased time to determine the remaining time (Radius Session-Timeout attribute).

Related information
Provisioning Implementation on page 88

Provisioning Executable

BillMax provides executables used by hooks for commonly provisioned systems. However, sometimes it is necessary to create a new executable. The executable should be written with the following guidelines:

- Return 0 for success, non-zero for failure.
- If failure, any changes that were made to the target systems should be rolled back.

If using the delivered /usr/local/billmax/local/service_hook-example, calling the executable using the following style is recommended.

```bash
program -arg1 -arg2 -arg3 value -argN serviceName
```

Resources

Resources are used to delineate different name spaces and are also used in provisioning. Resources are implement as a bit mask in a 32 bit field and as such there may be 32 resources. Values for Resources are 1,2,4,8,...

Each Service Definition may be defined with one or more Resource bit set. When a Service is created, the Resources of the associated Service Definition is copied to the Service.

⚠️ CAUTION: If after a Service is created the Resources on the Service Definition are modified, warnings will be printed in the nightly processing logs. Either the Service Definition Resources will have to be reset or the Service Resources updated. As Resources are tied to provisioning, caution should be take when making Resource changes.

Name spaces

Resources are used in name space delineation. By default, two non-closed Services that have a Resource bit in common many not have the same value for the Service d01 field.

Provisioning

Resources are use in the example Provisioning script /usr/local/billmax/local/service_hook-example. They are used to determine which systems are provisioned. The following is example code fragment that uses Resource 1:

```bash
#         ----- The billmax resource number (1-31)
#         |
#         V
```
```$SERVICES{1}{name} = "radius";
$SERVICES{1}{provision} = "yes";
$SERVICES{1}{data} = [ "-p", "{NEW_d02}" ];  # NOTE HOW THE PASSWORD FIELD IS INDICATED!!!
$SERVICES{1}{cmd}{create} = "/usr/sbin/useradd -M -g radius -s /bin/true -c " . escR("$ENV{USER_fname} $ENV{USER_lname}") . " -p " . escR($ENV{NEW_d02}) . " "$ENV{NEW_d01};
$SERVICES{1}{cmd}{delete} = "/usr/sbin/userdel $ENV{NEW_d01}";
$SERVICES{1}{cmd}{disable} = "/usr/sbin/usermod -L $ENV{NEW_d01}";
$SERVICES{1}{cmd}{enable} = "/usr/sbin/usermod -U $ENV{NEW_d01}";
$SERVICES{1}{cmd}{rename} = "/usr/sbin/usermod -l $ENV{NEW_d01} $ENV{OLD_d01}";
$SERVICES{1}{cmd}{update} = "/usr/sbin/usermod";
$SERVICES{1}{remotecmd} = "/usr/bin/ssh 192.168.1.1";
```

**Bandwidth Shaping**

Webopedia™ describes Bandwidth Shaping as “the process of managing or controlling (shaping) portions of network connection to the outside world and determining an allowed Bandwidth consumption based on types of activities”.

Bandwidth in this context refers to a maximum available network usage as measured in bytes (or some other units) per second.

For the ISP or WISP, Bandwidth Shaping is used primarily as a means to ensure a provided service level or capacity. For example, the ISP or WISP might sell three separate Internet services, each with its own available bandwidth and price.

By deploying a Bandwidth Shaper, the service provider can deliver the promised Bandwidth to their customers.

Secondarily, Bandwidth Shaping is used to throttle customer Bandwidth in a fair manner whenever available up stream Bandwidth or other resources are exceeded.

Lastly, Bandwidth Shapers often include monitoring and offer operational view of networks, bandwidth usage, bottleneck, etc.

BillMax has integrations with several products that do bandwidth shaping and can easily be modified to support other products. These Integrations maintain ISP or WISP subscriber accounts in these products. The data provisioned typically includes *customer ID, IP address, access point, and bandwidth plan*.

**How To**

**Activate Asynchronous Hook Processing Centos 7.x®**

1. cp /usr/local/billmax/pkg/bx_processHooksAsync.service /usr/lib/systemd/system
2. systemctl enable bx_processHooksAsync
3. Optionally cp /usr/local/billmax/local/hooksasync.conf-dist /usr/local/billmax/local/hooksasync.conf and edit the following settings
   
   **HOOKSASYNC_SLEEP**
   The number of seconds to wait before starting another iteration of processing
HOOKSASYNC_RETAIN_COMPLETE  The number of days to keep completed entries in the asyncprov table.

HOOKSASYNC_RETAIN_ERROR  The number of days to keep entries in the asyncprov table where the hook returned a non-zero code.

HOOKSASYNC_RETAIN_PENDING  The number of days to keep entries in the asyncprov table where the entry was created but not released for processing.

4. systemctl start bx_processHooksAsync

Examine the Account Hook Environment

The steps below with minor modification may be used to view other hook environments such as User and Service.

1. cd /usr/local/billmax/local
2. ln -s testhook account_hook
3. Create and edit an Account using the Staff Portal
4. Examine the results in the file /usr/local/billmax/logs/account_hook.log

Implement Service Provisioning

Implementing Provisioning will be demonstrated by provisioning RADIUS using RADIUS Reply Groups for MikroTik® for a 2M Up/4M Down Service when Open and 56K Up and Down when Suspended.

1. Determine data specific to the customer service that need to be captured and maintained.
   For RADIUS, three pieces of data are needed: username, password, and plan identifier
2. Define Resources
   a) Select System Administration > Resources
   b) Select New
   c) Configure the Resource
      For RADIUS enter:

      | Field       | Value       |
      |-------------|-------------|
      | Number      | 1           |
      | Name        | RADIUS      |
      | Description | RADIUS AAA  |
      | Type        | RADIUS      |

      Note: Make a note of the Resource number. This number will be needed for editing the service_hook. The Resource number should not be changed in BillMax unless the service_hook is modified as well.
   a) Click Save
3. Optionally add preconfigured RADIUS attributes for the hardware desired
   For MikroTik® on the command line as the BillMax system user, execute
   load_dictionary.pl -f /usr/local/billmax/freeradius/share/freeradius/dictionary.mikrotik
4. Create one or more RADIUS Reply Groups
   a) Select Billing Administration > RADIUS Reply Groups
   b) Click the RADIUS Reply Group number that needs provisioning or Select New
c) Add a Reply Group

d) Select **Add Attribute** and add a predefined attribute or add a new one. Repeat for as many attributes as necessary

Create a **2MB Up/4MB Down** Reply Group for an Open Service and a **Suspended** Reply Group for Suspended Services.

1. Create the 2MB Up/4MB Down Reply Group and add the following Attribute:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Operator</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mikrotik-Rate-Limit</td>
<td>=</td>
<td>2M/4M</td>
<td>reply</td>
</tr>
</tbody>
</table>

2. Create the **Suspended** Reply Group and add the following Attribute:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Operator</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mikrotik-Rate-Limit</td>
<td>=</td>
<td>56k</td>
<td>reply</td>
</tr>
</tbody>
</table>

If you execute the SQL `SELECT * FROM radgroupreply` using **Reports > One Time SQL** you should see the following

![Table 4 image]

5. Create or Modify a Product Definition to capture the Provisioning Data

   a) Select **Billing Administration > Recurring Service**
   b) Click the product definition number that needs provisioning or Select New
   c) Select the Provisioning tab and enter provisioning data
   
   For RADIUS, create of Service Definition named **2MB Up/4MB Down** with the following:

   - Set **Provisioning Variable Conflict** to **Never OK**
   - Set **Save Plain Text Password** to **Yes**
   - Select the **Resources RADIUS** check box.
   - Enter the following:

   **Table 4:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>RADIUS Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>d01</td>
<td>User Name</td>
<td>username</td>
</tr>
<tr>
<td>d02</td>
<td>Password</td>
<td>password</td>
</tr>
<tr>
<td>d03</td>
<td>Plan</td>
<td>plan identifier</td>
</tr>
</tbody>
</table>

   - In addition for d03, enter **2MB Up/4MB Down** for **Constraint or List ?** and 2MB Up/4MB Down entry as the **Default**.

   d) Select Save

6. If needed, create a program to transmit the provisioning data to the target system or use an existing program. See **Provisioning Executable** on page 91

   For RADIUS, copy `/usr/local/billmax/local/bx_provision_radius.pl-example` to `/usr/local/billmax/local/bx_provision_radius.pl`. Edit the file and set `$db`, `$dbuser` and `$dbpass` to the values needed to access the BillMax database. In addition, set `$DISABLEMETHOD` to `group`.

7. Activate `/usr/local/billmax/local/service_hook`

   a) **Warning:** The `service_hook` runs as the BillMax system user. It can be configured to run under a different id (including root) if needed. The use of non-privileged users is recommended. Please contact support@billmax.com for assistance if a different user id is needed.
Copy `service_hook-example` to `service_hook` in `/usr/local/billmax/local` and ensure that it is owned and executable by the `billmax system user`.

b) Modify the script to call the appropriate system specific provisioning executables

1. Edit the SERVICES data structure or hash in `/usr/local/billmax/local/service_hook`. Every Resource must have its own hash set of parameters. Below show the use of `/usr/local/billmax/local/bx_provision_radius.pl`

```perl
my $cmd = "$BILLMAXDIR/local/bx_provision_radius.pl";
$SERVICES{1}{name} = "radius";
$SERVICES{1}{provision} = "yes";
$SERVICES{1}{groupswitch} = "-g";
$SERVICES{1}{passwordswitch} = "-p"

"{NEW_state}", "-n", 
{"NEW_d01}",
"{NEW_plaintextpass}",
{"NEW_d03}" ];
-a create "."
escL($ENV{NEW_plaintextpass}) . "
escL($ENV{NEW_d03}) . " 
-a delete ".
-a update";

Note: The $SERVICES{1} above refers Resource # 1. Each of the cmd entries specify how to call the system specific provisioning executable ("$BILLMAXDIR/local/bx_provision_radius.pl") for that action defined. There are three actions: create, delete and update. The create and delete actions are explicitly defined. For updates, the information in the data entry is used to dynamically create the update command arguments. The data entry contains pairs of values that define the update switch and associated value that if changed will trigger a update call. In the above, the remotecmd entry is blank which indicates the command runs on the BillMax host. If populated it is, almost always, the ssh command and any needed arguments. In this way the $cmd executable may be executed locally or on a remote host.

8. Test the implementation

a) Go to a test Account within BillMax add a Service based on the Product Definition
For the RADIUS example add the 2MB Up/4MB Down Service Definition and enter:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>user</td>
</tr>
<tr>
<td>Password</td>
<td>p@ssw0Rd</td>
</tr>
</tbody>
</table>

If you execute the SQL `SELECT * FROM radusergroup` using Reports > One Time SQL you should see the following:

```
<table>
<thead>
<tr>
<th>username</th>
<th>groupname</th>
<th>priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>2MB Up/4MB Down</td>
<td>1</td>
</tr>
</tbody>
</table>
```
SELECT * FROM radcheck WHERE username='user' will result in:

<table>
<thead>
<tr>
<th>id</th>
<th>username</th>
<th>attribute</th>
<th>op</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>user</td>
<td>Cleartext-Password</td>
<td>:=</td>
<td>p@ssw0Rd</td>
</tr>
<tr>
<td>5</td>
<td>user</td>
<td>Max-All-Session</td>
<td>:=</td>
<td>0</td>
</tr>
</tbody>
</table>

b) Suspend the Service

SELECT * FROM radusergroup should not show:

<table>
<thead>
<tr>
<th>username</th>
<th>groupname</th>
<th>priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Suspended</td>
<td>1</td>
</tr>
</tbody>
</table>

Important: Whether or not a NAS will honor the changes made to the RADIUS table is dependent on the NAS. The following are possibilities:

- The NAS will accept a RADIUS COA record and contact RADIUS for new information.
- The NAS will accept an SNMP request to contact RADIUS for new information.
- Something else NAS specific.

Related information

Provisioning Glossary on page 89
An understanding of the following entities is essential to use Provisioning with BillMax.

Provisioning Executable on page 91

Manage Asynchronous Hook Processing

1. Select Billing Administration > List from the Top Menu.
2. Select the asynhooks List.
3. Choose the table for which the hook should be run asynchronously. The list shows the tables that support asynchronous processing.
4. Set the value for the selected table.
   - 1 - turn on asynchronous processing.
   - 0 - turn off asynchronous processing.

RADIUS Configuration

Confirm with your network engineers that Radius is deployed in your network. Determine if Radius will be used for only Authentication and Authorization or for Accounting also. Configuration steps are divided into these three sections.

Configure Bundled FreeRADIUS® Server

These steps will configure the FreeRADIUS® instance bundled with BillMax.

1. Configure and start the remote application server (Edge). Setup the Customer Portal on page 188
2. Add the FreeRADIUS® Remote Application.
   a) Select System Administration > Remote Applications
   b) Select New and do the following:
      1. Enter the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>RADIUS</td>
</tr>
<tr>
<td>Description</td>
<td>RADIUS</td>
</tr>
</tbody>
</table>
2. Click Generate Session ID
3. Click Save
4. Select System Administration > Manage Edge Process and start or restart the edge process

3. As the root user in the shell, execute `/usr/local/billmax/bin/setup_freeradius.pl`

4. Add the RADIUS server
   a) Select System Administration > RADIUS Servers
   b) Select New
   c) Enter the following
   
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Primary Server</td>
</tr>
<tr>
<td>Database Name</td>
<td><code>billmax</code></td>
</tr>
</tbody>
</table>

   CAUTION: This must be the same as the BillMax database name. Check `billmax.conf` file.

d) Select Save

5. Select Start Server if the local RADIUS server if not running as shown by Server Status

6. Test basic RADIUS functionality.
   a) Enter in Test Settings
   
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscriber Username</td>
<td><code>ping</code></td>
</tr>
<tr>
<td>Subscriber Password</td>
<td><code>w3e4r5t^y&amp;u8</code></td>
</tr>
<tr>
<td>Shared Secret</td>
<td><code>testing123</code></td>
</tr>
<tr>
<td>Type</td>
<td>Authorization</td>
</tr>
</tbody>
</table>

   b) Select Test on the menu. if successful an Access-Accept message will display

   ```
   Sent Access-Request Id 81 from 0.0.0.0:40341 to 127.0.0.1:1812 length 44
   User-Name = "ping"
   User-Password = "w3e4r5t^y&u8"
   Cleartext-Password = "w3e4r5t^y&u8"
   Received Access-Accept Id 81 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
   ```

c) Change Type to Accounting

d) Add to Test Settings the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionID</td>
<td><code>RADIUS0123</code></td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>33</td>
</tr>
</tbody>
</table>
Bandwidth Shaping with Radius

BillMax can provision any bandwidth shaper product or system that utilizes RADIUS (Remote Authentication Dial-In User Service) protocol and server for obtaining the desired bandwidth policy. Products using this approach include those from Mikrotik® and Cambium®.

The policies are provided as RADIUS reply attributes upon a successful session start up (authentication and authorization).

Related information

*RADIUS Configuration* on page 96
Bandwidth Management with Preseem®

BillMax can be used in conjunction with Preseem®’s bandwidth management to enforce subscriber and access point bandwidth limits and to manage streaming video and other high-bandwidth applications while improving subscriber Quality of Experience.

Presently Preseem uses database polling versus an API to obtain subscriber and network information from BillMax. Preseem requires read access to these tables:

- **Account**
- **User**
- **Service**
- **Servdef**
- **Servicechange**

To provide database access, create a database account and give it read access to the above listed tables. Preseem staff will provide the necessary “from” host IPs for this account. In addition, external firewall changes may be needed to allow MySQL® access (port 3306) from external hosts. BillMax strongly recommends that access to its database be given only to select hosts!

Use the following SQL to create the necessary account and access:

```sql
CREATE USER 'preseem'@'HOSTIP' identified by 'PASSWORD';
grant select on billmax.account to 'preseem'@'HOSTIP';
grant select on billmax.user to 'preseem'@'HOSTIP';
grant select on billmax.service to 'preseem'@'HOSTIP';
grant select on billmax.servdef to 'preseem'@'HOSTIP';
grant select on billmax.servicechange to 'preseem'@'HOSTIP';
```

Preseem will poll these tables periodically looking for change to specific fields. **Account** and **user** are used for identification purposes. **Servdef** and **servicechange** are used for obtaining the plan (bandwidth) information. **Service** is used to identify the IP and possibly the MAC address for the managed Internet access.

In addition to the above, Preseem needs information about Access Points. Access Point information location in BillMax varies by version and installation. During deployment please consult with BillMax support concerning how your data is organized. Preseem will need this information when configuring its polling or external tables.

Bandwidth Management with Saisei™

BillMax is integrated with Saisei™’s Traffic Manager (STM) product. With this integration, the following Saisei objects can be managed in real time from within BillMax:

- Rate Plans
- Users
- Hosts
- Access Points

Rate Plan objects hold the policy parameters (rate limits and other settings). User identify the subscriber. A Saisei User maps to a BillMax User and identifies a subscriber as well as a physical address. Hosts represent the devices that have their bandwidth “shaped” and are identified by IP address. These devices are typically Customer Premise Equipment (CPE). Each Host has an associated Access Point. Access Points are identified so overall bandwidth at the Access Point can be managed as necessary when its resources are exceeded. A general understanding of BillMax provisioning is required to complete the implementation.

Related information

*Provisioning* on page 87

Configure New Product Using Saisei

These steps will configure a new product in BillMax using Saisei.
1. Define a **Resource** for Saisei.
   a) Select **System Administration > Resources**
   b) Name the **Resource** 'Saisei'.
   c) Assign a new **Number**.

2. Define one or more Rate Plans.
   a) Select **System Administration > Attributes**
   b) Click **New** and enter **Name** and **Description**. Example: 25Mb
   c) Enter **Type** = Saisei and Click **Save**.
   d) Click **Add Attribute**.
   e) Enter the **Attribute**, **Operator** to '=' and **Value**.
   f) Repeat steps for more plans or attributes as needed.

Consult Saisei documentation for supported attributes. Possible attributes include *downstream_rate* and *upstream_rate*.

3. Define a Product Definition for a Service or a Package that provides internet access with bandwidth shaping via Saisei™'s STM.
   a) Select **Billing Administration > Recurring Service (or Package)**
   b) Choose **New** and enter the required fields, pricing, etc. Set **Technology** = 'Wireless'.
   c) Click **Provisioning Information** tab and check 'Saisei' **Resource**.
   d) Enter the following

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop Required</td>
<td>Yes</td>
</tr>
<tr>
<td>Access Point Required</td>
<td>Yes</td>
</tr>
<tr>
<td>d01</td>
<td>IP Address</td>
</tr>
<tr>
<td>d02</td>
<td></td>
</tr>
</tbody>
</table>
4. Create credentials for API use in the Saisei application.
   a) Select System > Administrators and click New Administrator.
   b) Enter Name='billmax', Password, and Privilege Level='superuser'. Click Create.

5. Add API credentials to saisei.pl script.
   a) On the BillMax host as the billmax user, edit /usr/local/billmax/local/saisei.pl script.
   b) Locate and set APIPASS and APIHOST.
   c) Make sure the STM firewall allows access from BillMax host on port 5029.
   d) Copy the SERVICE host entry at the top of the file for use in Step 6 and save the file.

6. Modify and enable the BillMax service hook.
   a) Note: If the service_hook script is not found, copy service_hook-example to service_hook.
      On the BillMax host as the billmax user, edit /usr/local/billmax/local/service_hook script.
   b) Paste the contents (copied from the saisei.pl script) after any existing $SERVICES statements.
   c) Edit the content so that '?' in the pasted data is replaced with the Resource number defined for Saisei and save the file.

7. Test the setup (assumes Saisei STM is fully configured).
a) **Note:** Make sure user has name, latitude and longitude values. Otherwise these values will not be passed to Saisei.

Add service or package with the product definition created in Step 3.

b) On the new service/package, enter **Tower (POP Region)**, **Access Point**, **IP Address**, and **Rate Plan**. Click **Save**.

8. Verify results.

   a) If errors show up to the CSR, record error message and provide them to BillMax support.

   b) Check Saisei application to see if it host the specified access point, rate plan, user, and host. If not, send contents of `/usr/local/billmax/logs/service_hook.log` to BillMax support. Send output of `ls -l /usr/local/billmax/local/service_hook` and `ls -l /usr/local/billmax/local/saisei.pl`.
Chapter 12

Remote Applications

Topics:

• Concepts
• How To
Concepts

Remote Applications Overview

Remote Applications provides interfaces to BillMax via Web Services.

A Remote Application consists of two parts:

1. The edge daemon that runs on the BillMax server. Current interfaces supported are:
   • ServerPlus®
   • BillMax Staff Portal functionality (limited set)

2. An application that uses an interface. BillMax include a Customer Portal application written in Laravel®

The Remote Applications interface is sold as an add-on to BillMax. Please contact sales@billmax.com for licensing information.

How To
Staff Portal

Topics:

- Concepts
- How To
Lists are an important key in configuring BillMax. Lists are an easy way to customize BillMax to fit your business needs quickly and easily. Lists come in two types:

- a selection of valid values for a particular item such as the Advertising Code on the Account page.
• a set of needed parameters for an operation such as payment

BillMax has the facilities to modify and add to the drop down lists throughout the system or set parameters according to business needs.

Staff Portal Overview

The Staff Portal is the main interface into BillMax. It is the interface used by a CSR, billing manager, etc. The Staff Portal is accessed using a web browser. The HTML and JavaScript is generated on the BillMax service using template files in /usr/local/billmax/uatemplates. Instructions on processing the templates files are contained in files in /usr/local/billmax/cfg/cgiaction.

Hidden Template Fields

Several templates located in /usr/local/billmax/templates have hidden HTML fields that correspond to database fields. These fields may be exposed by the end customer to store additional data. In general the fields are named "misc##" where ## corresponds to "01", "02", etc.

To determine which templates have hidden fields execute the following command grep misc /usr/local/billmax/uatemplates/*.tmp. Fields of the form "<input type="hidden" name="misc01" value="##Fmisc01#"/>" are linked with either integer or character fields in the database. Fields of the form "##Fmisc11#" are linked with either a time or a date field in the database.

Using the Generic Search Facility in the Staff Portal

When viewing a list of entries in BillMax via the Staff Portal, most of the time the left side of the window will have a menu similar to the following:
New

If New is not disabled, the clicking it will enable the creation of a new entry.

Refine Search

Refine Search enables you to filter, display data, and sort entries. What you can filter, display and sort is based on what type of entries you are viewing. What is available for each entry is predetermined and contains almost all the data associated with an entry plus some important data from other entries. For example, the list of Services also contains data from Accounts. The following is shown when viewing a list of Accounts.

Filtering

The Filter tab allows you to control what entries are displayed. As shown, Account that are Closed or in Collections are not displayed in the list. Accounts that have a Company name that contain nothing and are not Closed or in Collections are shown. The "contain nothing" matches all Accounts, so any Account that is not Closed or in Collections is shown.

SQL is built from these statements. It is very possible to do it incorrectly. For example the following has a trailing "AND"
which results in a "BAD SQL" error.

Although it looks ominous, simply click the OK and fix the Filter statements. Clicking the trashcan icon deletes a statement.

Tip: Make sure the first drop down on the first line each section is blank or, if desired, "(". It may not be "AND", "OR" or ")."

For the technically inclined, the Primary Filters are used in SQL WHERE clauses while the Secondary Filters are used in SQL HAVING clauses.

Sorting

The Sort tab allows you to specify in what order the entries are displayed. Below is an example of sorting for Account entries.

Use the shuffle icon to change the sort order by dragging a Sort Field up or down. The trashcan icon removes a Sort Field.

A Sort Field need not be displayed to be used in sorting.
**Displaying**

The Display tab allows you to choose what data for each entry are displayed. Data that are checked for display are sorted to the top of the list. The order of data displayed in the results is the order of the data display in the Display tab. The display order may be rearranged by dragging rows using the shuffle icon.

**Clear Search**

If **Clear Search** is followed by (F) it means that the list of entries has been filtered in some way. Clicking **Clear Search** reverts the entry display to a default setting removing any custom filter, sort and display settings.

**Option Sets**

**Option Sets** are a way of storing the custom filter, sort and display settings for later use. Currently **Option Sets** are associated with the BillMax user who created the **Option Set** and may not be shared amongst other users of BillMax. See the Flash tutorial [https://www.billmax.com/bx2012/docs/menu/BillMax2014HowToSaveOptions.htm](https://www.billmax.com/bx2012/docs/menu/BillMax2014HowToSaveOptions.htm) for more instruction.

**Re-Opening Accounts, Users, Services**

When opening an Account, User or Service (referred to as Entity) that has been Suspended or Closed, there are several factors that need to be considered:

- Should the associated entities that were Closed or Suspended be re-opened when the primary entity is re-opened?
- Has the Service been not Open so long that there is a gap in time in billing? If so, should billing occur such that billing is time continuous or should billing resume from a date where some or all of the gap in time is maintained?
- When there are billings that cover the time the Service was not Open, should those billing be reversed in the form of Sales Returns and possibly Store Credit Reversals?

Answers to these questions depend on the reason the Entity is being opened and the business practices of the company using BillMax. For example, if the Account was Suspended for being overdue and no gap in time for billing has occurred, time continuous billing with Sales Returns for the time the Service was dormant might be desirable. On the other hand, if the Account is Suspended at the customers request as they will be out of town for the winter (think Snowbirds), then time continuous billing is inappropriate.

When re-opening an Entity BillMax displays a dialog to provide input for these issues. The following is displayed when re-opening an Account. Similar dialogs are displayed when re-opening Users and Services.
**Re-Open Users/Services**

Yes - Re-opens Users and Services that were Suspended or Closed when the Account was Suspended or Closed.

No - Only the Account in re-opened.

**Billing Catch Up Days**

If blank, as is shown in this dialog, then the Next Billing Dates of the Services are in the future. The interpretation is that billing for the Service is current and the Suspension or Closure occurred after the current billing and before the next billing. If a number is displayed, then this displays the largest number of days representing a gap in time in billing for a Service. Whether or not to fill in all or part of the gap is a business decision.

**Date to Resume Billing**

This is effective only if **Billing Catch Up Days** is not blank. Use this date to create a time gap from the past to the present or the future. There is no effect for any Service with Next Bill Date in the future relative to this date. If this is the case, the assumption is that billing for that Service has already occurred.

**Sales Return for non-Open Period**

No Sales Return - no Sales Returns or Credit Reversals will be generated.

Sales Return - Consider the Services that are Opened. Sales Returns and/or Store Credit Reversals will be generated for each Sale and/or Store Credit with dates of service that span all or part of the dormant period of the Services.

**Reopening an Account/User/Package/Service**

Care should be given when re-Opening a BillMax Account, User, Package or Service (AUPS) as there are billing implications.

By default billing will not occur when an AUPS is not open. However, some users of BillMax want billing to continue while an AUPS is Suspended for being Overdue. This is specified on the Account Profile by Bill Suspended Overdue.
If an AUPS is Suspended and the Reason is Overdue, billing will continue if BillMax Suspended Overdue is Yes.

There are several ways to control the billing behavior when an AUPS is re-opened:

- **Default** - when a AUPS is re-opened, billing is caught up from the last time billing occurred if needed.
- **Sales Returns Issued** - when an AUPS is re-opened Sales Returns are given for billings if they exist from when the AUPS was not open to the current date. This is controlled at each level of AUPS by

Billing resumes on the AUPS is opened unless overridden by Date to Reopen/Resume Billing.

- **Skip billing** - skip billing until a specified date. This is controlled at each level by

Note that when an AUPS is opened via the Staff Portal, a dialog is displayed to set the options:

### Examples

**Suspend During Billed Term/Reopen During Billed Term**

- Bill Suspended Overdue is No.
- Customer billed from 1/1 - 1/31.
- Next Billing Date is 2/1.
- Payment due 1/15 or suspension occurs 1/16.
- AUPS Suspended 1/16.
- Customer makes Payment on 1/20.
- AUPS opened on 1/20.

If Sales Return When Opening is set to "Sales Return", Sales Returns from 1/16-1/19 will be issued.
If Sales Return When Opening is set to "No Sales Return", no Sales Returns will be issued. No effect on Next Billing Date.

**Suspend Day after Billed Term/Reopen Next Day**
- Bill Suspended Overdue is No.
- Customer billed from 1/1 - 1/31.
- Next Billing Date is 2/1.
- Payment due 1/31 or suspension occurs 2/1.
- AUPS Suspended 2/1.
- Customer makes Payment on 2/2.
- AUPS opened on 2/2.

If Sales Return When Opening is set to "Sales Return", Sales Returns for 2/1 will be issued.
If Sales Return When Opening is set to "No Sales Return", no Sales Returns will be issued.
Next Billing Date set to 3/1.

**Suspend During Billed Term/Reopen Shortly After Billed Term**
- Bill Suspended Overdue is No.
- Customer billed from 1/1 - 1/31.
- Next Billing Date is 2/1.
- Payment due 1/15 or suspension occurs 1/16.
- AUPS Suspended 1/16.
- Customer makes Payment on 2/15.
- AUPS opened on 2/15.

If Sales Return When Opening is set to "Sales Return" billing from 2/15 to 2/28 will occur and Sales Returns from 1/16-1/31 will be issued.
Next Bill Date will be 3/1.

**Bill Suspended Overdue is Yes/Reopen After Billed Term in which Suspension Occurred**
- Bill Suspended Overdue is Yes.
- Customer billed from 1/1 - 1/31.
- Next Billing Date is 2/1.
- Payment due 1/15 or suspension occurs 1/16.
- AUPS Suspended 1/16.
- Customer billed from 2/1-2/18.
- Next Bill Date is 3/1.
- Customer makes Payment on 2/15.
- AUPS opened on 2/15.

If Sales Return When Opening is set to "Sales Return" Sales Returns from 1/16-2/14 will be issued.

**Suspend During Billed Term/No Sales Return/Reopen Months Later/Use Date to Resume Billing**
- Bill Suspended Overdue is No.
• Company policy is customer resumption for overdue is no Sales Returns.
• Customer billed from 1/1 - 1/31.
• Next Billing Date is 2/1.
• Payment due 1/15 or suspension occurs 1/16.
• AUPS Suspended 1/16.
• Service reinstated 6/1 outside of BillMax.
• Customer makes Payment on 6/15 for 1/1-1/31 and 6/1-1/30.
• AUPS opened on 6/15.

Sales Return When Opening is set to "No Sales Return". Date to Reopen/Resume Billing set to 6/1.

Customer is billed 6/1-6/31. Next Bill Date 7/1.

How To

Modify Existing Staff Portal Pages

Expose Hidden Template Fields (Text/Integer)
Modify Staff Portal templates to expose hidden fields that are linked to text or integer fields in the database.

Modifying the Account template to expose misc01 will be described.
1. Edit /usr/local/billmax/uatemplates/account.tmp
2. Locate the HTML element <fieldset> where the misc01 field will be displayed on the Account page.
3. In the fieldset, add the appropriate HTML elements <li> and <label>.
4. Move the element '<input type="hidden" name="misc01" value="##Fmisc01#"/>' after the <label>.
5. Change 'type="hidden"' to 'type="text"'.
6. Save the file

```
<input type="hidden" name="misc01" value="##Fmisc01#"/>
```

becomes

```
<li>
  <label for="accountMisc01">Address Line</label>
  <input id="accountMisc01" type="text" name="misc01" value="##Fmisc01#"/>
</li>
```

Turn a Staff Portal Input Field into a Yes/No Selection

1. Edit the appropriate template file in /usr/local/billmax/uatemplates. For example, edit /usr/local/billmax/uatemplates/account.tmp.
2. Make the edits as shown in the example. A value of 1 will be associated with "Yes" and 0 will be associated with "No".

```
<li>
  <label for="accountMisc01">Address Line</label>
  <input id="accountMisc01" type="text" name="misc01" value="##Fmisc01#"/>
</li>
```
Changing which Data are Required

1. Edit the appropriate template file in /usr/local/billmax/uatemplates. For example, edit /usr/local/billmax/uatemplates/account.tmp.

2. Add or remove elements from the required JavaScript array using the ID of the field pre pending with the "#" symbol.

<table>
<thead>
<tr>
<th>Making the Account Second Address Line Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change</td>
</tr>
<tr>
<td>required.push('#accountCfname');</td>
</tr>
<tr>
<td>required.push('#accountClname');</td>
</tr>
<tr>
<td>required.push('#accountAddr1');</td>
</tr>
<tr>
<td>required.push('#accountCity');</td>
</tr>
<tr>
<td>required.push('#accountZip');</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>to</td>
</tr>
<tr>
<td>required.push('#accountCfname');</td>
</tr>
<tr>
<td>required.push('#accountClname');</td>
</tr>
<tr>
<td>required.push('#accountAddr1');</td>
</tr>
<tr>
<td>required.push('#accountCity');</td>
</tr>
<tr>
<td>required.push('#accountZip');</td>
</tr>
<tr>
<td>required.push('#accountAddr2');</td>
</tr>
</tbody>
</table>

Configure the Default Account Company Search to Include Accounts Closed or in Collections

1. Select Billing Administration > List from the Top Menu.
2. Select Next from the Context Menu until the searches List is in view.
3. Choose the searches List.
4. Choose the Account Company List Item.
5. Remove all data from the Description box.
6. Paste the following into the Description box.

   ?
   FORM=genericidx&action=account&searchfield=account_company&filter_types=0,0,0&sortingtxt=account_company:1,account_clname:1,account_cfname:1,account_number:1&searchoperator=1&searchval=

7. Click Save.
8. Verify Result.
   a) Reload the web page.
b) Perform an **Account Company** search.
c) Select **Refine Search** from the Context Menu.
d) Verify the only **Filter Field** is **Company**.

### Add Staff Portal Validation

Describes in general terms how to add Client (Browser) and Server Side validation for data coming from the Staff Portal. See also *Modify Existing Staff Portal Pages* on page 114

1. Discover what template file in `/usr/local/billmax/uatemplates` is being used to display the page to which validation will be added.
   a) While viewing the page view the source of the page.
   b) Look for the string `name="FORM"` as part of an HTML hidden field. In general the value of the hidden field will reference the template file name. There may be multiples entries so several template file many need to be examined to determine the correct one

2. Client (Browser) Validation may be added to the template file. In general this is done by adding JavaScript to the `localVerifyData` JavaScript function in the template. If one does not exist in the template, it can be added.

3. Server Side Validation may be accomplished by adding to the constraint entry for a data element in the corresponding file in `/usr/local/billmax/cfg/cgiaction`. The constraint entry may be either a Regular Expression or a macro from the List `regexmacros`.

---

**Add Server Side validation to the field Phone Number on the Account page**

1. Edit `/usr/local/billmax/cfg/cgiaction/account.cat`.
2. In the section `account,cphone`, change `constraints:^[` to `constraints:^[0-9]{3}[-][0-9]{3}[-][0-9]{4}$`
3. Execute `mkca -i account.cat`.

   This will enforce a 10 digit number of the form ###-###-####.
Chapter 14

System Administration

Topics:

• Concepts
• How To
Concepts

Billing Log Files

Log files are stored in `/usr/local/billmax/logs` and `/usr/local/billmax/tmp`. The compression and retention of log files is specified in `/usr/local/billmax/local/manageFiles` using XML.

These are example entries from `/usr/local/billmax/local/manageFiles`.

```
<BxDirInfo>
  <dirName>${BILLMAXDIR}/tmp</dirName>
  <BxFileInfo>
    <fileRegex>^[a-zA-Z0-9].*</fileRegex>
    <ndays>3</ndays>
    <includedirectories>1</includedirectories>
    <compress>0</compress>
  </BxFileInfo>
</BxDirInfo>

<BxDirInfo>
  <dirName>${BILLMAXDIR}/logs</dirName>
  <BxFileInfo>
    <fileRegex>nightly_.*closeofday\.log</fileRegex>
    <ndays>3</ndays>
    <includedirectories>0</includedirectories>
    <compress>1</compress>
  </BxFileInfo>
</BxDirInfo>
```

$BILLMAXDIR$ Typically replaced with `/usr/local/billmax`.

dirName The directory to consider.

fileRegex A Regular Expression that specifies the files to consider. In first example, any file name beginning with and consisting of letters (upper and lowercase) and numbers. In the second example, any file consisting of `nightly_` followed by any characters then followed by `closeofday.log`.

ndays The number of days the file should be retained.

includedirectories Specifies whether or not to include subdirectories in the processing.

0 Do not include subdirectories.

1 Include subdirectories.

compress Specifies whether or not to compress the log files.

0 Do not compress the log files.

1 Compress the log files.

Note: Deletion based on `ndays` is done by comparing the system date with the with the file date. Files will not be deleted during BillMax testing by means of running the BillMax date forward as the system date itself is not changed.
Backups, Redundancy and Archiving

Backups, Redundancy and Archiving are strategies that may be employed to ensure data integrity and preserve the viability of a company's operations. For some companies, Backups will be sufficient. For others, all three may be needed.

BillMax offers a backup and three month archival service for BillMax related databases and files. If interested please contact sales@billmax.com. Whether or not BillMax services are utilized, having good backups cannot be emphasized enough.

Backups

Backing up "BillMax" consists of making a copy, if needed in compressed form, of the following:

- On the BillMax Server:
  - The /opt/billmax directory.
  - The /usr/local/billmax directory.
  - The billmax database.
  - The billmax_reports database.
  - Any system files used for emailing, printing, networking, etc.

- On the BillMax Customer Portal Server, the files used for the customer portal. The location of these files differs depending on the setup of the web server hosting the Customer Portal.

A program such as tar may be used to create a backup of the directories. There are many utilities such as mysqldump and mysqlhotcopy to create database backups. Some customers prefer commercial solutions. Other customers prefer to shutdown the database completely and use tar to backup the database files.

Regardless of what is used to backup BillMax, storage and maintenance of the Backups is critical.

⚠️ DANGER: STORE YOUR BACKUPS OFFSITE!!! This cannot be emphasized enough. True story: A customer had the BillMax servers and all company paperwork in one location. The location burned to the ground. The business never recovered.

⚠️ DANGER: IF POSSIBLE, STORE YOUR BACKUPS IN A DIFFERENT GEOGRAPHIC REGION!!!. True story: A company had facilities in both Dallas and Fort Worth, Texas. The Fort Worth site was hit by a tornado, the Dallas site was flooded, all from the same storm front.

⚠️ DANGER: DO NOT RELY ON RAID!!! There have been several instances where multiple disks and even multiple power supplies in a RAID array have failed simultaneously. Whereas RAID has its place, it is not a substitute for Backups.

It is strongly recommended that recovering from a backup be tested periodically to make sure backups are being done correctly.

Redundancy

Redundancy may be used when waiting for a backup to be restored is not desirable. Redundancy consists of having a server that has a duplicate copy of the software and a duplicate version of the database. Database replication may be of use in this scenario.

⚠️ DANGER: DATABASE REPLICATION IS NOT A SUBSTITUTE FOR BACKUPS!!! True story: A BillMax customer relied on replication rather than backups. When an administrator issued an improper DELETE command on the production database, the replication database was also affected. Without backups the data was lost.

Archiving

Archiving backups is useful if there is a need to revisit database status and activity from the past. True story: A company had a disgruntled ex-employee that managed to still have access to BillMax. Rather than cause a catastrophic failure this employee would periodically delete accounts from BillMax. When discovered, the company
was able to analyze 6 months of data and logs to rebuild the deleted accounts and collect evidence of the malfeasance.
In this case Backups and Redundancy were insufficient.

**Customizing BillMax**

BillMax is designed to be customized, if desired, by the end consumer. The interface is a combination of HTML, J
Query and JavaScript. The server side binaries are written in ANSI C and the source code is available. The library
functions not available for customization.

BillMax uses Subversion to manage combine local changes to a file and BillMax changes to a file upon an upgrade. It is
required that after making a modification to a file that the file be committed to the Subversion repository using the
`svn commit filename`. If a C file is modified, this also includes the resulting binary file.

⚠️ **Warning:** Uncommitted files will cause an upgrade to abort. If this occurs commit the uncommitted files and
restart the upgrade.

ᴴᴰ **DANGER:** Unless there are special circumstances, it a mandatory that all edits, creation of new files, etc. be
done by the BillMax system user, typically `billmax`. If done by a different system user, BillMax may become inoperable.

If customizing BillMax is desired, it is recommended that you take advantage of the 250 Account development license that is included with leasing BillMax. This enables you to run BillMax on another server and test changes before affecting the production server.

Related Links:

- [Modify Existing Staff Portal Pages](#) on page 114
- [How To](#) on page 59 (Customer Communications)
- [Modify a Server-side Binary](#) on page 125
- [Automated Processing Overview](#) on page 26

**Security**

**Data Security**

**BillMax Installation**

Data security is greatly influenced by the end customer's business practices. An installation of BillMax assumes
adherence to standard industry security practices. They include but are not limited to the following:

- The BillMax server(s) hosting both the Staff Portal and the BillMax database are behind a suitably configured
  firewall.
- The Staff Portal is accessible only to the users that need access.
- All network traffic from the web browser of a Staff Portal user to the BillMax server is transmitted using HTTPS.
- All network traffic from the web browser of a Customer Portal user to the Customer Portal server is transmitted
  using HTTPS.
- All network traffic from the Customer Portal Server to the BillMax edge service is transmitted using HTTPS.
- The Customer Portal server is behind a firewall such that the only public access is to the web server through
  HTTPS.
- Physical access to the servers is limited to those who need physical access to the server.
- Login accounts on the servers are limited to those who need access to the server.

**Encrypted Data**

Encrypted data is divided into three categories:

1. Passwords for access to BillMax portals.
2. Passwords for provisioning purposes.
3. Credit Card numbers and Bank Account numbers.
The encryption scheme used to encrypt passwords used to access the Staff Portal or the Customer Portal is configurable through settings in `/usr/local/billmax/local/billmax.conf`. The default encryption algorithm is MD5.

When a end customer uses the Customer Portal to register, the user name and password are encrypted using BLOWFISH and temporarily written to the disk on the Customer Portal server.

The encryption scheme used to encrypt passwords used for provisioning purposes is also configurable through settings in `/usr/local/billmax/local/billmax.conf`. The default encryption algorithm is DES.

**CAUTION:** BillMax provides means for storing provisioning passwords in clear text. It is strongly recommended this not be done if there is no technical reason to do this.

### Credit Card numbers and Bank Account numbers

- **Note:** For the purposes of this discussion, Number will refer to both Credit Card numbers and Bank Account numbers.
- **Note:** For the purposes of this discussion, "third party processors" are those supported by BillMax out of the box and do not include any custom third party processors.

Business practices surrounding Numbers are extremely important due to the sensitive nature of the data. Some, but not all, aspects to consider are:

- If a paper application has the Number, is the application destroyed or the Number redacted?
- Are CSRs trained to not write down Numbers on pieces of paper?
- If calls are recorded, what is done to secure the recordings or to disable the recording if a Number is being provided by the end customer?

Storage of Numbers in BillMax depends on the third party processor that will use the Numbers. If supported by both the processor and BillMax, a Token may be stored in place of the Number. Currently this option is available if IPPay® is the processor. Both Numbers and Tokens are stored using AES encryption.

If PCI compliance is enabled in BillMax, the BillMax customer is prompted to change the AES encryption key every 90 days. The AES encryption key may be composed of phrases entered through the Staff Portal by two different users with Administrative privileges for additional security. See [*Change the AES Encryption Key*](#) on page 125.

Numbers entered through the Staff Portal and Customer Portal are never written to disk in plain text when using a third party processor such as IPPay. The number travels from the browser to the server and is encrypted before being written to the database. If the Number is tokenized, the encrypted Token is written to the database. When used, the Number or Token is decrypted in memory and sent directly to the third party processor using HTTPS.

**DANGER:** If using the NACHA file format, Bank Account numbers are stored in the NACHA file in plain text. This is unavoidable.

For identification purposes, the last four digits of a Number are stored in plain text and may be displayed in either the Staff Portal or the Customer Portal. These last four digits may also be displayed on a Billing Statement or Statement to help the end customer identify the means of payment.

### User Names and Passwords

There are user names and passwords stored in several places in BillMax:

1. The password field in the user table. Used for customer logins to the Customer Portal. Allowed values governed by the `customerpassword` regular expression.
2. The `d02` field in the service table. Used for Service provisioning. Allowed values governed by the `customerpassword` regular expression.
3. The password field in the auth table. Used for logins to the Staff Portal. Allowed values governed by the `staffpassword` regular expression.
4. The password field in the edge table. Used by the Customer Portal *application* login. Must be at least 4 or more characters.
PCI Compliance

The following describes how BillMax is part of a customers overall PCI compliance strategy. More information on the PCI Data Security Standards may be found at [https://www.pcisecuritystandards.org/security_standards/index.php](https://www.pcisecuritystandards.org/security_standards/index.php).

The following elements are set to PCI DSS values by default:

- Access to BillMax is logged and kept for 90 days.
- AES encryption is used to encrypt credit card and bank account numbers or their tokens. A warning is issued when the AES encryption key is more than 90 days old.
- Strong passwords are required for access to the BillMax Staff Portal.
- The maximum number of failed login attempts to the Staff Portal is 3.
- Passwords may not be reused until 4 other passwords have been used.
- A password is required to login to the Staff Portal after 15 minutes (900 seconds) of inactivity.

⚠️ **CAUTION:** How to change this value is one of the most requested items. See *Modify the Staff Portal Timeout* on page 127. This is not recommended.

- Passwords must be reset every 90 days.

Other security measure taken are:

- Credit card and bank account numbers may be stored as tokens. This is dependent on the third party processor.
- Strong Password may be required for access to the Customer Portal.

Authorized Users

Authorized Users are entries in the auth table. These entries may represent people who are allowed to access BillMax Staff Interface, remote application IDs that are used to automatically add or retrieve data from BillMax or people such as Sales Agents who don't necessarily need to have access to the BillMax Staff Portal but need to be tracked in some way.

⚠️ **Important:** An Authorized User has access to all Virtual Companies. BillMax does not support limiting access for an Authorized User to a specific set of Virtual Companies.

Authorization Levels

Access and functionality available to an Authorized User are controlled through Authorization Levels. The Authorization Levels for a particular Authorized User is stored as a bit mask. The following are the predefined levels:

- 1 - Read
- 2 - Write
- 4 - Billing Read
- 8 - Billing Write
- 16 - Administration

If someone has complete access to the system, all values are added together and the result (31) is stored in BillMax for that Authorized User.

Authorization Roles stored in the BillMax List `bxroles` are a short cut for specifying a set of Authorization Levels that may be selected when a new Authorized User is added to the system The Value for the List Entry should be the sum of the desired Levels.

How To

**Convert from CentOS 6.x® to CentOS 7.x®**

Converting from CentOS 6.x® to CentOS 7.x® is a manual process. It entails building a CentOS 7.x® machine and moving data from the old machine to the new machine.
1. Install BillMax on a CentOS 7.x® 64 bit machine. See Install CentOS 7.x for BillMax on page 16. It needs to be the same version of BillMax as on the CentOS 6.x® machine. Use the following information from the old machine on the new machine:

   1. **system username** - typically billmax.
   2. **software repository username/password** - typically billmax/billmax.
   3. The same root password for the MariaDB® database as for the MySQL® database.

2. When running the BxConfigure.cgi process, use the following information from the old machine on the new machine unless otherwise noted:

   1. **username/password** for the BillMax database.
   2. The Admin **password** need only pass the requirements. It will be overwritten.
   3. The AES key will be overwritten.

3. After requesting and entering a new license, make sure you keep a copy. The license will be overwritten and need to be re-entered later.

4. **Important:** From '/usr/local/billmax through uploadedfiles)' there are no spaces or carriage returns.

   As root and in the /root directory, run the following command: `diff -r /usr/local/billmax /opt/billmax/software | grep -v -E ".svn$|\.o$|\.d$" | grep -v -E /usr/local/billmax/(documents|efp|logs|tmp|uploadedfiles|cdrfiles|backups|prepaidcards|snapshots|importtools|uploadedfiles)' > upgradedifferences.dat

5. Examine the results in upgradedifferences.dat. Any pertinent changes will have to be done on the new server.

6. Shut down httpd and mysqld on both servers.
   a) /sbin/service mysqld stop on the old server.
   b) /sbin/service httpd stop on the old server.
   c) systemctl stop mariadb on the new server.
   d) systemctl stop httpd on the new server.

7. Copy the files from the billmax and billmax_reports database on the old servers to the billmax and billmax_reports databases on the new server. Make sure you maintain ownership and permissions.

8. Execute systemctl start mariadb on the new server.


10. Redo any local customizations that were noted in upgradedifferences.dat.

11. Copy contents of the following directories located in /usr/local/billmax on the old server to the comparable locations on the server. Do not copy the ./svn subdirectories. Make sure you maintain ownership and permissions.

   - cdrfiles
   - documents
   - efp
   - logs
   - prepaidcards
   - uploadedfiles

12. A backup at this point is recommended.

13. Execute systemctl start httpd on the new server.

14. If the new server has a new name or IP address and these are significant to the customer portal configuration, remember to update the customer portal files and clear the PHP cache.

**Licensing**

**Determine Information for License Key (Staff Portal)**

Select System Administration > License
The following display will show the needed information. The following is an example. The pertinent information is highlighted.

![Machine Information](image)

**Determine Information for License Key (Command Line)**

1. Log into the BillMax Server.
2. If not already, become the BillMax system user.
3. Execute `/usr/local/billmax/bin/machid`

   This is a typical result. The pertinent information is highlighted.

   ```
   BillMax Host Information. Fri Mar 11 09:15:00 CST 2016
   Operating System: Linux 2.6.32-220.el6.x86_64
   Machid: eth0:00:0C:29:E0:51:BD
   Ipaddr: 192.168.196.150
   inet6: error fetching interface information: Device not found
   int6: error fetching interface information: Device not found
   ```

**Install the BillMax License Using the Staff Portal**

1. Copy the entire license key from the email message. The license key begins with the first non-blank link after the line `-- License Key --`.
2. Login to the BillMax Staff Portal using an ID with Administrator privileges.
3. Select **System Administration > License**
4. Click Clear.
5. Paste the license into the License Key field.
6. Click Save in the Context Menu.

**Install License Key (Command Line)**

1. Log into the BillMax Server.
2. If not already, become the BillMax system user.
3. Execute `/usr/local/billmax/bin/license_install` and follow the instructions.

**Modify a Server-side Binary**

Describes steps necessary to add custom processing to a server-side BillMax binary.

Do this when specialized business rules or specialized processing need to be implemented on the BillMax server.

1. Login or `su` to the BillMax system user, typically `billmax`.
2. Edit the appropriate C file located in either `/usr/local/billmax/src/bin/binaryname` or `/usr/local/billmax/src/cgi-bin/binaryname`.
3. Execute `gmake`.
4. Test if necessary
5. Execute `gmake install` to put into production

**System Settings**

**Change the AES Encryption Key**

Describes how to change the encryption key used to encrypt credit card and electronic check bank account information stored in BillMax.

You must know one of the two possible AES Passphrases associated with the current encryption.
Perform these steps periodically. If not done within a 90 day period, error message 11500 will appear in the nightly processing output. See *Database Checking (dbck) Error codes* on page 166

1. Login in the BillMax Staff Portal using an ID that has Administrative rights.
2. Select from the Top Menu **System Administration > Change Encryption**.
3. Enter the **Old** and **New AES Passphrases**:
4. Select **Process Numbers** from the Context Menu.

**Setup USPS Address Verification**
Describes the steps necessary to activate USPS Address Verification for accurate and complete addresses.

2. Set the BX_USPSUSERID variable to your USPS user id in the `/usr/local/billmax/local/billmax.conf` file.
3. Select from the Top Menu **System Administration > Global Options**.
4. Enter the **usps** in **Address Verification Function**.
5. Select from the Top Menu **System Administration > Global Options**.
6. Click **Stop Edge**. Click **Start Edge**.

**Install the BillMax License Using the Staff Portal**

1. Copy the entire license key from the email message. The license key begins with the first non-blank link after the line "-- License Key --".
2. Login to the BillMax Staff Portal using an ID with Administrator privileges.
3. Select **System Administration > License**

4. Click **Clear**.
5. Paste the license into the **License Key** field.
6. Click **Save** in the Context Menu.
Modify the Customer Portal Application Login Credentials

1. Modify the credentials on the BillMax server.
   a) Select System Administration > Remote Applications from the Top Menu.
   b) Choose the entry for the Customer Portal.
   c) Enter the Application Login Name if needed.
   d) Enter the Application Password.
   e) Select Save from the Context Menu.
2. Restart the edge server.
   a) Login or su to the BillMax server as root.
   b) Execute /sbin/service bx_edge restart.
3. Edit /usr/local/billmax/html/portal/billmaxedge/lib/php/local/billmaxedge.ini and set the user and password entries with the values from steps 1.c and 1.d.
4. If the Customer Portal software is installed on a different server than BillMax, the billmaxedge.ini file on the remote server will have to have the same edits.
5. If testing, you may need to restart your web browser or clear the cache.

Modify the Staff Portal Timeout

This describes how to modify the Staff Portal Timeout. This is not recommended if PCI compliance is a concern.

See PCI Compliance on page 122

2. Choose No for PCI Mode.
3. CAUTION:
   - 0 - disables timeout completely.
   - must be 900 seconds or less if PCI Mode is Yes

   Enter the number of seconds in Session Timeout.
4. Select Save from the Context Menu.

Upgrade BillMax

1. Login as root.
2. Stop any BillMax processes if they are running
   a) fetchmail - Service script is bx_fetchmail.
   b) edge - Service script is bx_edge.
   c) processHooksAsync - Service script is bx_processHooksAsync.
3. Make backups. See Backups, Redundancy and Archiving on page 119
4. Do the following:
   a) cd/usr/local/billmax
   b) svn commit -m "commit information"
   c) rpm -Uvh nameOfRPM
   d) cd/opt/billmax/install
   e) Execute ./BxInstall.pl
   f) Execute yum -y update.
   g) Supply "n" if you see "Continue printing (Y/n)?"
   h) Supply "F" if you see "Enter two indexes for each column to rename, (R)elist, or (F)inish:

5. Resolve any inconsistencies. There should not be any if no customizations have been made. After resolving an inconsistency, you may need to issue the command svn resolved filename where filename is the name of the file that had the inconsistency.
6. `cd /opt/billmax/install. Execute ./BxUroot-4.0.pl`
7. `su` to the `billmax` system user - typically "`billmax`"
8. `cd /opt/billmax/install. Execute ./BxUpost-4.0.pl`
9. Test the interface.
10. Start any BillMax processes that were stopped in step 2.
Chapter 15

Taxes

Topics:
- Concepts
- How To
Concepts

Tax Classes

Tax Classes are used to group products that are subject to the same taxes for a geographic area. They are used along with Tax Regions to determine which Tax Group to use when determining the tax rates to assess. Examples are:

- Sales
- VOIP
- Phone

Tax Groups

A Tax Group combines a Tax Class, a Tax Region and one or more Tax Items. For any Sale or Store Credit, the Tax Class and Tax Region are used to select the Tax Group, and the Tax Items belonging to the Tax Group are used to assess the Taxes.

The Tax Group is where one specifies:

- Taxes are assessed on individual Sales and Store Credits
- Taxes are assessed on the sum of Sales and Store Credits
- Taxes are assessed at different "levels" supporting a "Tax on Tax" scenario.

<table>
<thead>
<tr>
<th>Setting up taxes for Sales Tax, VOIP Tax and Internet Tax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assume the following Tax Items:</td>
</tr>
<tr>
<td>• State Sales Tax</td>
</tr>
<tr>
<td>• County Sales Tax</td>
</tr>
<tr>
<td>• City Sales Tax</td>
</tr>
<tr>
<td>• State VOIP Tax</td>
</tr>
<tr>
<td>• County Internet Tax</td>
</tr>
<tr>
<td>• USF Tax</td>
</tr>
</tbody>
</table>

Assume the following: for a generic Tax Regions - Region A - as an example:

- A sales of goods subject to "Sales Tax" will be assessed the State, County and City Sales Taxes.
- A sale of goods subject to "VOIP Tax": will be assessed the State VOIP Tax and the USF Tax.
- A Sale of goods subject to "Internet Tax" will be assessed the State Sales Tax, the County Internet Tax and the USF Tax.

Three Tax Groups will be created for Region A, one for Tax Class "Sales", one for Tax Class "VOIP", one for Tax Class "Internet":

- Tax Group with Tax Region "Region A" and Tax Class "Sales" will have Tax Items
  - State Sales Tax
  - County Sales Tax
  - City Sales Tax
- Tax Group with Tax Region "Region A" and Tax Class "VOIP" will have Tax Items
  - State VOIP Tax
  - USF Tax
- Tax Group with Tax Region "Region A" and Tax Class "Internet" will have Tax Items
  - State Sales Tax
  - County Internet Tax
  - USF Tax
These taxes and combinations of taxes are defined by the local taxing authorities. Every Sale and Store Credit created in BillMax uses the location of the Sale (Tax Region) and the type of goods sold (Tax Class) to determine which Tax Group to select to assess the various Tax Items.

**Tax Items/Fees**

Tax Items may:

- Specify the individual tax rates for each taxing authority that assesses taxes on a Sale or Store Credit.
- Specify the combined tax rates of all taxing authorities that assess taxes on a Sale or Store Credit.
- A combination of the above.

### Individual Taxes for General Sales
- State
- County
- City

### Combined Taxes for General Sales
- Sales Tax - State + County + City

### Mixture of styles for VOIP
- Sales Tax - State + County + City
- Universal Service Fund (USF)

Tax Items may be classified as a Tax or a Fee. Taxes and Fees are displayed separately on the standard Invoice or Billing Statement.

Tax Items may also be classified as a Universal Service Fund (USF) contribution factor, Federal or State. By definition, these will be displayed as Fees. If a State USF rate, then the State must be specified. The State of the Service Location/User is matched against the Tax Item State to select the rate. Tax Items classified as USF contribution factor will not be eligible for use in a Tax Group. These are used independently as the rate specified may be modified by Safe Harbor or Traffic Study percentages to create an effective rate.

**Tax Regions**

Tax Regions represent a geographic area that share a common set of taxes for goods and services. Taxes for a Tax Region may be regulated by one or more taxing authorities.

The Tax Region **Not Applicable** is a setting that disables all taxation for goods and services for a particular Account. Use this setting only if goods and services are not taxed for any reason across all customers in the geographic region. An example of this is the state of Oregon which does not have a sales tax.

The difference between setting the Tax Region to **Not Applicable** and choosing **No** for **Taxable** on an Account has to do with setting up tax rates and sales tax reporting:

- Choosing **Not Applicable** for Account **Tax Region**
  - No taxes have to be configured
  - There is no reporting of Sales Tax
- Choosing **No** for Account **Taxable**
  - Taxes have to be configured
  - Customer should supply documentation for tax exempt status
  - Sales tax for taxable and non-taxable sales are reported.
Tax Rules

Tax Rules are instructions to BillMax possibly to increase or decrease the amount of a Sale subject to tax. BillMax supports the following Tax Rules:

- A monthly exemption. This is applicable in Texas where the first $25.00 per month of all Internet Access services sold to a customer is non-taxable. All Sales in excess of $25.00 are taxed. BillMax tracks this by month and makes the appropriate adjustments when Sales, Sales Returns, Store Credits and Store Credit Reversals are invoiced for an Account.
- A percentage exemption. This is used when sales are subject to a particular tax, but because of the nature of the sale, the whole amount is not subject to tax. For example, 90% of a sale might be subject to tax. 10% is exempt.
- A monthly maximum. This is the converse of the monthly exemption. Only a certain amount for a particular good or service is subject to tax. Any amount over is not taxable.

Tax Specification on the Account

A customer may be deemed to be subject to taxes or not. If a customer claims to be exempt from taxation, then they should provide documentation supporting the claim.

In general, choose Yes for Taxable. Choose No for Taxable if the supporting documentation is supplied.

Choosing No means that no sale to the Account will be taxed.

An Account may be subject to taxes in general, but be exempt from a specific tax. This is managed using the Account Tax Exemptions tab.

Tax Specification on the User

The only tax setting on the User is the Tax Region. This enables one to sell Services located at different locations that have different taxes. A User will always inherit the Account Taxable setting.

How To

Add a Tax Class

Tax Classes are stored as a BillMax List.

1. Select Billing Administration > Tax Classes from the Main Menu.
2. Click Add.
3. Enter the Item.
4. Click Save.

Add A Tax Group

2. Select New.
3. Enter the Name. Example: Tarrant County Sales Tax
4. Review and enter any other field for the new tax group such as Tax Class or Tax Region. The tool tips give detailed information about each field.
5. Select Save.
6. Select Add Tax Item.
7. Choose appropriate Tax Item.
8. Click Save.
Add A Tax Item or Fee

1. Select Billing Administration > Tax/Fee Items.
2. Select New.
3. Enter the Name that will be used in the drop-down list of Taxes.
4. Enter the Rate in the decimal form x .xx. Example: 8.25.
5. Review and enter any other field for the new tax item. The tool tips give detailed information about each field.
6. Select Save.

Add A Tax Region

1. Login to the BillMax Staff Portal using an ID with Administrator privileges.
2. Select Billing Administration > Tax Regions.
3. Click the New button on the Context Menu.
4. Enter Name for the Tax Region location.
5. Click Save.

Add a Tax Rule

1. Select Billing Administration > Tax Rules from the Main Menu.
2. Select New from the Context Menu.
3. Enter the Name. This will be displayed on the Service Definition pages when apply a Tax Rule to a product.
4. Select the Rule Type.
5. Specify the Amount. This is a dual purpose field. If a monetary amount is expected, enter an amount that conforms to the monetary format. If a percentage is expected, enter a decimal value, i.e. .1 for 10%.

Check Tax Setup

Validation of the Tax Group settings validates the structure of the settings with respect to Tax Regions and Products sold. It does not validate the data entered, i.e. it will not validate tax rates.

Select Billing Administration > Check Tax Setup from the Main Menu.

Note: The most common error is when there is not a Tax Group for a Tax Region/Tax Class combination. This is detected by verifying for each User the Tax Region setting and the Tax Class settings of the recurring Services being sold to the User.

Configure Taxes Using the BillMax Tax Engine

Taxes are charged based on a Tax Group derived from the User’s Tax Region and the Tax Class of the item or service. Tax Regions are set on an Account and inherited on the User. Tax Regions can be overridden on the User.

1. Repeat Add A Tax Region on page 133 for each geographic area that share a common set of taxes for goods and services.
2. Repeat Add A Tax Item or Fee on page 133 for each tax or fee that needs to be assessed.
3. Repeat *Add A Tax Group* on page 132 for each combination of *Tax Class*, *Tax Region*, and *Tax Items* needed. For an example, see *Tax Groups* on page 130.

4. Run *Billing Administration > Check Tax Setup* until there are no errors.

5. To view all *Tax Groups* defined, run *Billing Administration > Tax Setup*.

6. If there are tax exemptions needed, a *Tax Rule* can be created to be used on individual *Service* or *Package Definitions*.

**Tax Setup Report**

Method to list Tax Group settings sorted by Tax Class or Tax Region

1. Select *Billing Administration > Tax Setup* from the Main Menu.

2. Choose *Display By*.

Chapter 16

Reference

Topics:

- API
- Database Checking (dbck) Error codes
- Database Tables
- nightly Typical Batch Processes
API

Command Line

**enter_account.cgi**
Used to create or modify an Account record. If creating an Account, a User record is also created.

**Usage**
/usr/local/billmax/cgi-bin/enter_account.cgi

**Environment**
All data is passed via environment variables of the form FORM_variablename with the following exceptions:

- Dates are passed as FORM_variablename_year, FORM_variablename_month and FORM_variablename_day. A four digit year is required. Two digits for month and day respectively are required.
- Times are passed as FORM_variablename_hour, FORM_variablename_minute and FORM_variablename_second. Two digits are required for each.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>srchnumber</td>
<td>Integer</td>
<td>If non zero, the number of an existing Account.</td>
<td>Yes</td>
</tr>
<tr>
<td>number</td>
<td>Integer</td>
<td>The number to use for the Account. Recommended to use 0 for a new Account.</td>
<td>Yes</td>
</tr>
<tr>
<td>state</td>
<td>Integer</td>
<td>Status of the Account.</td>
<td>Yes if a new Account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Closed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Suspended</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Collections</td>
</tr>
<tr>
<td>startdate</td>
<td>Date</td>
<td></td>
<td>Yes if a new Account</td>
</tr>
<tr>
<td>enddate</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>adcode</td>
<td>Integer</td>
<td>Advertising Code</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 or a value from the adcodes List.</td>
<td></td>
</tr>
<tr>
<td>cancelreason</td>
<td>Integer</td>
<td>Cancel Reason</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 or a value from the cancelreasons List.</td>
<td></td>
</tr>
<tr>
<td>referredby</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cfname</td>
<td>String</td>
<td>First Name</td>
<td></td>
</tr>
<tr>
<td>cmname</td>
<td>String</td>
<td>Middle Name</td>
<td></td>
</tr>
<tr>
<td>clname</td>
<td>String</td>
<td>Last Name</td>
<td></td>
</tr>
<tr>
<td>company</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>--------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>cphone</td>
<td>String</td>
<td>Phone Number</td>
<td></td>
</tr>
<tr>
<td>fphone</td>
<td>String</td>
<td>Fax Number</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>textmessage</td>
<td>String</td>
<td>Email address equivalent for a cell phone.</td>
<td></td>
</tr>
<tr>
<td>taxable</td>
<td>Integer</td>
<td></td>
<td>Yes if a new Account.</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>Tax Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Tax Exempt for all taxes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Taxable</td>
<td></td>
</tr>
<tr>
<td>taxregion</td>
<td>Integer</td>
<td></td>
<td>Yes if a new Account.</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>Tax Region Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;0</td>
<td>A reference to the number field in the taxregion table.</td>
<td></td>
</tr>
<tr>
<td>reseller</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>paytype</td>
<td>Integer</td>
<td>Default Payment Method</td>
<td>Yes if a new Account.</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Cash/Check</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Credit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Electronic Check</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>EFT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Credit Once, then change to Cash/Check</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>invmethod</td>
<td>String</td>
<td>Automated Billing Methods. This is contextual depending on the Account Profile. A &quot;0&quot; indicates no automated method is selected. Otherwise, a comma separated list such as &quot;1,2,4,16...&quot; indicates which Document Delivery Methods from the Account Profile should be used. Note each entry in the comma separated list is a power of 2.</td>
<td>Yes if a new Account.</td>
</tr>
<tr>
<td>addr1</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>addr2</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>addr3</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>addr4</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>city</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zip</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>country</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>county</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>invday</td>
<td>Integer</td>
<td>Billing/Prorate Day of Month. If 0, then the setting from the Virtual Company will be used.</td>
<td></td>
</tr>
<tr>
<td>gracedate</td>
<td>Date</td>
<td>Date an overdue Account has until Overdue processing is started.</td>
<td></td>
</tr>
<tr>
<td>misc01</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc02</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc03</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc04</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc05</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc06</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc07</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc08</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc09</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc10</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc11</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>misc12</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc13</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc14</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc15</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc16</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc17</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc18</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc19</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc20</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>customerof</td>
<td>Integer</td>
<td>Virtual Company. A reference to the number field in the config table.</td>
<td>Yes if a new Account.</td>
</tr>
<tr>
<td>generation</td>
<td>Time stamp</td>
<td>Blank Used for new Account</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYYY-MM-DD The time stamp of the existing Account record.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HH:MM:SS</td>
<td></td>
</tr>
<tr>
<td>reason</td>
<td>Integer</td>
<td>The reason an Account is not open. 0 or a value from the reasons List.</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>Integer</td>
<td>0 or a value from the titles List.</td>
<td></td>
</tr>
<tr>
<td>bmui</td>
<td>Integer</td>
<td>Portal Access.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 Portal Access disallowed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Portal Access allowed.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>-------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>useradmin</td>
<td>Integer</td>
<td>Account Administrative Portal User</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: No User specified</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1: Set to the first User created under this Account</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;1: An existing User of this Account</td>
<td></td>
</tr>
<tr>
<td>immune</td>
<td>String</td>
<td>Comma separate list of values, each value a power of 2.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Immune from automated suspension</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: Immune from Late Fees</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4: Will not receive overdue emails</td>
<td></td>
</tr>
<tr>
<td>profile</td>
<td>Integer</td>
<td>Account Profile. A reference to the number field in the profile table.</td>
<td></td>
</tr>
<tr>
<td>vataxid</td>
<td>String</td>
<td>Value Added Tax ID</td>
<td></td>
</tr>
<tr>
<td>datetoreopen</td>
<td>Date</td>
<td>If an account is not Open, the date in the future to reopen the account automatically.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>flags</td>
<td>String</td>
<td>Comma separate list of values, each value a power of 2.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Issue Sales</td>
<td>Returns for services billed in the future when Closing the Account.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Issue Deposit</td>
<td>Returns when Closing the Account.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Reopen Users and Services</td>
<td>that where Suspended or Closed when the Account was Suspended or Closed.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Issue Sales</td>
<td>Returns for period billed but Account was Suspended or Closed.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>enddatereason</td>
<td>Integer</td>
<td>Reason that the End Date has been set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 The End Date was set because the Account is closing. Normal Suspend then Close will take place.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 The End Date was set at the customer's request. The Account will be Suspended indefinitely until further action is taken.</td>
<td></td>
</tr>
<tr>
<td>autoeft</td>
<td>Integer</td>
<td>Automated Payment Processing modifiers.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comma separate list of values, each value a power of 2.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Disable all automate Collections and Refunds.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Disable Automated Refunds</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 Allow one Automated Refund, then disable.</td>
<td></td>
</tr>
<tr>
<td>importid</td>
<td>String</td>
<td>If imported from another billing system, the key in the old system for the account.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>pctdiscount</td>
<td>Decimal</td>
<td>Percent discount that all non-Usage Sales to the Account will get. Implemented as a Store Credit for each Sale. Supply &quot;2.5&quot; for a 2.5 percent discount.</td>
<td></td>
</tr>
<tr>
<td>pctdiscountexpdate</td>
<td>Date</td>
<td>Date that the discount expires.</td>
<td></td>
</tr>
<tr>
<td>usagepctdiscount</td>
<td>Decimal</td>
<td>Percent discount that all Usage Sales to the Account will get. Implemented as a Store Credit for each Sale.</td>
<td></td>
</tr>
<tr>
<td>usagepctdiscountexpdate</td>
<td>Date</td>
<td>Date that the Usage discount expires.</td>
<td></td>
</tr>
<tr>
<td>loginid</td>
<td>String</td>
<td>Portal Login Name for the Administrative User.</td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Portal Password for the Administrative User.</td>
<td></td>
</tr>
<tr>
<td>guarantor</td>
<td>Integer</td>
<td>Another Account that will pay the bills for this Account. If non-zero, a reference to the number field in the account table.</td>
<td></td>
</tr>
<tr>
<td>timezone</td>
<td>Integer</td>
<td>Value from the timezones List.</td>
<td></td>
</tr>
</tbody>
</table>

**enter_saleorcredit**

Used to create or modify a Pending Sale or Store Credit Transaction.

**Usage**

/usr/local/billmax/bin/enter_saleorcredit [options]

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s service</td>
<td>Service number</td>
</tr>
<tr>
<td>-c chargeType</td>
<td>Not Applicable for a Service or Item Sale. A value from the List chargetypes.</td>
</tr>
<tr>
<td>-m rate</td>
<td>Does not have to be a monetary amount. Can be fractional.</td>
</tr>
<tr>
<td>-d 'description'</td>
<td></td>
</tr>
<tr>
<td>-a account</td>
<td>Account Number</td>
</tr>
<tr>
<td>-q quantity</td>
<td></td>
</tr>
<tr>
<td>-M amount</td>
<td>The amount of the transaction. Generally should equal rate multiplied by quantity rounded</td>
</tr>
</tbody>
</table>
### Option | Notes
--- | ---
-T `taxable` | Not applicable for a Service Sale.  
  0 | Not Taxable  
  1 | Taxable  
  -1 | Tax Not Applicable  
-g `taxregion` | Not applicable for a Service Sale.  
  -1 | Not Applicable  
  >0 | A value from the `taxregion` table  
-f `taxclass` | Not Applicable for a Service or Item Sale. A value from the List `taxclasses`.  
-k `taxrule` | Not applicable for a Service Sale.  
  -1 | Not Applicable  
  >0 | A value from the `taxregion` table  
-b `startdate` | Value in the form YYYY-MM-DD.  
-e `enddate` | Value in the form YYYY-MM-DD.  
-F `item` | Value of an Item Definition.  
-U `usage` | Not Applicable for a General or Item Sale. Not applicable if -r specified  
  0 | Sale should not be treated as a usage sale  
  1 | Sale should be treated as a usage sale  
-r `recurring` | Not Applicable for a General or Item Sale. Not applicable if -U specified  
  0 | Sale should not be treated as a recurring sale  
  1 | Sale should be treated as a recurring sale  

**enter_service.cgi**  
Used to create or modify a Service or Package record.  

### Usage  
/usr/local/billmax/cgi-bin/enter_service.cgi  

### Environment  
All data is passed via environment variables of the form `FORM_variablename` with the following exceptions:  
- Dates are passed as `FORM_variablename_year`, `FORM_variablename_month` and `FORM_variablename_day`. A four digit year is required. Two digits for month and day respectively are required.  
- Times are passed as `FORM_variablename_hour`, `FORM_variablename_minute` and `FORM_variablename_second`. Two digits are required for each.  

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>srchnumber</td>
<td>Integer</td>
<td>If non zero, the number of an existing Service.</td>
<td>Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>number</td>
<td>Integer</td>
<td>The number to use for the Service. Recommended to use 0 for a new Service.</td>
<td>Yes</td>
</tr>
<tr>
<td>override</td>
<td>Integer</td>
<td>Used to change a Service number. Use of this variable is not recommended.</td>
<td></td>
</tr>
</tbody>
</table>

- **0**: Non-zero `srchnumber` must equal non-zero `number`.
- **1**: Non-zero `srchnumber` may not equal non-zero `number`. If `srchnumber` does not equal `number`, the Service number field will change to `number` from `srchnumber`. 
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| mkemail | Integer| Used to set the email field on an Account and User if the emails fields are blank.  

- **0**: Do not set either the Account or User email fields.
- **1**: If either the Account or User field is blank, the email field will be set using the `service.d01` field. If the field value does not contain the "@" character, then "@" and the Virtual Company Default Domain value will be concatenated to the field value when setting the email. |          |
<p>| refdeposit | Integer | Deprecated                                                                                                                                                                                                 |          |
| user | Integer | Number for user table representing User to which the Service belongs                                                                                                                                  | Yes if a new Service. |
| state | Integer | Status of the Service.                                                                                                                                                                                   | Yes if a new Service. |
|        |        | <strong>0</strong>: Open                                                                                                                                  |          |
|        |        | <strong>1</strong>: Closed                                                                                                                               |          |
|        |        | <strong>2</strong>: Suspended                                                                                                                            |          |
|        |        | <strong>3</strong>: Wait for Payment                                                                                                                    |          |
|        |        | <strong>5</strong>: Prospective                                                                                                                         |          |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>startdate</td>
<td>Date</td>
<td></td>
<td>Yes if a new Service</td>
</tr>
<tr>
<td>enddate</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>servdef</td>
<td>Integer</td>
<td>Number of the Service or Package Definition of the underlying Product of the Service or Package.</td>
<td>Yes if a new Service</td>
</tr>
<tr>
<td>package</td>
<td>Integer</td>
<td>Number of the Package to which the Service is associated.</td>
<td></td>
</tr>
<tr>
<td>service</td>
<td>Integer</td>
<td>Number of the Service to which this Service is associated. Only specify if this Service is a Service Fee</td>
<td></td>
</tr>
<tr>
<td>free</td>
<td>Integer</td>
<td>Usage is not recommended. If a Service is marked Free, no billing activity is recorded. A better method is to give a 100% discount.</td>
<td></td>
</tr>
<tr>
<td>taxable</td>
<td>Integer</td>
<td>Used in conjunction with the User Tax Region to assess taxes.</td>
<td>Yes if a new Service.</td>
</tr>
</tbody>
</table>

- `0` Service is not free.
- `1` Service is free.
- `-1` Tax Not Applicable
- `0` Tax Exempt for all taxes
- `1` Taxable
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsetup</td>
<td>Integer</td>
<td>0: Assess the Setup amount immediately&lt;br&gt;1: Do not assess the Setup amount&lt;br&gt;2: Assess the Setup amount when the Service is opened for the first time.</td>
<td></td>
</tr>
<tr>
<td>descr</td>
<td>String</td>
<td>Description. May be displayed on Invoice.</td>
<td></td>
</tr>
<tr>
<td>d01</td>
<td>String</td>
<td>Primary identifier for a Service.</td>
<td></td>
</tr>
<tr>
<td>copypw</td>
<td>Integer</td>
<td>If non-zero, the number of a Service from which to copy the password and plain text password fields to the current Service.</td>
<td></td>
</tr>
<tr>
<td>d02</td>
<td>String</td>
<td>A value that is encrypted when stored. Typically used for passwords.</td>
<td></td>
</tr>
<tr>
<td>d03</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d04</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d05</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d06</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d07</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d08</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d09</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d10</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>invdate</td>
<td>Date</td>
<td>Next Billing Date for a Service.</td>
<td>Yes if a new Service.</td>
</tr>
<tr>
<td>nextusagedate</td>
<td>Date</td>
<td>Next Usage Billing Date for a Service.</td>
<td>Yes if a new Service.</td>
</tr>
<tr>
<td>pop</td>
<td>Integer</td>
<td>Point of Presence&lt;br&gt;0 or a value from the popregions List.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>iprice</td>
<td>Money</td>
<td>First Price</td>
<td></td>
</tr>
<tr>
<td>iduration</td>
<td>Integer</td>
<td>First Duration. A value from the durations List.</td>
<td></td>
</tr>
<tr>
<td>icnt</td>
<td>Integer</td>
<td>Number of times to assess First Price. Must be greater than 0. &quot;1&quot; is suitable for a single recurring price.</td>
<td></td>
</tr>
<tr>
<td>price</td>
<td>Money</td>
<td>Second Price</td>
<td></td>
</tr>
<tr>
<td>duration</td>
<td>Integer</td>
<td>Second Duration. A value from the durations List.</td>
<td></td>
</tr>
<tr>
<td>cnt</td>
<td>Integer</td>
<td>Number of times to assess Second Price. If 0, Second Price and Third Price not used.</td>
<td></td>
</tr>
<tr>
<td>lprice</td>
<td>Money</td>
<td>Third Price</td>
<td></td>
</tr>
<tr>
<td>lduration</td>
<td>Integer</td>
<td>Third Duration. A value from the durations List.</td>
<td></td>
</tr>
<tr>
<td>lcnt</td>
<td>Integer</td>
<td>Number of times to assess Third Price. If 0, Third Price not used.</td>
<td></td>
</tr>
<tr>
<td>setup</td>
<td>Money</td>
<td>Setup amount</td>
<td></td>
</tr>
<tr>
<td>setup_taxable</td>
<td>Integer</td>
<td>Used in conjunction with the User Tax Region to assess taxes.</td>
<td></td>
</tr>
<tr>
<td>ponumber</td>
<td>String</td>
<td>Purchase Order Number</td>
<td></td>
</tr>
<tr>
<td>ou_notice</td>
<td>Float</td>
<td>Used for metered billing.</td>
<td></td>
</tr>
<tr>
<td>ou_susp</td>
<td>Float</td>
<td>Used for metered billing.</td>
<td></td>
</tr>
<tr>
<td>ouprime_notice</td>
<td>Float</td>
<td>Used for metered billing.</td>
<td></td>
</tr>
<tr>
<td>ouprime_susp</td>
<td>Float</td>
<td>Used for metered billing.</td>
<td></td>
</tr>
<tr>
<td>refaccount</td>
<td>Integer</td>
<td>The number of an existing referring Account.</td>
<td></td>
</tr>
<tr>
<td>misc01</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc02</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc03</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc04</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>misc05</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc06</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc07</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc08</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc09</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc10</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc11</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc12</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc13</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc14</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc15</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc16</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc17</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc18</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc19</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>misc20</td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>generation</td>
<td>Time stamp</td>
<td>Used for new Account</td>
<td>Yes</td>
</tr>
<tr>
<td>reason</td>
<td>Integer</td>
<td>The reason an Service is not open. 0 or a value from the reasons List.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>wdeposit</td>
<td>Integer</td>
<td>0: Assess the Deposit amount immediately. 1: Do not assess the Deposit amount. 2: Assess the Deposit amount when the Service is opened for the first time.</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>Float</td>
<td>Quantity to bill.</td>
<td></td>
</tr>
<tr>
<td>quantityfree</td>
<td>Float</td>
<td>Portion of Quantity that is not subject to billing.</td>
<td></td>
</tr>
<tr>
<td>cyclept</td>
<td>Integer</td>
<td>Point in a billing cycle. Used for cyclical billing.</td>
<td></td>
</tr>
<tr>
<td>taxrule</td>
<td>Integer</td>
<td>-1 or a number reference a Tax Rule from the taxrule table.</td>
<td></td>
</tr>
<tr>
<td>salesrep</td>
<td>Integer</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>datetoreopen</td>
<td>Date</td>
<td>If a Service is not Open, the date in the future to reopen the Service automatically.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>flags</td>
<td>String</td>
<td>Comma separate list of values, each value a power of 2.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Issue Sales Returns for Services/Packages billed in the future when Closing the Service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Issue Deposit Returns when Closing the Service/Package.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>Reopen Services of Packages that were Suspended or Closed when the Package/Package was Suspended or Closed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>Issue Sales Returns for period billed but Account was Suspended or Closed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64</td>
<td>Bill a following full term if the first billing is prorated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>256</td>
<td>Create a Package, not a Service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024</td>
<td>Display zero priced usage sales on Invoices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2048</td>
<td>Ignore Discounts.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>enddatereason</td>
<td>Integer</td>
<td>Reason that the End Date has been set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 The End Date was set because the Service is closing. Normal Suspend then Close will take place.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 The End Date was set at the customer's request. The Service will be Suspended indefinitely until further action is taken.</td>
<td></td>
</tr>
<tr>
<td>executehook</td>
<td>Integer</td>
<td>0 Do not execute the Service Hook program.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Execute the Service Hook program.</td>
<td></td>
</tr>
<tr>
<td>importid</td>
<td>String</td>
<td>If imported from another billing system, the key in the old system for the Service</td>
<td></td>
</tr>
<tr>
<td>pctdiscount</td>
<td>Decimal</td>
<td>Percent discount that all non-Usage Sales to the Account will get. Implemented as a Store Credit for each Sale. Supply &quot;2.5&quot; for a 2.5 percent discount.</td>
<td></td>
</tr>
<tr>
<td>pctdiscountexpdate</td>
<td>Date</td>
<td>Date that the discount expires.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>usagepctdiscount</td>
<td>Decimal</td>
<td>Percent discount that all Usage Sales to the Account will get. Implemented as a Store Credit for each Sale.</td>
<td></td>
</tr>
<tr>
<td>usagepctdiscountexpdate</td>
<td>Date</td>
<td>Date that the Usage discount expires.</td>
<td></td>
</tr>
<tr>
<td>discountamount</td>
<td>Money</td>
<td>A fixed amount for automatically generating a Store Credit.</td>
<td></td>
</tr>
<tr>
<td>cardid</td>
<td>String</td>
<td>Used for Prepaid Services.</td>
<td></td>
</tr>
<tr>
<td>pin</td>
<td>String</td>
<td>Used for Prepaid Services</td>
<td></td>
</tr>
<tr>
<td>vcarddef</td>
<td>Integer</td>
<td>Used for Prepaid Services</td>
<td></td>
</tr>
<tr>
<td>contractenddate</td>
<td>Date</td>
<td>Date representing the date the Contract for the Service expires.</td>
<td></td>
</tr>
<tr>
<td>locationuser</td>
<td>Integer</td>
<td>Number for user table representing alternative Location for the Service from User.</td>
<td></td>
</tr>
<tr>
<td>salesagent</td>
<td>String</td>
<td>Comma separated numbers representing Authorized Users for whom should included in commission reports for this Service</td>
<td></td>
</tr>
<tr>
<td>etfee</td>
<td>Money</td>
<td>Early termination fee.</td>
<td></td>
</tr>
<tr>
<td>dobilling</td>
<td>Integer</td>
<td>0 Do not billing the Service immediately.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Bill the Service immediately. This is the default.</td>
<td></td>
</tr>
</tbody>
</table>

Add a Item Sale

1. Login or su to the BillMax system user, typically billmax.

2. Execute `/usr/local/billmax/bin/enter_saleorcredit -a account -F itemdef -m rate -q quantity -M amount -T taxable -g taxregion -d description`

Add a General Sale

1. Login or su to the BillMax system user, typically billmax.
2. Execute `/usr/local/billmax/bin/enter_saleorcredit -a account -c chargetype -m rate -q quantity -M amount -T taxable -g taxregion -f taxclass -k taxrule -d description`

*Add a Service Sale*

1. Login or `su` to the BillMax system user, typically `billmax`.
2. Execute `/usr/local/billmax/bin/enter_saleorcredit -s service -m rate -q quantity -M amount -d description`

*Add a Credit Card to an Account (API)*

A method for adding a Credit Card to an Account outside of the Staff Portal or Customer Portal. The following is done on the BillMax server.

1. Populate the environment with the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORM_number</td>
<td>0</td>
</tr>
<tr>
<td>FORM_action</td>
<td>SAVE</td>
</tr>
<tr>
<td>FORM_paytype</td>
<td>2</td>
</tr>
<tr>
<td>FORM_account</td>
<td>Account Number</td>
</tr>
<tr>
<td>FORM_fname</td>
<td>First name</td>
</tr>
<tr>
<td>FORM_mname</td>
<td>Middle name</td>
</tr>
<tr>
<td>FORM_lname</td>
<td>Last name</td>
</tr>
<tr>
<td>FORM_company</td>
<td>Company</td>
</tr>
<tr>
<td>FORM_addr1</td>
<td>Address</td>
</tr>
<tr>
<td>FORM_city</td>
<td>City</td>
</tr>
<tr>
<td>FORM_statename</td>
<td>State</td>
</tr>
<tr>
<td>FORM_zip</td>
<td>Zip/Postal code</td>
</tr>
<tr>
<td>FORM_country</td>
<td>Country</td>
</tr>
<tr>
<td>FORM_phone</td>
<td>Phone number</td>
</tr>
<tr>
<td>FORM_email</td>
<td>Email address</td>
</tr>
<tr>
<td>FORM_batchmethod</td>
<td>0</td>
</tr>
</tbody>
</table>

| FORM_ccnum       | Credit Card number                                                 |
| FORM_cvnum       | CVV number                                                          |
| FORM_ccexp_month | Expiration month                                                    |
| FORM_ccexp_year  | Expiration year - 4 digit                                           | Credit card added to Account and used for automated payments |
| Reference | 156 |

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORM_recurnotok</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Credit card</td>
</tr>
<tr>
<td>FORM_recurnotok</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Prepaid/Gift Card</td>
</tr>
<tr>
<td>FORM_generation</td>
<td>blank</td>
</tr>
</tbody>
</table>

All variables are required, but ones not relevant may be left blank.

2. Execute `/usr/local/billmax/cgi-bin/enter_efpdata.cgi`. Success return `EX_OK` and “Status: 204 No Response\n\n” is written to `stdout`.

**Edge**

The edge interface supplies a WSDL SOAP interface to various functions within BillMax. To access the edge application must be running with network accessibility.

Acronyms used

RAE
Remote Application Entry - The record for the Remote Application in BillMax.

**edgeLogin**
Used to access the edge application.

**Input**

```xml
<SOAP-ENV:Body>
 <ns:edgeLogin>
   <protocol>3</protocol>
   <user>asdf</user>
   <password>asdf</password>
 </ns:edgeLogin>
</SOAP-ENV:Body>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>Integer</td>
<td>Value must be 3</td>
<td>Yes</td>
</tr>
<tr>
<td>user</td>
<td>String</td>
<td>Username of the RAE</td>
<td>Yes</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Password of the RAE</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Output**

**Important:** Every time `edgeLogin` is successful, the returned `sessionid` is modified. As a result, programming for concurrent access to edge services as identified by a unique user/password combination should have only one call to `edgeLogin`.

```xml
<SOAP-ENV:Body>
 <ns:edgeAppInfoStatus>
   <code>0</code>
   <message></message>
   <info>
     <number>0</number>
     <name></name>
     <sessionid></sessionid>
     <salesview></salesview>
     <defdomain></defdomain>
     <hotspoturl></hotspoturl>
 </ns:edgeAppInfoStatus>
</SOAP-ENV:Body>
```
<phone></phone>
<email></email>
<company>0</company>
<profile>0</profile>
<interface>0</interface>
<companyname></companyname>
<wwwurl></wwwurl>
<logourl></logourl>
<flags>0</flags>
<locsym></locsym>
</info>
</ns:edgeAppInfoStatus>
</SOAP-ENV:Body>

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>Integer</td>
<td>0 Success</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>info.number</td>
<td>Integer</td>
<td>Number of RAE</td>
</tr>
<tr>
<td>info.name</td>
<td>String</td>
<td>Name of RAE</td>
</tr>
<tr>
<td>info.sessionid</td>
<td>String</td>
<td>The current token used to access the edge application for the RAE. Use in subsequent calls to edge.</td>
</tr>
<tr>
<td>info.salesview</td>
<td>String</td>
<td>Salesview of RAE</td>
</tr>
<tr>
<td>info.defdomain</td>
<td>String</td>
<td>Default domain of the Virtual Company</td>
</tr>
<tr>
<td>info.hotspoturl</td>
<td>String</td>
<td>Hot spot URL of RAE</td>
</tr>
<tr>
<td>info.phone</td>
<td>String</td>
<td>Phone of RAE</td>
</tr>
<tr>
<td>info.email</td>
<td>String</td>
<td>Email of RAE</td>
</tr>
<tr>
<td>info.company</td>
<td>Integer</td>
<td>Company of RAE</td>
</tr>
<tr>
<td>info.profile</td>
<td>Integer</td>
<td>Profile of RAE</td>
</tr>
<tr>
<td>info.interface</td>
<td>Integer</td>
<td>Type of RAE</td>
</tr>
<tr>
<td>info.companyname</td>
<td>String</td>
<td>Not in use</td>
</tr>
<tr>
<td>info.wwwurl</td>
<td>String</td>
<td>Not in use</td>
</tr>
<tr>
<td>info.logourl</td>
<td>String</td>
<td>Not in use</td>
</tr>
<tr>
<td>info.flags</td>
<td>Integer</td>
<td>Flags from RAE</td>
</tr>
<tr>
<td>info.locsym</td>
<td>String</td>
<td>Local monetary symbol from Global Options</td>
</tr>
</tbody>
</table>

**getLists**
Retrieve allowable List data from BillMax

**Input**

```xml
<SOAP-ENV:Body>
    <ns:getLists>
```
```
<appsessionid>myappsessionid</appsessionid>
</ns:getLists>
</SOAP-ENV:Body>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>appsessionid</td>
<td>String</td>
<td>info.sessionid returned from edgeLogin</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Output**

```
<SOAP-ENV:Body>
<ns:listStatus>
<code>0</code>
<message>Success</message>
<list>
  <elems>
    <listlistname</list>
    <descr>description</descr>
    <value>value</value>
    <multi>0</multi>
    <explanation>explanation</explanation>
    <flags>1</flags>
  </elems>
  .
  .
</list>
</ns:listStatus>
</SOAP-ENV:Body>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>String</td>
<td>0 Success</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>list</td>
<td>Container for one or more elems</td>
<td></td>
</tr>
<tr>
<td>elems.list</td>
<td>String</td>
<td>Name of BillMax List to which this entry belongs.</td>
</tr>
<tr>
<td>elems.descr</td>
<td>String</td>
<td>Description of List Item</td>
</tr>
<tr>
<td>elems.value</td>
<td>String</td>
<td>Value of List Item - Integer or String depending on context.</td>
</tr>
<tr>
<td>elems.multi</td>
<td>Integer</td>
<td>0 If List is not invmethods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 If List is invmethods</td>
</tr>
<tr>
<td>elems.explanation</td>
<td>String</td>
<td>Explanation of List Item</td>
</tr>
<tr>
<td>elems.flags</td>
<td>Integer</td>
<td>Flags value of List entry</td>
</tr>
</tbody>
</table>

**Typical Result for titles List**

```
<elems>
```
Staff Interface

The edge staff interface provides access to BillMax data by way of authenticating an Authorized User using the auth table.

1. **Important:** An Authorized User must have Remote Applications/Staff checked to use this interface.

1. **Important:** If an Authorized User is going to use the Staff Portal and this interface concurrently, the Authorized User must have Concurrent Access set to Yes.

The following steps are used to access the edge staff interface:

1. Login to the edge application with edgeLogin on page 156.
2. Log the Authorized User into BillMax with authLogin on page 159.
3. To get data, use staffGetRecord on page 160.
4. To put data, use staffPutRecord on page 163.

Two PHP scripts are supplied as examples for access edge services. They are both located in /usr/local/billmax/src/bin/edge.

`authtest.php-dist`  Access edge services using SOAP routines.

`authcurltest.php-dist`  Access edge services using CURL routines with HTTPS POST.

`authLogin`  Access BillMax using edge services with authentication as an Authorized User.
Input

```xml
<SOAP-ENV:Body>
<ns:authLogin>
<appsessionid></appsessionid>
<loginid></loginid>
<password></password>
</ns:authLogin>
</SOAP-ENV:Body>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>appsessionid</td>
<td>String</td>
<td>info.sessionid returned from edgeLogin</td>
<td>Yes</td>
</tr>
<tr>
<td>loginid</td>
<td>String</td>
<td>Authorized User Login Id</td>
<td>Yes</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Authorizes User Password</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Output

```xml
<SOAP-ENV:Body>
<ns:edgeAuthInfoStatus>
<code>0</code>
<message></message>
<info>
<loginid></loginid>
.fname></fname>
 lname></lname>
<auth>0</auth>
<sessionid></sessionid>
</info>
</ns:edgeAuthInfoStatus>
</SOAP-ENV:Body>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>String</td>
<td>0 Success</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>info.loginid</td>
<td>String</td>
<td>Authorized User Login Id</td>
</tr>
<tr>
<td>info.fname</td>
<td>String</td>
<td>Authorized User First Name</td>
</tr>
<tr>
<td>info.lname</td>
<td>String</td>
<td>Authorized User Last Name</td>
</tr>
<tr>
<td>info.auth</td>
<td>Integer</td>
<td>Authorized User Number</td>
</tr>
<tr>
<td>info.sessionid</td>
<td>String</td>
<td>Session Id for the Authorized User. Use in subsequent requests for the Authorized User.</td>
</tr>
</tbody>
</table>

**staffGetRecord**

Used to retrieve data from a table in BillMax.

Input

```xml
<SOAP-ENV:Body>
```
Table 5:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>appsessionid</td>
<td>String</td>
<td>info.sessionid from edgeLogin</td>
<td>Yes</td>
</tr>
<tr>
<td>authsessionid</td>
<td>String</td>
<td>info.sessionid from authLogin</td>
<td>Yes</td>
</tr>
<tr>
<td>table</td>
<td>String</td>
<td>Name of a supported table in BillMax</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following tables are supported:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• account</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• user</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• service</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• servdef</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• config</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• profile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• efpdata</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ticket</td>
<td></td>
</tr>
<tr>
<td>number</td>
<td>Integer</td>
<td>number field for table in BillMax</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Return a zeroed dataset for the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• account</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• user</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• service</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• efpdata</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ticket</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; 0 Return the record that matches this value.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>form</td>
<td>String</td>
<td>Associated Form value from RAE Portal data.</td>
<td>By default the following are supported but may be locally customized: account table user table service table efpdata table</td>
</tr>
</tbody>
</table>

### Output

```xml
<SOAP-ENV:Body>
<ns:dataStatus>
  <code>0</code>
  <message></message>
  <data>
    <elems>
      <name></name>
      <list></list>
      <length>0</length>
      <stype>0</stype>
      <dtype>0</dtype>
      <permission>0</permission>
      <message></message>
      <flags>0</flags>
      <cdata/></cdata>
    </elems>
    <elems>
      <name></name>
      <list></list>
      <length>0</length>
      <stype>0</stype>
      <dtype>0</dtype>
      <permission>0</permission>
      <message></message>
      <flags>0</flags>
      <cdata/></cdata>
    </elems>
  </data>
</ns:dataStatus>
</SOAP-ENV:Body>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>String</td>
<td>0 Success</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>data</td>
<td>Container</td>
<td>Contains one or more <code>elems</code></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>data.elems.name</td>
<td>TABLE.field</td>
<td>Each name is table name in caps followed by a period followed by field name.</td>
</tr>
<tr>
<td>data.elems.list</td>
<td>String</td>
<td>Name of BillMax List if the field uses a List as source of options.</td>
</tr>
<tr>
<td>data.elems.length</td>
<td>Integer</td>
<td>Field length</td>
</tr>
<tr>
<td>data.elems.stype</td>
<td>Integer</td>
<td>1 string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 pointer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 float</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 double</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 special</td>
</tr>
<tr>
<td>data.elems.dtype</td>
<td>Integer</td>
<td>Internal Use Only</td>
</tr>
<tr>
<td>data.elems.permissions</td>
<td>Integer</td>
<td>Permissions as specified by the RAE Portal data for the form. A bit mask:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Write</td>
</tr>
<tr>
<td>data.elems.cdata</td>
<td>String</td>
<td>Present if elems.stype is 1, 2 or 6</td>
</tr>
<tr>
<td>data.elems.idata</td>
<td>Integer</td>
<td>Present if elems.stype is 3</td>
</tr>
<tr>
<td>data.elems.fdata</td>
<td>Float</td>
<td>Present if elems.stype is 4</td>
</tr>
<tr>
<td>data.elems.ddata</td>
<td>Double</td>
<td>Present if elems.stype is 5</td>
</tr>
<tr>
<td>data.elems.message</td>
<td>String</td>
<td>Populated by <code>staffPutRecord</code> for create or update for specified field denied.</td>
</tr>
<tr>
<td>data.elems.flags</td>
<td>Integer</td>
<td>Internal Use Only</td>
</tr>
</tbody>
</table>

**staffPutRecord**  
Used to create or update a record for the specified table.

**Input**  
When update a record, the fields `number` and `generation` must be passed and part of the `data` container. Only fields that are updated need be passed.

```xml
<SOAP-ENV:Body>
<ns:staffPutRecord>
  <appsessionid></appsessionid>
  <authsessionid></authsessionid>
  <create>0</create>
  <table></table>
  <form></form>
</ns:staffPutRecord>
```
<data>
<elems>
  <name></name>
  <list></list>
  <length>0</length>
  <stype>0</stype>
  <dtype>0</dtype>
  <permission>0</permission>
  <message></message>
  <flags>0</flags>
  <cdata></cdata>
</elems>
<elems>
  <name></name>
  <list></list>
  <length>0</length>
  <stype>0</stype>
  <dtype>0</dtype>
  <permission>0</permission>
  <message></message>
  <flags>0</flags>
  <cdata></cdata>
</elems>
</data>
</ns:staffPutRecord>
</SOAP-ENV:Body>

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>appsessionid</td>
<td>String</td>
<td>info.sessionid from edgeLogin</td>
<td>Yes</td>
</tr>
<tr>
<td>authsessionid</td>
<td>String</td>
<td>info.sessionid from authLogin</td>
<td>Yes</td>
</tr>
<tr>
<td>create</td>
<td>Integer</td>
<td>Denotes whether a new record is being created</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update existing record</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Create new record</td>
<td></td>
</tr>
<tr>
<td>table</td>
<td>String</td>
<td>Name of a supported table in BillMax</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following tables are supported:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• account</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• user</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• service</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• efpdata</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ticket</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>form</td>
<td>String</td>
<td>Associated Form value from RAE Portal data.</td>
<td>By default the following are supported but may be locally customized:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>account account table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>user user table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>service service table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>efpdata efpdata table</td>
</tr>
<tr>
<td>data</td>
<td>Container</td>
<td>Contains one or more elems</td>
<td></td>
</tr>
<tr>
<td>data.elems.name</td>
<td>TABLE.field</td>
<td>Each name is table name in caps followed by a period followed by field name.</td>
<td></td>
</tr>
<tr>
<td>data.elems.sstype</td>
<td>Integer</td>
<td>Specifies the type of data being passed</td>
<td>1 string</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 pointer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 float</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 double</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 special</td>
</tr>
<tr>
<td>data.elems.cdata</td>
<td>String</td>
<td>Present if elems.sstype is 1, 2 or 6</td>
<td></td>
</tr>
<tr>
<td>data.elems.idata</td>
<td>Integer</td>
<td>Present if elems.sstype is 3</td>
<td></td>
</tr>
<tr>
<td>data.elems.fdata</td>
<td>Float</td>
<td>Present if elems.sstype is 4</td>
<td></td>
</tr>
<tr>
<td>data.elems.ddata</td>
<td>Double</td>
<td>Present if elems.sstype is 5</td>
<td></td>
</tr>
<tr>
<td>data.elems.message</td>
<td>String</td>
<td>Text if update or create not allowed. Returned.</td>
<td></td>
</tr>
<tr>
<td>data.elems.flags</td>
<td>Integer</td>
<td>Internal Use Only</td>
<td></td>
</tr>
</tbody>
</table>

**Output**

See result of \textit{staffGetRecord} on page 160.

**Customer Interface**

The edge customer interface provide access to BillMax by way of authenticating a Customer using the \textit{user} table.
**Database Checking (dbck) Error codes**

Error codes generated by dbck. Will most likely be noticed in the BillMax nightly log.

<table>
<thead>
<tr>
<th>Code</th>
<th>Information or Error</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10003</td>
<td>Informational</td>
<td>The account has a Payment Method of Credit Card or Electronic Check. However, there is either no Credit Card or Bank Account on file or no Credit Card or Bank Account is marked to be used for Automated Payment Processing.</td>
</tr>
<tr>
<td>11500</td>
<td>Informational</td>
<td>The key used for AES encrypting the Credit Card and Bank Account Information for payment processing has expired. For PCI compliance it is recommended that it be changed at least every 90 days. The old key is needed to set a new key. If the old key has been forgotten, contact <a href="mailto:support@billmax.com">support@billmax.com</a> for recovery. Access to the system will be needed by BillMax Support.</td>
</tr>
<tr>
<td>13222</td>
<td>Error</td>
<td>The message indicates a bug in processing. Please contact <a href="mailto:support@billmax.com">support@billmax.com</a></td>
</tr>
</tbody>
</table>
| 10814  | Error/Informational  | The resources used for Provisioning on the Service do not match the Resources on the associated Service Definition. This condition will occur when the Resources on the Service Definition are changed after the Service was created. To correct:  
  
  Make a backup of the database  
  Execute the following SQL  
  UPDATE service, servdef SET  
  service.resources=servdef.resources  
  WHERE service.servdef=servdef.number AND  
  service.resources!=servdef.resources; |
| 10847  | Error/Informational  | Multiple non-closed Services that share a common resource bit have the same value for the service.d01 (typically login or MAC address) field. In addition, a related Service Definition does not allow duplicates. To fix change the value on one or more Services that have a conflict or modify the Service Definitions that share the common resource. |
| 13907  | Error                | The message indicates a bug in processing. Please contact support@billmax.com                                                                |

**Database Tables**

The following are fields that are common across most of the database tables:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>Unique Identifier</td>
</tr>
<tr>
<td>entdate</td>
<td>The date the record was created</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>enttime</td>
<td>The time the record was created.</td>
</tr>
<tr>
<td>generation</td>
<td>The date and time the record was last modified. Current value in database</td>
</tr>
<tr>
<td></td>
<td>must be submitted on update.</td>
</tr>
<tr>
<td>empl</td>
<td>The user that created the record. A reference to auth.number.</td>
</tr>
<tr>
<td>lastmodempl</td>
<td>The user that last modified the record. A reference to auth.number.</td>
</tr>
<tr>
<td>state</td>
<td>The status of a record: closed; open; suspended; active etc.</td>
</tr>
</tbody>
</table>

If the field of a table is populated using a BillMax List as a lookup, it will be noted by "List listname".

If the field is a foreign key, the table and field referenced will be listed as "table.field".

Fields named "misc##" where ## represents a two digit number are database fields for the BillMax customer's use. In general the HTML for these fields is hidden by default in the template associated with the table.

**account Table (Account)**

The Account table contains information related to the entity being billed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>adcode</td>
<td>Advertising Code. Used in Advertising report. List adcodes.</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>addr1</td>
<td>Address Line 1</td>
<td>char(50)</td>
</tr>
<tr>
<td>addr2</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>addr3</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>addr4</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>appointmentroute</td>
<td>Default Appointment Route. Used with Scheduling. List appointmentroutes.</td>
<td>int(11)</td>
</tr>
<tr>
<td>autoeft</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>bacct</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>balance</td>
<td>Account balance.</td>
<td>bigint(20)</td>
</tr>
<tr>
<td>bcity</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>bmui</td>
<td>Customer Portal access:</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td></td>
<td>1 Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 No</td>
<td></td>
</tr>
<tr>
<td>bname</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>brout</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>bstate</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>cancelreasons</td>
<td>Reason an Account canceled. List cancelreasons</td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>ccaddr</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>cccity</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>cccountry</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccdeclines</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccexp</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccfname</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>cclname</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccname</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccnexttry</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccnum</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccnumenc</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccpass</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccphone</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccpin</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>ccstatename</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>cctype</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>cczip</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>cfname</td>
<td>First Name</td>
<td></td>
</tr>
<tr>
<td>city</td>
<td></td>
<td>char(30)</td>
</tr>
<tr>
<td>clname</td>
<td>Last Name</td>
<td>char(30)</td>
</tr>
<tr>
<td>cmname</td>
<td>Middle Name</td>
<td>char(30)</td>
</tr>
<tr>
<td>company</td>
<td>Customer's company.</td>
<td>char(30)</td>
</tr>
<tr>
<td>country</td>
<td></td>
<td>char(30)</td>
</tr>
<tr>
<td>county</td>
<td></td>
<td>char(30)</td>
</tr>
<tr>
<td>cphone</td>
<td>Phone</td>
<td>char(30)</td>
</tr>
<tr>
<td>customerof</td>
<td>Virtual Company</td>
<td>int(11)</td>
</tr>
<tr>
<td>datetoreopen</td>
<td>Date in the future to automatically reopen an Account.</td>
<td>date</td>
</tr>
<tr>
<td>deposit</td>
<td>Total deposits held.</td>
<td>bigint(20)</td>
</tr>
<tr>
<td>email</td>
<td></td>
<td>char(100)</td>
</tr>
<tr>
<td>empl</td>
<td></td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>enddate</td>
<td>Date Account Closed or Date to Close. The date after end of service.</td>
<td>date</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>enddatereason</td>
<td>Reason for closing. List reasons</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>entdate</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>enttime</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>finactivity</td>
<td>Internal Use.</td>
<td></td>
</tr>
<tr>
<td>flags</td>
<td>Internal Use.</td>
<td></td>
</tr>
<tr>
<td>fphone</td>
<td>Fax.</td>
<td>char(40)</td>
</tr>
<tr>
<td>generation</td>
<td></td>
<td>timestamp</td>
</tr>
<tr>
<td>gracedate</td>
<td>Date through which reminders for overdue balances will not be sent and automated suspension will not occur.</td>
<td>date</td>
</tr>
<tr>
<td>guarantor</td>
<td>Account number of an Account in BillMax that will pay outstanding Debits for the current Account.</td>
<td>int(11)</td>
</tr>
<tr>
<td>immune</td>
<td>Bit Field. The following are bit values:</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td></td>
<td>1 Do not automatically Suspend</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 Do not assess Late Fees</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 Do not send Overdue Emails</td>
<td></td>
</tr>
<tr>
<td>importid</td>
<td></td>
<td>char(100)</td>
</tr>
<tr>
<td>invday</td>
<td>Billing/Prorate Day of Month</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>invvmethod</td>
<td>Billing Statement/Invoicing Methods. A bit mask that references First, Second, … from the Account Profile Document Delivery Methods</td>
<td>int(11)</td>
</tr>
<tr>
<td>lastmodempl</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>latedate</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>lockowner</td>
<td></td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>locktime</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc01</td>
<td>Integer Field through misc05</td>
<td>int(11)</td>
</tr>
<tr>
<td>misc02</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc03</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc04</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc05</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>misc06</td>
<td>Character (50) Field through misc10</td>
<td>char(50)</td>
</tr>
<tr>
<td>misc07</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc08</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc09</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc10</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc11</td>
<td>Date Field through misc15</td>
<td>date</td>
</tr>
<tr>
<td>misc12</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc13</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc14</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc15</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc16</td>
<td>Time Field through misc20</td>
<td>time</td>
</tr>
<tr>
<td>misc17</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>misc18</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>misc19</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>misc20</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>nlatenotes</td>
<td></td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>number</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>pastduebal</td>
<td>Past Due Balance</td>
<td>bigint(20)</td>
</tr>
<tr>
<td>payerid</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>paytype</td>
<td>Default Payment Method:</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td></td>
<td>1 Cash/Check</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 Credit Card</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 Electronic Check</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 EFT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 Credit Card Once, then Cash/Check</td>
<td></td>
</tr>
<tr>
<td>pctdiscount</td>
<td>Percent Discount for all Sales to this Account</td>
<td>decimal(10,6)</td>
</tr>
<tr>
<td>pcdiscountrxpdate</td>
<td>Date the Discount expires.</td>
<td>date</td>
</tr>
<tr>
<td>popupnote</td>
<td>Reference to record in notes table that is displayed when an Account is accessed in the Staff Portal.</td>
<td>int(11)</td>
</tr>
<tr>
<td>profile</td>
<td>Account Profile</td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>purchases</td>
<td>Sales amounts owed</td>
<td>bigint(20)</td>
</tr>
<tr>
<td>reason</td>
<td>Reason why Account is not Open. List reasons</td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>referredby</td>
<td>Referral Code</td>
<td>char(30)</td>
</tr>
<tr>
<td>reseller</td>
<td>Reseller ID</td>
<td>char(30)</td>
</tr>
<tr>
<td>sla</td>
<td>Service Level Agreement</td>
<td>int(11)</td>
</tr>
<tr>
<td>startdate</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>state</td>
<td>Status of the Account:</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td></td>
<td>0                              Open</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1                              Closed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2                              Suspended</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4                              Collections</td>
<td></td>
</tr>
<tr>
<td>statename</td>
<td>State or Province</td>
<td>char(30)</td>
</tr>
<tr>
<td>summarydate</td>
<td>Last date a Billing Statement was generated.</td>
<td>date</td>
</tr>
<tr>
<td>suspdate</td>
<td>Date the Account was Suspended.</td>
<td>date</td>
</tr>
<tr>
<td>taxable</td>
<td></td>
<td>tinyint(4)</td>
</tr>
<tr>
<td></td>
<td>1                              Account is subject to paying taxes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0                              Account is not subject to paying taxes. Examples are non-profit and government organizations.</td>
<td></td>
</tr>
<tr>
<td>taxregion</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td></td>
<td>-1                             Not Applicable. Only use if never collecting any type of tax.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>otherwise                      Foreign Key to taxregion.number</td>
<td></td>
</tr>
<tr>
<td>textmessage</td>
<td>Number to use to send text messages. If not integrated with SMS service, use email address for text message device.</td>
<td>char(100)</td>
</tr>
<tr>
<td>timezone</td>
<td>List timezones.</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>title</td>
<td>Salutation. List titles.</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>ucash</td>
<td>Amount of unused Payments.</td>
<td>bigint(20)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ucredit</td>
<td>Amount of unused Store Credit.</td>
<td>bigint(20)</td>
</tr>
<tr>
<td>usagepctdiscount</td>
<td>Percentage discount for Usage Sales.</td>
<td>decimal(10,6)</td>
</tr>
<tr>
<td>usagepctdiscountexpdate</td>
<td>Date percentage discount for usage sales expires.</td>
<td>date</td>
</tr>
<tr>
<td>useradmin</td>
<td>User authorized to make Account adjustment through the Customer Portal.</td>
<td>int(11)</td>
</tr>
<tr>
<td>vataxid</td>
<td>Value Added Tax ID or customer.</td>
<td>char(15)</td>
</tr>
<tr>
<td>verifiedaddr</td>
<td>Verification results if using Address Verification:</td>
<td>int(11)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Not Verified</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Verified</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Address Verification in use, but address saved even though Address Verification failed.</td>
</tr>
<tr>
<td>zip</td>
<td>ZIP/Postal Code</td>
<td>char(30)</td>
</tr>
</tbody>
</table>

**ticket Table (Ticket)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td></td>
</tr>
<tr>
<td>generation</td>
<td></td>
</tr>
<tr>
<td>empl</td>
<td></td>
</tr>
<tr>
<td>lastmodempl</td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td></td>
</tr>
<tr>
<td>originator</td>
<td>Who authored the data for the ticket.</td>
</tr>
<tr>
<td>entdate</td>
<td></td>
</tr>
<tr>
<td>enttime</td>
<td></td>
</tr>
<tr>
<td>user</td>
<td>user:number - deprecated</td>
</tr>
<tr>
<td>reference</td>
<td>The number of the record from the fromtable table.</td>
</tr>
<tr>
<td>queue</td>
<td>queue.number</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>state</td>
<td>0 Open</td>
</tr>
<tr>
<td></td>
<td>-1 Inactive</td>
</tr>
<tr>
<td></td>
<td>-1 Unassigned</td>
</tr>
<tr>
<td></td>
<td>1 Closed</td>
</tr>
<tr>
<td></td>
<td>2 Reopened</td>
</tr>
<tr>
<td>priority</td>
<td>Calculated from the queue problems and automatically updated if Service Level Agreement in effect.</td>
</tr>
<tr>
<td>assignedto</td>
<td>auth.number</td>
</tr>
<tr>
<td>problem</td>
<td>Populated from the queue problems</td>
</tr>
<tr>
<td>resolution</td>
<td>Populated from the problem resolutions</td>
</tr>
<tr>
<td>lastactdate</td>
<td>Date of last activity for the ticket.</td>
</tr>
<tr>
<td>lastacttime</td>
<td>Time of last activity for the ticket.</td>
</tr>
<tr>
<td>misc01</td>
<td></td>
</tr>
<tr>
<td>misc02</td>
<td></td>
</tr>
<tr>
<td>misc03</td>
<td></td>
</tr>
<tr>
<td>misc04</td>
<td></td>
</tr>
<tr>
<td>misc05</td>
<td></td>
</tr>
<tr>
<td>misc06</td>
<td></td>
</tr>
<tr>
<td>misc07</td>
<td></td>
</tr>
<tr>
<td>misc08</td>
<td></td>
</tr>
<tr>
<td>misc09</td>
<td></td>
</tr>
<tr>
<td>misc10</td>
<td></td>
</tr>
<tr>
<td>externalid</td>
<td>Used to populate reference to third party system.</td>
</tr>
<tr>
<td>fromtable</td>
<td>table to which the ticket is attached</td>
</tr>
</tbody>
</table>

**user Table (User)**

The User table is used for service locations and alternate contacts. Multiple User records may be linked to one Account.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>account</td>
<td>Foreign key to account.number</td>
<td>int(11)</td>
</tr>
<tr>
<td>addr1</td>
<td>Address Line 1</td>
<td>char(50)</td>
</tr>
<tr>
<td>addr2</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>addr3</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>addr4</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>Description</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>aphone</td>
<td>Additional Phone.</td>
<td>char(40)</td>
</tr>
<tr>
<td>appointmentroute</td>
<td>Appointment Route for service location. Used with Scheduling. List appointmentroutes.</td>
<td>int(11)</td>
</tr>
<tr>
<td>billinglabel</td>
<td>Additional information added to Location Based Billing Statements/Invoices</td>
<td>char(100)</td>
</tr>
<tr>
<td>censusblock</td>
<td></td>
<td>char(8)</td>
</tr>
<tr>
<td>censustract</td>
<td></td>
<td>char(8)</td>
</tr>
<tr>
<td>city</td>
<td></td>
<td>char(30)</td>
</tr>
<tr>
<td>cleartextpassword</td>
<td>Clear text password for Customer Portal logins. Storing of the clear text is controlled by List systemstate entry usercleartextpassword.</td>
<td>char(100)</td>
</tr>
<tr>
<td>company</td>
<td>Company name at service location.</td>
<td>char(50)</td>
</tr>
<tr>
<td>country</td>
<td></td>
<td>char(30)</td>
</tr>
<tr>
<td>county</td>
<td></td>
<td>char(30)</td>
</tr>
<tr>
<td>countyfips</td>
<td></td>
<td>char(8)</td>
</tr>
<tr>
<td>custsecret</td>
<td>Original use deprecated. May be used for miscellaneous data.</td>
<td>char(50)</td>
</tr>
<tr>
<td>dataphone</td>
<td>Data Phone.</td>
<td>char(30)</td>
</tr>
<tr>
<td>datetoreopen</td>
<td>Date in the future to automatically reopen a User.</td>
<td>date</td>
</tr>
<tr>
<td>dlnumber</td>
<td>Original use deprecated. May be used for miscellaneous data.</td>
<td>char(20)</td>
</tr>
<tr>
<td>dlstate</td>
<td>Original use deprecated. May be used for miscellaneous data.</td>
<td>char(30)</td>
</tr>
<tr>
<td>dob</td>
<td>Original use deprecated. May be used for miscellaneous data.</td>
<td>date</td>
</tr>
<tr>
<td>dphone</td>
<td>Day phone.</td>
<td>char(40)</td>
</tr>
<tr>
<td>email</td>
<td></td>
<td>char(100)</td>
</tr>
<tr>
<td>empl</td>
<td></td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>enddate</td>
<td>Date User Closed or Date to Close. The date after end of service.</td>
<td>date</td>
</tr>
<tr>
<td>enddatereason</td>
<td>Reason for closing. List reasons</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>entdate</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>enttime</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>flags</td>
<td>Internal Use</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>fname</td>
<td>First Name</td>
<td>char(30)</td>
</tr>
<tr>
<td>generation</td>
<td></td>
<td>timestamp</td>
</tr>
<tr>
<td>importid</td>
<td></td>
<td>char(100)</td>
</tr>
<tr>
<td>lastaccess</td>
<td>Last time Customer Portal used by this User.</td>
<td>int(11)</td>
</tr>
<tr>
<td>lastip</td>
<td>Last IP from which Customer Portal accessed.</td>
<td>char(16)</td>
</tr>
<tr>
<td>lastmodempl</td>
<td></td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>lname</td>
<td>Last Name</td>
<td>char(30)</td>
</tr>
<tr>
<td>location</td>
<td>Point object representing Latitude/Longitude</td>
<td>point</td>
</tr>
<tr>
<td>locationnote</td>
<td>Note about Service Location</td>
<td>text</td>
</tr>
<tr>
<td>loginid</td>
<td>Customer Portal user name.</td>
<td>char(100)</td>
</tr>
<tr>
<td>maiden</td>
<td>Original use deprecated. May be used for miscellaneous data.</td>
<td>char(30)</td>
</tr>
<tr>
<td>misc01</td>
<td>Integer Field through misc05</td>
<td>int(11)</td>
</tr>
<tr>
<td>misc02</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc03</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc04</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc05</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>misc06</td>
<td>Character (50) Field through misc10</td>
<td>char(50)</td>
</tr>
<tr>
<td>misc07</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc08</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc09</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc10</td>
<td></td>
<td>char(50)</td>
</tr>
<tr>
<td>misc11</td>
<td>Date Field through misc15</td>
<td>date</td>
</tr>
<tr>
<td>misc12</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc13</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc14</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc15</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>misc16</td>
<td>Time Field through misc20</td>
<td>time</td>
</tr>
<tr>
<td>misc17</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>misc18</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>misc19</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>misc20</td>
<td></td>
<td>time</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>mname</td>
<td>Middle Name</td>
<td>char(30)</td>
</tr>
<tr>
<td>modem</td>
<td>List modems</td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>mspeed</td>
<td>List mspeeds</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>nphone</td>
<td>Night Phone</td>
<td>char(40)</td>
</tr>
<tr>
<td>ntries</td>
<td>Number of failed login attempts using the Customer Portal</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>number</td>
<td></td>
<td>int(11)</td>
</tr>
<tr>
<td>os</td>
<td>List oss</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>password</td>
<td>Password for the Customer Portal</td>
<td>char(11)</td>
</tr>
<tr>
<td>pctdiscount</td>
<td>Percent Discount for all Sales to this User. Overrides Account Percentage Discount.</td>
<td>decimal(10,6)</td>
</tr>
<tr>
<td>pctdiscountexpdate</td>
<td>Date the Discount expires.</td>
<td>date</td>
</tr>
<tr>
<td>prevreason</td>
<td>The previous reason a User was not Open.</td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>prevstate</td>
<td>The previous Status of a non-Open User/</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>reason</td>
<td>The reason an User is not open.</td>
<td>mediumint(8)</td>
</tr>
<tr>
<td>reasonbyaccount</td>
<td>Internal Use Only.</td>
<td></td>
</tr>
<tr>
<td>resetdate</td>
<td>Date password reset request started.</td>
<td>date</td>
</tr>
<tr>
<td>resetttime</td>
<td>Time password reset request started.</td>
<td>time</td>
</tr>
<tr>
<td>resettoken</td>
<td>Token for password reset request.</td>
<td>char(32)</td>
</tr>
<tr>
<td>schedulearea</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>securityanswer1</td>
<td>Answer for securityquestion1</td>
<td>char(255)</td>
</tr>
<tr>
<td>securityanswer2</td>
<td></td>
<td>char(255)</td>
</tr>
<tr>
<td>securityquestion1</td>
<td>List securityquestions</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>securityquestion2</td>
<td></td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>sessionid</td>
<td>Session id for Customer Portal.</td>
<td>char(32)</td>
</tr>
<tr>
<td>ssn</td>
<td>Original use deprecated. May be used for miscellaneous data.</td>
<td>char(12)</td>
</tr>
<tr>
<td>startdate</td>
<td></td>
<td>date</td>
</tr>
<tr>
<td>state</td>
<td>Status of the User:</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Open</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Closed</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Suspended</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>statefips</td>
<td></td>
<td>char(2)</td>
</tr>
<tr>
<td>statename</td>
<td>State or Province</td>
<td>char(30)</td>
</tr>
<tr>
<td>suspdate</td>
<td>Date the User was Suspended.</td>
<td>date</td>
</tr>
<tr>
<td>taxregion</td>
<td>Tax region for Services billed at this location.</td>
<td>int(11)</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Applicable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Only use if never collecting any type of tax.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>otherwise</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Foreign Key to taxregion.number.</td>
<td></td>
</tr>
<tr>
<td>timezone</td>
<td>List timezones</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>title</td>
<td>Salutation. List titles</td>
<td>tinyint(4)</td>
</tr>
<tr>
<td>usagepctdiscount</td>
<td>Percentage discount for Usage Sales at this Service Location.</td>
<td>decimal(10,6)</td>
</tr>
<tr>
<td>usagepctdiscountexpdate</td>
<td>Date percentage discount for usage sales expires.</td>
<td>date</td>
</tr>
<tr>
<td>verifiedaddr</td>
<td>Address verified using Address Verification API - currently only USPS supported.</td>
<td>int(11)</td>
</tr>
<tr>
<td>zip</td>
<td>ZIP/Postal Code</td>
<td>char(30)</td>
</tr>
</tbody>
</table>

**config Table (Virtual Company)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td></td>
</tr>
<tr>
<td>generation</td>
<td></td>
</tr>
<tr>
<td>empl</td>
<td></td>
</tr>
<tr>
<td>lastmodempl</td>
<td></td>
</tr>
<tr>
<td>sortorder</td>
<td></td>
</tr>
<tr>
<td>entdate</td>
<td></td>
</tr>
<tr>
<td>enttime</td>
<td></td>
</tr>
<tr>
<td>state</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>company</td>
<td></td>
</tr>
<tr>
<td>addr1</td>
<td></td>
</tr>
<tr>
<td>addr2</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>addr3</td>
<td></td>
</tr>
<tr>
<td>addr4</td>
<td></td>
</tr>
<tr>
<td>city</td>
<td></td>
</tr>
<tr>
<td>statename</td>
<td></td>
</tr>
<tr>
<td>zip</td>
<td></td>
</tr>
<tr>
<td>country</td>
<td></td>
</tr>
<tr>
<td>phone</td>
<td></td>
</tr>
<tr>
<td>fax</td>
<td></td>
</tr>
<tr>
<td>taxid</td>
<td></td>
</tr>
<tr>
<td>vataxid</td>
<td></td>
</tr>
<tr>
<td>bcfname</td>
<td>Billing Contact First Name</td>
</tr>
<tr>
<td>bclname</td>
<td>Billing Contact Last Name</td>
</tr>
<tr>
<td>bcemail</td>
<td>Billing Contact Email</td>
</tr>
<tr>
<td>acfname</td>
<td>Administrative Contact First Name</td>
</tr>
<tr>
<td>aclname</td>
<td>Administrative Contact Last Name</td>
</tr>
<tr>
<td>acemail</td>
<td>Administrative Contact Email</td>
</tr>
<tr>
<td>primestart</td>
<td></td>
</tr>
<tr>
<td>primeend</td>
<td></td>
</tr>
<tr>
<td>ccexpwarn</td>
<td></td>
</tr>
<tr>
<td>cctries</td>
<td></td>
</tr>
<tr>
<td>ccspan</td>
<td></td>
</tr>
<tr>
<td>defdomain</td>
<td></td>
</tr>
<tr>
<td>ccdecipher</td>
<td>Deprecated</td>
</tr>
<tr>
<td>autocredit</td>
<td></td>
</tr>
<tr>
<td>syncinvdate</td>
<td></td>
</tr>
<tr>
<td>latetaxable</td>
<td></td>
</tr>
<tr>
<td>lateonlate</td>
<td></td>
</tr>
<tr>
<td>rounding</td>
<td></td>
</tr>
<tr>
<td>defaultprofile</td>
<td></td>
</tr>
<tr>
<td>btries</td>
<td></td>
</tr>
<tr>
<td>bspan</td>
<td></td>
</tr>
<tr>
<td>graphic</td>
<td></td>
</tr>
</tbody>
</table>
**nightly Typical Batch Processes**

The following is a brief description of the typical Batch Process executed during the processing of the nightly Batch Process Group.

**dbck**

Check the BillMax database for data issues. Some data issues are severe and should be fixed immediately. Others are warnings that something needs to be looked at.

**managefiles**

Uses a file of XML directives for file retention purposes. The XML file will specify if files should be compressed and how long they should be retained. As delivered BillMax uses the file `/usr/local/billmax/local/manageFiles`.

**closeofday**

The main processing program for BillMax. Uses the `systemstate` List item `closeofday date` to keep track of the last time it ran. This date is very important for catching up billing if the process was disabled for any reason. The following functions are performed by `closeofday`:

- **UpdateTaxRates**
  - Puts into effect scheduled tax rate changes on Tax Items.

- **OpenFutureStartdateServices**
  - Opens services that were created with the `Prospective Status` with a `Reason of Future Startdate` where the `Start Date` is today or in the past.

- **OpenDateToReOpenEntities**
  - Opens accounts, Users and Services that are in `Suspended Status` where `Date to Reopen/Date to Resume Billing` is today or in the past.

- **SuspendExpiredAccounts**
  - Change the `Status` of Open Accounts/Users/Packages/Services and underlying entities to `Suspended` where the `End Date` is today or in the past.

- **CloseSuspendedAccounts**
  - Change the `Status` of Open Service Definitions to `No New Users` where the `End Date` is today or in the past.

- **ClearDiscounts**
  - Remove the relevant `Discount information` from Accounts/Users/Packages/Services when `Date`
Discounts Expire and/or Date Usage Discounts Expire is today or in the past.

**ChangeToBilling**
Bill the billing changes needed for Service Changes. The actual change may be in the future, but due to pre-billing, the new rate may need to be assessed earlier. Billing results in Pending Transactions.

**ChangeToProvisioning**
Execute the provisioning changes needed for a Service Change.

**ChargeForUsage**
Bill for Usage. Billing results in Pending Transactions.

**ResetUsageCounters**
Clear usage counters for Services that have usage collected but not billed.

**ReopenOverUsageServices**
Services that were Suspended for over usage if a new term is in effect.

**BillServices**
Bill Services and Packages where the combination of Next Bill Date and Account Profile Bill Timing setting indicate that the Service should be billed. Billing results in Pending Transactions.

**SetNextUsageDates**
Set the Services Next Usage Date where ChargeForUsage was not used.

**UpdateStaticBalances**
Update the financial fields stored in the Account table that summarize the Account's financial status.

**ResetGraceDates**
Clears Accounts' Grace Dates that are in the past.

**SendUsageReports**
Send Usage reports to customers as indicated in Usage Tier Plans.

**PurgeSamples**
Delete billed Usage data based on Usage Tier
Plan **Purge Sample Frequency.**

**IssueReferralCredits** - Issue referral credits. This results in Pending Transactions.

**AssessLateFeesByMonth** - When applicable, assess Late Fees based on the monthly balance.

**AssessLateFeesByInvoice** - When applicable, assess Late Fees based on each overdue Invoice.

**ResetOverDueCounters** - Clear data tracking late notices if the Account is no longer overdue.

**ReopenOverDueAccounts** - Reopen any Accounts that were Suspended for being overdue but are no longer overdue.

**ReopenOverDueServices** - Reopen any Service that were Suspended for being overdue but are no longer overdue.

**SuspendOverDueAccounts** - Suspend overdue Accounts and Services.

**SetFinactivity** - Flag Accounts that have Pending Transactions.

**DoInvoicing** - If applicable, created Invoices from Pending Transactions.

**OpenWaitForPaymentServices** - Services that have Status *Wait For Payment* of which the billings have been paid.

**WarnCCexpiration** - Send an e-mail to customers who have a credit card on file with an upcoming expiration date.

**WarnServiceExpiration** - Send an e-mail to customers who have a Service with an upcoming *End Date*.

**WarnGraceDate** - Send an e-mail to customers who have a Service with an upcoming *Grace Date*. 
CloseNotes
Close expired Notes.

CloseExpiredServiceAgents
Commission entries that have an End Date today or in the past.

Commission various system notices that will be output to the log file.

CloseExpiredServiceAgents
Close Service Agent
Commission entries that have an End Date today or in the past.

NoticeLongSuspendAccounts
NoticeLongSuspendUsers
NoticeLongSuspendServices
NoticeAccountLocks
NoticeRefundsDue
NoticeLicenseExpiration
NoticeUnFundedRefunds

ProcessNightlyTemplates
Process any Customer Notices.

CollectStats
Collect statistics.

CleanTrackProcess
Delete old entries tracking Authorized Users use of BillMax.

RemovePaymentInfo
Remove credit card and bank information from Accounts that have been closed the requisite number of days.

efpbatch
There are two process, one for collecting via credit card and debit card, the other for collecting via electronic check.

finalize_billing
Assesses any Document Fees and creates a Billing Statement record if appropriate. Uses the systemstate List item finalize_billing date to keep track of the last time it ran. This date is very important for catching up billing if the process was disabled for any reason.

queue_billingdocs
Creates database records for Invoices and Billing Statements that will be rendered and sent to the customer via e-mail or print. Uses the systemstate List item queue_billingdocs date to keep track of the last time it ran for performance reasons.

send_documents
Creates the document XML data, renders the documents and sends them.

managedocuments
Uses a file of XML directives for database entry and file retention purposes for Documents. The XML file will specify how long various Document types should be retained. As delivered BillMax uses the file /usr/local/billmax/local/manageDocuments.
Part II

Customer Portal

Topics:

• Concepts
• How To
• Reference
Chapter 17

Concepts

Topics:
- Customer Portal Overview
- Architecture
Customer Portal Overview

The Customer Portal is what a customer uses to interact with BillMax. Using the Customer Portal a customer may do the following:

• Make a Payment
• Add a recurring Payment Method
• Change personal data
• Add, change and remove Services
• Create a new Account and order new Services
• Check for Service Availability during new Customer registration

Architecture

The Customer Portal uses a Remote Application (see Remote Applications on page 103) on the BillMax server. The client software is written in Laravel® and is expected to reside on a public web server that is not the BillMax server. The public web server needs to support PHP version 7.3 or later.
Chapter 18

How To

Topics:

- Setup the Customer Portal
- Customize the Customer Portal
Setup the Customer Portal

1. The portal requires at a minimum, PHP version 7.3. This version is not yet available via the standard CentOS® repositories. To install PHP 7.3 and related modules. As root perform:
   a) For CentOS7, 
   b) For CentOS 7.x®, rpm -i remi-release-7*.rpm.
   c) Execute yum-config-manager --enable remi-php73
   d) Execute yum install php
   e) Execute yum -y install httpd mod_ssl php php-cli php-common php-pdo php-xml php-mbstring php-gd
   f) systemctl restart httpd

2. Generate a new application encryption key:
   a) cd /usr/local/billmax/portal.v2
   b) php artisan key:generate

3. Each client portal instance requires a Remote Application entry. To create an entry, perform:
   a) Select System Administration > Remote Applications from the Main Menu.
   b) Select New from the Context Menu.
   c) Enter Name.
   d) Enter Description.
   e) Enter Email. Note that this field is only used for the old portal. The new portal uses a client side configuration instead.
   f) Enter Phone. Note that this field is only used for the old portal. The new portal uses a client side configuration instead.
   g) Choose Customer Portal for Type.
   h) Enter Portal URL. This is base URL for portal. Specify the https:<portal-host>/<path-to-portal>. This is used in email links for the forgotten password and login id recovery tasks. You may use https:<billmaxserver>/testportal if wanting to test off the BillMax server.
   i) Choose Company and Customer Profile. These values are used only for new customer registrations.
   j) Enter Application Login Name and Application Password. These will be used by the Customer Portal client software to authenticate access to the Customer Portal Server.
   k) Modify the Customer Portal Login Parameters as required. Note that the security questions and CAPTCHA options are presently not used in the new portal.

4. vi /usr/local/billmax/portal.v2/config/edge.php. Enter the username and password using the value specified in Application Login Name and Application Password previously entered via the Staff Portal. Make sure the server entry has the correct name or IP address and port. You may configure other settings in this file as needed.

5. Create and Install the Edge SSL certificate
   a) As the BillMax system user, execute /usr/local/billmax/bin/certs.pl.
   b) Supply values for the prompts. For every password/passphrase request, use the same value entered for the first request including the challenge password. Make note of the value for use in a subsequent step.

6. Install the edge service - CentOS 7.x®. As the root user:
   a) cp /usr/local/billmax/pkg/bx_edge.service /usr/lib/systemd/system
   b) systemctl enable bx_edge
   c) systemctl start bx_edge

7. Test functionality by pointing a browser at https://billmax-host/testportal where billmax-host is the address of the BillMax server. A view similar to what is shown indicate success. If you get a permission error, check the zbillmax.conf file in /etc/httpd/conf.d. In the /usr/local/
billmax/html section, change AllowOverride None to AllowOverride All and restart the Apache server.

8. Click Login to test access to the edge service.

9. It is recommended that all customizations to the portal be done on the BillMax host then "pushed" to the remote web server after each update. Doing this will ensure your portal remains current and your customization are retained across upgrades. To copy the files, perform:
   a) cd /usr/local/billmax/portal.v2
   b) tar --exclude=.svn -c -z -f /tmp/portal.v2.tgz . Be sure to use "." instead of "*" otherwise some hidden files will be excluded from the tar file.
   c) scp /tmp/portal.v2.tgz <remote-user>@<remote-host>:

10. When installing the portal at a remote site, it must be installed so that only the portal's public directory and it's contents are in the Apache web server document tree. This can be accomplished by setting (typically in a VirtualHost section) the DocumentRoot variable to the public directory path or a symbolic link may be used. As an example of the latter technique, the /usr/local/billmax/html/testportal link points to /usr/local/billmax/portal.v2/public.

11. When installing the portal at a remote site, configure the Apache server so the portal files are only served via HTTPS. Either disable port 80 or redirect port 80 traffic to HTTPS.

### Customize the Customer Portal

1. Edit /usr/local/billmax/html/portal.v2/config/edge.php to set static data and to enable/disable different aspects of functionality.

2. If customers will be allowed to change their service via the Customer Portal, add one or more entries to the changeclasses List. **Product Definitions** associated with a specific list entry will represent the pool from which the customer may choose when requesting a change in service. If the price increases, it is considered a **Service Change Up**, if the price decreases, it is considered a **Service Change Down**

3. To dynamically add items for purchase, edit the relevant **Product Definitions** and edit the **Customer Portal** settings on the **General** tab.
Customer Portal

- Sell Online: Yes
- Item Name: Basic Wireless Internet
- Item Description: 2MB down/2MB up
- Change Class: Unspecified
Chapter 19

Reference

Topics:

- Customer Portal API
- Portal REST Endpoints
Customer Portal API

Portal REST Endpoints

The REST endpoints for the Customer Portal may be qualified by the route segment /nm. If present, the results by the request will not display the built-in Portal menu.

/getAccount

/nm/getAccount
On the BillMax server after the Customer Portal is configured, the URL `/testportal/portaliframe.html` may be used to examine the various endpoints.
Part

III

Reports

Topics:

- FCC 477
- CDR Billing Reports
CDFR Billing Reports

Concepts

CDFR Billing Reports Overview

CDFR Keys Not Associated shows Keys that have not been loaded into BillMax but are not associated with a particular package/service for billing purposes. After the key has been added to the appropriate package/service, then the menu option Associate CDRs will make the association.

CDFR Listing lists all CDRs that are uploaded but are not associated with a package/service. The Bill To Service column will show the information that needs to be added to a D01 field on a package/service. From this report, the CDRs can be deleted or re-rated for the CDRs that are not yet billed.

CDFR Billing Usage shows the detailed billing lines for CDRs for a particular package/service, invoice, billing statement, or transaction.

Future CDR Billings lists the CDRs that are outstanding for a particular account.

CDFR Files lists the CDR files that have been uploaded into BillMax. The files can be deleted and reloaded as keys are added to packages/services.

FCC 499A is a report to provide the lines necessary to complete the required government report on a yearly basis. For full details, choose Show Detail. The detail can include CDRs or just the financial data from each account with VOIP services.

FCC 499Q is a report to provide the lines necessary to complete the required government report on a quarterly basis.
Concepts

FCC 477 Overview

The FCC requires providers of broadband internet, local telephone service, mobile wireless, and VOIP to report on their services and coverage semi-annually. BillMax generates FCC 477 report data for broadband internet and VOIP providers only.

The FCC 477 reports asks for two datasets. "Broadband subscription" refers to locations and service attributes for services subject to FCC broadband reporting. There are some speed minimum but other than that your wireless, fiber, dsl, etc. services apply. These are services provided by you and not services being resold, "Broadband deployment" refers to where you can currently provide broadband service and what type of service.

In order for BillMax to have the pertinent location data, it uses the service location - the address on the user of the service. The location data consists of the latitude and longitude as well as census tract. BillMax has a built-in service for generating this information from a valid address or the latitude and longitude (if known). A special license is required for this functionality.

On each service or package definition there is a FCC Classification section. This section records the service attributes and speeds needed.

The data that specifies the deployment in BillMax comes from POP Regions. For wireless, POP Regions represents towers typically but can also be Access Points if you wish. For each POP there is a section for location information and deployment. Together these parameters tell us something about the coverage and type of service.

After the data has been entered and verified above, there are still several steps needed to generate the report data. There are tools to help give confirmation and visibility to the data needed. BillMax will validate the data entered and issue errors for any missing items. BillMax will show you the POP Regions on a map so that the coverage generated can be checked to extend beyond all Service locations. A list of counties that you offer service must be provided to BillMax Support. BillMax Support will load the counties and then the Census block data will be merged with the coverage. After all data is validated and no errors remain, the FCC 477 data sets will be generated so that they can be uploaded to the FCC web site.

Related information
https://www.fcc.gov/economics-analytics/industry-analysis-division/form-477-resources

How To

Geocode User Locations
For the FCC 477, each User (Service Location) must be geocoded and include Census Tract and Block information.

1. Contact support@billmax.com for the Mapping module and ask to have all existing locations geocoded.
2. If the Mapping module is activated as each new Account is created, the User Address will automatically be geocoded based on the address entered.
3. If a more exact Latitude and Longitude have been collected, click the User number and enter the values under FCC Location Data and click Get Census From Location to override the location.

Verify Service/Packet FCC Classification
For the FCC 477, each Service/Package must be classified accordingly.

1. Click Billing Administration > Recurring Service,
2. For each Service definition that must be reported to the FCC, click the Service Definition number.
3. If a more exact Latitude and Longitude have been collected, click the User number and enter the values under FCC Location Data and click Get Census From Location to override the location.
Create POP Regions

Each tower or access point must be entered as a Pop Region to calculate the coverage for FCC 477 report.

1. Select System Administration>Pop Regions, and choose Yes for Enable Location. Select Save.
2. For each item that must be reported to the FCC, click Add.
3. Enter the name or identifier for the tower or access point and Save.
4. For each item, click the number in the Location column.
5. Enter the Postal Address for the item. If the Latitude and Longitude are known, then these values can be entered and then click Get Census From Location.
6. Enter all parameters under FCC Deployment Data and Save.

Validate Data Pass 1

Each Service/Package that is to be reported must be connected to a Pop Region to generate accurate data for FCC 477 report.

1. Select Reports and select FCC 477.
2. Click the type of services offered in Instruction 2.
3. Select Validate Data from the menu.
4. Use the Validation Results to correct the following errors:

<table>
<thead>
<tr>
<th>Error</th>
<th>Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid census data</td>
<td>User Address is not geocoded.</td>
<td>Verify address is valid and country is USA or enter Latitude and Longitude and click Get Census From Location. Save the User.</td>
</tr>
</tbody>
</table>

Generate Coverage

The FCC report requires coverage information from each POP Region.

1. Select Reports and select FCC 477.
2. Check the Pop Regions or click Select All on Instruction 4.
3. Select Generate Coverage from the menu.
4. Use the Service Map By Pop to make sure that coverage extends beyond all Service locations.

Merge Census Blocks

1. Select Reports and select FCC 477.
2. If the counties that services are offered has changed or never been provided to BillMax Support, select Show Service FIPS codes from the menu.
3. For Load Query, choose State and county fips codes
4. Run the report and send the results to support@billmax.com. Wait for confirmation from BillMax support before continuing.
5. Select Merge Census Blocks from the menu.
   This step must be done every time the coverage is generated.

Generate 477 Report

1. Select Reports and select FCC 477.
2. Click the type of services offered in Instruction 2.
3. Select Validate Data from the menu to make sure no errors remain.
4. View the Validation Results to correct the any outside of deployment errors.
   These are the options
Service location does not fall within the coverage of a POP.

Change the location data of the User for the Service. Adjust the POP location parameters to include the Service location and click Generate Coverage. Edit the coverage using Service Map by POP.

State/County Fips was incomplete

Run Show Service FIPS codes and resubmit to support@billmax.com. Merge the Census Blocka and Validate Data again.

5. Lastly, choose Generate FCC 477 Report data.
   A zip files of 2 CSV files is created to be uploaded to the FCC web site.
Part IV

Scheduling and Appointments

Topics:

- Concepts
- How To
Chapter 21

Concepts

Topics:

- BillMax Scheduling Overview
BillMax Scheduling Overview

Scheduling consists of Authorized Users' schedules, Appointments and a Calendar view.

Activating the Scheduling module in BillMax requires a license with Scheduling activated.

Authorized User schedule entries are entered via the Calendar accessed through Scheduling. One time daily and recurring weekly schedules may be created. In addition, timeslots may be used to designate Authorized User unavailability and an optional Unavailable Reason specified.

Each Authorized User may be color-coded for ease of identifying schedules when viewing the Calendar. Each Authorized user may be associated with one or more Job Class. Appointment Types will use the Job Class to select Assignees for that Appointment Type. Each Authorized User may be excluded from the Scheduling system to reduce the potential lists of Assignees.

When the Calendar is accessed via the Main Menu or the Calendar is accessed via the User/Package/Service tab of an Account, an entry made from the Calendar is an Appointment.

Appointments are classified by Appointment Types. Appointment Types have several pieces of information:

- Name
- What Job Classes an Authorized User must have to be selected for the Appointment.
- Data to be included for every document of the Appointment Type. This consists of free form text and a series of Name/Value pairs.
- The Image File Class if images are desired on the document.

Note that all documents are generated using the XSL file /usr/local/billmax/cfg/xsltfiles/fo/appointments.xsl. Regardless of the name of the Appointment Type, all Work Orders/Documents will structurally be the same unless this file is modified.

When creating an Appointment, the list of potential Assignees is comprised of Assignees where the Assignee Job Class intersects the Appointment Type Job Class.

There are several visual cues for Appointments when looking at the Calendar view:

- solid color
  - silver - the appointment is assigned to an Authorized User (Assignee) that has not been color coded.
  - grey - the appointment is not assigned.
  - other - the appointment is assigned to an Assignee that has been color-coded.
- Patterned with stripes of equal width - the Assignee may be scheduled but is unavailable.
- Patterned with stripes of differing widths - the Assignee is scheduled.

The patterned entries show only when a specific Assignee is selected for the Calendar display.

Emails

Emails may be configured to be sent upon Appointment creation and Appointment modification of the date, time or status. In addition nightly processing may be configured to send an Appointment reminder email the day before the appointment. By default emails are sent to both the Account and User emails.

Emails are sent using the System Email Templates mechanism. Email templates are:

- AppointmentScheduled - only sent if new appointment has the Status of Scheduled.
- AppointmentChange - includes if appointment is deleted.
- AppointmentReminder - sent during nightly processing the day before the appointment.

Variables available to the email templates are:

- ${user.*} - Fields from the User/Service Location record.
- ${account.*} - Fields from the Account associated with the User.
- ${config.*} - Fields from the Virtual Company associated with the Account.
- ${ACCOUNTNAME} - number, company, name on Account
• $\{USERNAME\}$ - number, name on User
• $\{EMAILS\}$ - Account and User emails combined
• $\{APPTDATE\}$ - normalized date of appointment
• $\{STARTTIME\}$ - start time of appointment - 12 hour format
• $\{ENDTIME\}$ - end time of appointment - 12 hour format
• $\{NEWDATETIME\}$ - a sentence reflecting the new date/time of an appointment.
• $\{NEWSTATUS\}$ - a sentence reflecting the new Status of the appointment.
Chapter 22

How To

Topics:

- *Jump Start Scheduling*
- *Prepare an Authorized User for Scheduling*
- *Setup Scheduling and Appointments*
- *Add Customer Appointment from the Account*
- *Generate Appointment Documents*
Jump Start Scheduling

Steps to setup Scheduling for the first time.

1. Authorized Users to be an Assignee.
   a) Select System Administration > Authorized User from the Main Menu.
   b) Select the Authorized User by clicking the number in the Number column.
   c) Click on the Calendar Color widget.
   d) Select the color by using the vertical bar to select the color family and clicking inside the large block to select the color. Make sure to click Choose to apply the selection. Hint: use light colors so that text (black) is legible when viewing the calendar.
   e) If to be assigned to an appointment, specify one or more Job Classes.
   f) If scheduling relevant, make sure May Schedule on Calendar is Yes.

2. Setup geographically related areas (Routes) to specify on Appointments. This is to minimize travel while providing customer service.
   a) Select Correspondence > Routes from the Main Menu.
   b) Add List Items beginning with a Value of 1.

3. Schedule Assignees
   a) Select Correspondence > Assignee Scheduling from the Main Menu.
   b) Click Weekly Schedule to enter an Assignee's default weekly schedule. In addition, enter the schedule of the Assignee by clicking Save/Schedule.
   c) Click on a calendar time/date cell to schedule an individual time slot for Availability or Unavailability of an Assignee.

4. Review the setting of each default Appointment Types via Correspondence > Assignee Types.

5. Review the Calendar Defaults via Correspondence > Calendar Defaults.
   • starttime - the default Calendar Start Time when displaying the Calendar and there are no Appointments before this time.
   • endtime - the default Calendar End Time when displaying the Calendar and there are no Appointments after this time.
   • appointmentlength - the default length of an Appointment. Used to compute the End Time of a new Appointment.
   • displaydays - the default number of Calendar day to display.
   • defaultstatus - the default Status for a new Appointment.
   • defaultappointmenttype - the default Appointment Type for a new Appointment.
   • schedulelength - the default length of a one time Schedule timeslot entered for an Assignee.

6. Review the Templates for automated emails via Correspondence > System Email Templates
   Relevant Templates are:
   • AppointmentScheduled
   • AppointmentChange
Prepare an Authorized User for Scheduling

1. Select **System Administration > Authorized User** from the Main Menu.
2. Select the Authorized User by clicking the number in the Number column.
3. Click on the **Calendar Color** widget.
   
   ![Calendar Color Widget]
   
   4. Select the color by using the vertical bar to select the color family and clicking inside the large block to select the color. Make sure to click **Choose** to apply the selection. Hint: use light colors so that text (black) is legible when viewing the calendar.

5. Specify one or more **Job Classes**.
6. If scheduling relevant, make sure **May Schedule on Calendar** is **Yes**.

Setup Scheduling and Appointments

Different options for managing Scheduling and Appointments

Access the following using **Correspondence** from the Main Menu.

1. **Assignee Scheduling** - add an individual time block for an assignee and/or create a recurring Weekly Schedule.
2. **Calendar Defaults** - settings to manage the default behavior of the Calendar.
3. **Reasons** - setup reason why an Appointment has a certain Status.
4. **Routes** - setup geographically related areas to specify on Appointments. This is to minimize travel providing customer service.
5. **Statuses** - specify Statuses available for Appointments.
6. **Types** - specify different types of Appointments

Add Customer Appointment from the Account

1. Open an Account.
2. Click the **User/Package/Service** tab.
3. Click the User **wrench** and select **Schedule Appointment**.
4. Click on a Date/Time to schedule Appointment.
5. Specify Appointment options and click **Save**.
Generate Appointment Documents

1. Select Reports > Customer Appointments from the Main Menu.
2. Specify the options desired.
3. Click Get Documents.
Ticketing

Topics:

- Concepts
- How To
# Chapter 23

## Concepts

**Topics:**
- BillMax Ticketing Overview
- Messages and Notes
- Service Level Agreements (SLAs)
- Issues and Resolutions
- Team Members and Ticket Subscribers
- Tickets
- Queues
BillMax Ticketing Overview

Ticketing is the BillMax solution for tracking issues. These issues are associated with Accounts at the User level. Tickets may be created and responded to using the Staff Portal, the Customer Portal and email.

Ticketing is composed of the following components:

- Messages and Notes
- Service Level Agreements (SLAs)
- Issues and Resolutions
- Team Members and Ticket Subscribers
- Appointments - see Scheduling and Appointments on page 201
- Tickets
- Queues
- Activity Log

The following are definitions that are helpful in understanding the subsequent documentation:

**Authorized User**
A person authorized to use BillMax via System Administration > Authorized Users

**Assignee**
The Authorized User to whom a Ticket is assigned.

Messages and Notes

Messages and Notes are two different ways of storing data in a Ticket.

A Message is the recommended mechanism for storing communications related to a Ticket. If a Queue has been configured for fetching emails, a Message is the only mechanism by which fetched emails are automatically stored within a Ticket. If a Queue has been configured for sending email, the data and attachments of a Message are sent in the email. The following are characteristics of Messages:

- Messages may be created via email. If attachments are included as part of the email, they are stored with the Message.
- Messages may be created using the Staff Portal.
- Messages received via email from the customer may be distributed via email to Ticket Subscribers via email. This is configurable.
- Messages received via email from BillMax Staff may be distributed via email to Ticket Subscribers as well as the customer. This is configurable.
- Messages may be viewed and created by a customer using the Customer Portal. This is configurable.
- Messages may not be edited.
- Messages created via the Staff Portal have "Edit as New" functionality. The end result is a new Message.
- Messages belonging to a Ticket may be moved to a different Ticket.
- Messages may be moved to a new Ticket whilst creating the new Ticket.
- Messages have an HTML editor in the Staff Portal and Customer Portal.

Notes associated with Tickets are the same Notes used throughout BillMax. The following are characteristics of Notes:

- They are used to store simple text.
- They are created through the Staff Portal only.
- They may be edited and deleted.

The intended use of Notes is for comments about the Ticket, not communications about the Issue.
Service Level Agreements (SLAs)

SLAs are the mechanism for establishing an initial Ticket Priority, a Time to Complete and Ticket reminders based on the Issue of the Ticket. An SLA specified on the Queue is consulted if no SLA is specified on the Account.

The implementation of the SLA is through Service Level Agreement Items (SLA Items). An SLA may have multiple SLA Items. Each SLA Item encapsulates the performance metrics for an Issue Class. An SLA Item with the Issue Class Default is used if an SLA is in effect but no SLA Item has a matching Issue Class from the Ticket. An example of selection of the SLA and SLA Item is as follows:

1. Assume a Ticket is created in the billing Queue with the Issue Credit Card Expired that is part of the Credit Card Issues Issue Class.
2. Does the associated Account have an SLA? If so, use the Account's SLA. If not, does the Queue have an SLA? If so, use the Queue SLA. If not, no SLA is in effect. No performance metrics will be added to the Ticket.
3. Assuming an SLA is in effect, either Account or Queue, does the SLA have an SLA Item with Issue Class Credit Card Issues? If so, use the SLA Item's performance metrics for the Ticket. If not, does the SLA have an SLA Item with Issue Class Default? If so, use the SLA Item's performance metrics for the Ticket. If not, no performance metrics will be added to the Ticket.

The SLA Item encapsulates the performance rules for an Issue Class. An SLA Item consists of:

- The Issue Class.
- The time in minutes allotted to complete the task and close the Ticket.
- The initial Priority of the Ticket.
- The amount to increase the Priority of a Ticket every time an increase is warranted.
- The amount of time between Priority increases.
- The number of Priority increases after which the amount of time between Priority increases will be cut in half. This will happen repeatedly after every number of Priority increases.
- The minimum amount of time allowed between Priority increases.

Note: Automated Escalation of Tickets only occurs if escalatetickets is run periodically. This is typically done in a Batch Process Group.

Issues and Resolutions

A Queue is made up of Issues that are appropriate to the Queue. For example, the billing queue might have the following Issues (with the Issue Class in parentheses):

- Billed for Service not ordered (Default)
- Missing Invoice (Default)
- Payment Sent/Not Received (Default)
- Credit Card Expired (Credit Card Issues)
- Credit Card Declined (Credit Card Issues)
- Credit Card double charged (Credit Card Issues)

Each Issue will have one or more Resolutions. Example for Credit Card Declined are:

- New card provided
- Check being sent
- Bank account for ECH provided

The Issue Class noted in parentheses above is used as part of the SLA Item determination.
Team Members and Ticket Subscribers

Team Members

Each Queue may have a list of Team Members. A Team Member has the following options:

- **Supervisor** - Yes or No. A Supervisor of a Queue may have more functionality than a non-Supervisor depending on the Queue settings.
- **Round Robin** - Yes or No. If a Queue is set to do automated assignment of Tickets via round robin selection, this setting determines whether the Team Member is available for selection in the round robin. This is useful for excluding Team Members such as Supervisors or Team Members that are unavailable.

A Team Member also has settings that control whether or not notifications of Ticket Updates or Messages are sent to them. There are settings for when the Team Member is the Assignee and when just a monitor. The following settings are used as defaults when adding a Team Member as a Ticket Subscriber. This facilitates per Ticket control over receiving Ticket Updates or receiving Messages. Ticket Updates are sent only if the System Email Templates template **Ticket Notification is Active**. The settings are:

- **Ticket Updates** - Ticket Updates for which emails are sent are:
  - New - A new Ticket is created.
  - Assignment - A Ticket has a new Assignee. If a Assignee is changed from one Authorized User to another, an email will be sent to both parties
  - Status
  - Queue
  - Priority - An email will be sent only if the Priority changes and the Priority Threshold of the Ticket Subscriber is met or exceeded.
  - Issue
  - Delete - A Ticket is deleted.
- **Receive Messages** - This denotes whether or not Messages added to a Ticket are forwarded to the Ticket Subscriber.

Ticket Subscribers

Ticket Subscribers are Authorized Users that are monitoring the progress of a Ticket. They optionally may choose to receive Messages when emailed.

Team Members of a Queue are by definition a Ticket Subscriber of Tickets associated with the Queue. The Team Member may choose to add a Ticket Subscriber record to customize the settings for a particular Ticket.

In additions to Team Members, any Open Authorized User may be a Ticket Subscriber if allowed by the Queue. A Assignee that is not a Team Member is always a Ticket Subscriber.

Ticket Subscribers may be removed from Tickets. In addition, **Ticket Updates**, **Receives Messages** and **Priority Threshold** may be set on a per Ticket basis for a Ticket Subscriber.

If the Queue on a Ticket is changed, existing Ticket Subscribers are removed. If the Issue on a Ticket is changed and Service Level Agreements are in place, the Priority Thresholds for the Ticket Subscribers will be reset.

If settings for receiving Messages on the Ticket Subscriber conflict with the Queue settings, the most restrictive setting will be used. In other words, if the Queue setting does not allow sending of Messages, even if the Ticket Subscribers requests it, Messages will not be sent. However, if the Queue does allow Messages to be sent and the Ticket Subscriber does not want to receive Messages, Messages will not be sent to the Ticket Subscriber.
Tickets

Tickets represent a single Issue to be addressed. A Ticket is associated with a Queue and is composed of the following:

- An **Issue** to be resolved.
- The **Resolution** of the **Issue**
- The **Status**.
- The **Assignee**.
- Performance metrics for the Ticket if an **SLA Item** can be determined for the Ticket Issue. These include:
  - **SLA Deadline** - the date and time at which the Ticket should be resolved.
  - **Priority** - the initial Priority of a Ticket is taken from the **SLA Item**. The Priority is increased through time based on **SLA Item** settings if the program `escalatetickets` is activated through **Batch Processing**. If a Ticket Issue is changed, the Ticket Priority will be reset and recalculated based on the **SLA Item** performance metrics of the new Issue from the creation time of the Ticket.
  - **Next Priority Increase**

Multiple Tickets may be merged together. They will be merged to the oldest Ticket in the group.

A single Ticket may be split into multiple Tickets by creating new Tickets from Messages. For SLA purposes, the date and time of the new Ticket will be used when determining performance metrics for the new Ticket.

Associated with a Ticket are:

- Messages
- Notes
- Appointments
- Files
- Ticket Subscribers

Queues

Queues are used to group Tickets based on issue similarity. Examples are **billing**, **sales** and **technical support**. The following settings are also part of Queues:

- Functionality
- Assignment
- Outbound Email
- Inbound Email
- Outbound Ticket Updates
- Team Members
- Issues and Resolutions

**Functionality**

A Queue may be configured as to whether or not the following are in use:

- Service Level Agreements
- Messages
- Appointments
- The **On Hold** Status.
Assignment

A Ticket may be assigned to one Authorized User only and only by an Authorized User. The group of Authorized Users to whom a Ticket may be assigned is called the Assignee Pool. The Assignee Pool may be limited to Team Members of the Queue or may consist of every Open Authorized User.

Assignment Authority determines which Authorized Users are allowed to assign a Ticket. A Team Member designated as a Supervisor is always allowed to assign a Ticket to anyone in the Assignee Pool. Assignment Authority may be limited to Team Members or granted to all Authorized Users. An Authorized User with Assignment Authority may be limited to assigning a Ticket to them self if To Self Only is set to Yes. The To Self Only setting does not apply to a Queue Supervisor.

At creation, an unassigned Ticket may be automatically assigned to a designated Authorized User or be automatically assigned to Team Members on a round robin basis. Exclusion of Team Members from the round robin may be done individually. Round robin does not include Authorized Users that are not Team Members of the Queue.

Outbound Email

Outbound Email settings determine where Messages are sent and what notifications will be automatically sent to the customer.

Using the Outbound setting, sending Messages from BillMax may be:

- Disabled completely.
- Sent to Authorized Users only.
- Sent to the non Authorized Users that are part of the email as well as Authorized Users.

Sending of Messages automatically created via incoming email and those created using the Customer Portal are governed solely by these settings. Messages created through the Staff Portal have more control over the non Ticket Subscriber recipients.

The From Email setting must match the name of the Queue if receiving email into the Queue. For example using the domain example.com, if the Queue is named support, then the From Email needs to be named support@example.com. support@example.com is the email address to which customers are expected to send support email.

Inbound Email

Inbound Email setting determine how email is retrieved using the fetchmail program. Note if email aliases are being used to funnel all emails through a single email box, only one Queue should have Use Gateway set to Yes. If Outbound Email is configured to send email and Use Gateway is Yes, then Email Address and From Email need to be the same value.

Outbound Ticket Updates

Outbound Ticket Updates use System Email Templates to communicate with Customers and Authorized Users about ticket activity. For updates sent to Authorized Users, the System Email Templates ticketnotification is used. If the List entry StaffPortalUrl of the systemstate List is configured, a link to the ticket will be part of the notification.

Settings are:

- Notifications From Email - It is recommended that this be an email different from the From Email for Outbound Messages so that email loops with auto responders do not occur.
- Acknowledge New - the System Email Template to use when notifying the customer a Ticket has been created for them.
- Acknowledge Close - the System Email Template to use when notifying the customer a Ticket has been closed for them.

Other setting within the Queue control:

- Message Creation - controls what Authorized Users are allowed to create Messages for a Ticket. Choices are Team Members Only, Assignee Only, and Anyone (any Authorized User).
• **Ticket Subscriber** - controls what *Authorized Users* is allowed to be a Ticket Subscriber. Choices are *Team Members Only* and *Anyone*.

• **Ticket Deletion** controls what *Authorized Users* are allowed to delete a Ticket. Choices are *Team Members Only*, *Supervisor Only*, *Anyone* and *No Deletion*.

• **Assignee Defaults** - These are used when the Assignee is not a Queue Team Member. Otherwise the Team Member setting are used.
Chapter 24

How To

Topics:

• Enabling Inbound Email
• Enabling Outbound Email Messages
• Enabling Ticket Notifications
• Jump Start Ticketing
• Jump Start SLA Use
• Stop Assignment of Tickets to Certain Authorized Users
Enabling Inbound Email

Steps to add inbound email functionality to a Queue. Email is downloaded from an email box using `fetchmail`.

- An external email address or email alias is needed for each Queue that uses the Gateway. The preferred protocol is IMAP.
- An alias may not be used unless there is at least one Queue for which the Email Address is not an alias. Therefore to use an alias, you must have at least two Queues.

1. Login to the BillMax server as root.
   a) `cd /usr/local/billmax/bin`
   b) Execute `./setup_ticketing`
2. Login to the Staff Portal.
3. Select `Correspondence > Queues` from the Main Menu.
4. Select New to create a Queue or select an existing Queue that will be the first Queue that will receive emails and create Messages from the emails.
5. For the Queue:
   a) Set Use Gateway to Yes.
   b) Set Email Address to the name of the email box. The user-name part of the value must be the same as the Queue name.
   c) Set Alias to No
   d) Fill in the rest of the Inbound Email fields with data needed to retrieve emails from the email box using `fetchmail`.
6. For subsequent Queues:
   a) Set Use Gateway to Yes.
   b) Set Email Address to the name of an email box or an email alias. The user-name part of the value must be the same as the Queue name.
   c) Set Alias to No if an email box, Yes if an email alias.
   d) If Alias is No, fill in the rest of the Inbound Email fields with data needed to retrieve emails from the email box using `fetchmail`.

Enabling Outbound Email Messages

Steps to add outbound email functionality to Ticketing. Messages may be sent as emails to customers and/or Authorized Users.

- sendmail based outbound email capability is required for your BillMax server. This requirement is the same as required to send statements in email. This requirement usually requires specific DNS setup (SPF records) or the use of a smart host relay.
- Each Authorized Users on page 122 that will use outbound Email must have a unique email address and the email address must not be an email address of a User associated with an Account.

1. Login to the Staff Portal.
2. Select `Correspondence > Queues` from the Main Menu.
3. Select New to create a Queue or select an existing Queue.
4. For the Queue:
   a) Enter the From Email under Outbound Message as Email. If expecting replies to the outbound email to be automatically entered in the Queue, this value must be the Name@domain. An example is "support@example.com" for a Queue named "support". This will be the email "From" address when sending Messages as emails.
   b) Set Outbound Messages to Yes if Messages are to be emails to customers and Authorized Users or to Internal Only if just to Authorized Users.
Enabling Ticket Notifications

Notifications may be sent to Customers and Authorized Users concerning Ticket activity.

- sendmail based outbound email capability is required for your BillMax server. This requirement is the same as required to send statements in email. This requirement usually requires specific DNS setup (SPF records) or the use of a smart host relay.
- Each Authorized Users on page 122 that will use outbound Email must have a unique email address and the email address must not be an email address of a User associated with an Account.

1. Set up Authorized User notifications:
   a) Configure the StaffPortalUrl in the systemstate List if a link to the Ticket is desired.
   b) Make the System Email Template ticketnotification Active.

2. Set up Customer notifications. Each Queue may have its own templates.
   a) Set Acknowledge New Template.
   b) Set Acknowledge Close Template.

3. Set the Notifications From Email. It is recommended that this be a different email than the From Email used for Outbound Messages as Email.

Jump Start Ticketing

Minimal steps to setup Ticketing for the first time. The result is a Ticketing Queue that will allow all Open Authorized Users to manage Tickets in the Queue.

1. Login to the Staff Portal.
2. Select Correspondence > Queues from the Main Menu.
3. Select New and create a Queue.
   a) Enter the Name with letters and number only. A value suitable for the user-name part of an email. An example is "support".
   b) Enter the Description.
   c) Enter the Notifications From Email under Outbound Ticket Updates. This value is the email "From" address when sending Ticket Updates. As replies to ticket updates are not expected, a value such as "Example Inc. Support <DoNotReply@example.com>" might be used.
   d) Enter at least one Issue with one Resolution
   e) Accept all other defaults.

Jump Start SLA Use

The result is a Queue where all Issues are subject to an SLA.

1. Create the SLA
   a) Login to the Staff Portal.
   b) Select Correspondence > SLAs from the Main Menu.
   c) Select New and create an SLA.
   d) Add an Service Level Item with an Issue Class of Default.
   e) Choose a Suggested Time to Completion Settings.
   f) Save.
2. Add the SLA to a Queue
   a) Select Correspondence > Queues from the Main Menu.
   b) Select an existing Queue.
   c) Set the Service Level Agreement (SLA) to the newly created SLA.
d) Save

Stop Assignment of Tickets to Certain Authorized Users

Keep Tickets from being assigned to certain Authorized Users such as the BillMax system user, root, etc.

1. Login to the Staff Portal.
2. Select System Administration > Authorized Users from the Main Menu.
3. Select each Authorized User:
   a) Select No for May be a Ticket Assignee.
   b) Click Save.
Part VI

Salesforce Integration

Topics:

• Concepts
• How To
• Reference
Chapter 25

Concepts

Topics:

- Integration Overview
SalesForce® is a Customer-Relationship Management (CRM) tool. Integration with BillMax is limited to the exchange of data between the systems. Salesforce Accounts are linked to BillMax Accounts and BillMax Users are linked with Salesforce Contacts. Data entry and edits may be done from either system and the other system's data will change accordingly. SalesForce® allows Accounts and Contacts to be deleted. If deletions occur no action is taken in BillMax.

The synchronization of data is implemented by two separate mechanisms. Data created or modified in SalesForce® is pushed to BillMax via Salesforce Outbound Message API and the BillMax Soap/XML endpoint(URL). Data created or modified in BillMax is pushed to SalesForce® via Account and User hooks.

There is a Salesforce List in BillMax that changes the behavior of the data synchronization based on the parameters.

- **Logfile** - If set to '1', salesforce.log is always written. If '0', only errors are written to the file.
- **Virtual Company** - Number of Virtual Company to use when creating Accounts in BillMax. Default is the value from a Remote Applications on page 103 entry.
- **Profile** - Number of Account Profile to use when creating Accounts in BillMax. Default is the value from Remote Application entry.

**Related information**

https://salesforce.com
Chapter 26

How To

Topics:

- Update Salesforce Schema
- Add Database Trigger to Populate Account.Name
- Setup Outbound Messages
- Setup Workforce Actions and Workflow Rules
- Edit BillMax Customer Portal Configuration File
- Create Salesforce CRM User
- Configure BillMax Hooks
Update Salesforce Schema

1. Open Setup>Object Manager from the upper menu options.
2. Click on the Account.
3. Click on the Fields & Relationships.
4. Use the New button to create

<table>
<thead>
<tr>
<th>Type</th>
<th>Label</th>
<th>Length</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Company</td>
<td>50</td>
<td>BillingCompany</td>
</tr>
<tr>
<td>Text</td>
<td>Email Address</td>
<td>100</td>
<td>BillingEmail</td>
</tr>
<tr>
<td>Text</td>
<td>First Name</td>
<td>30</td>
<td>BillingFirstName</td>
</tr>
<tr>
<td>Text</td>
<td>Last Name</td>
<td>30</td>
<td>BillingLastName</td>
</tr>
</tbody>
</table>

The "_c" will be added to the Field Name to show that it is a customized field and will be used for API Name.

Add Database Trigger to Populate Account.Name

1. In Salesforce launch the Developer Console (in the Setup symbol in the top right).
2. Select File>New>Apex Trigger.
3. Enter Name as AccountName.
4. Enter sObject as Account
5. Enter as the trigger

```apex
trigger AccountName on Account (before insert, before update) {
    for (Account a: Trigger.new){
        a.Name = a.BillingFirstName__c+' '+a.BillingLastName__c;
    }
}
```

6. Launch the Developer Console (in the Setup symbol in the top right).
8. Enter Name as BillMaxAccount.
9. Enter for the page

```apex
<apex:page standardController="Account"
    action="[!URLFOR($Action.Account.New,null,[acc2=\'<ignore>'],true)]">
</apex:page>
```

10. Click Setup>Object Manager.
11. Select Account.
12. Select Buttons, Links, and Actions.
14. Click on the radio button for Visualforce page and Save.
15. Select the custom page BillMaxAccount.
16. Click SAVE.
Setup Outbound Messages

1. In Salesforce select **Setup>Process Automation>Workflow Actions>Outbound Messages** and click **New Outbound Message**.
2. Choose **Account** as the Object.
3. Give the Outbound Message the name **BillMaxAccount**.
4. **CAUTION:**

   Salesforce is very picky about certificates and URLs. They must match and the certificate needs to be signed by a known signing authority. If a change in Salesforce doesn't get pushed to BillMax, check the Outbound Message Status at **Setup>Environments>Monitoring>Outbound Messages**.

   For Endpoint URL, enter: https://<your-publicly-accessible-site>/<path-to-portal>/salesforce/account.

5. Select the Salesforce Account fields specified in **Salesforce Field Mappings** on page 234 for Accounts and click **Save**.
6. In Salesforce select **Setup>Process Automation>Workflow Actions>Outbound Messages** and click **New Outbound Message**.
7. Choose **Contact** as the Object.
8. Give the Outbound Message the name **BillMaxContact**.
9. **CAUTION:**

   Salesforce is very picky about certificates and URLs. They must match and the certificate needs to be signed by a known signing authority. If a change in Salesforce doesn't get pushed to BillMax, check the Outbound Message Status at **Setup>Environments>Monitoring>Outbound Messages**.

   For Endpoint URL, enter: https://<your-publicly-accessible-site>/<path-to-portal>/salesforce/contact.

10. Select the Salesforce Contact fields specified in **Salesforce Field Mappings** on page 234 for Contacts and click **Save**.

Setup **Workforce Actions and Workflow Rules**

1. In Salesforce select **Setup>Process Automation>Workforce Rules** and click **New Rule**.
2. Choose **Account** as the Object, and name the rule **BillMaxAccount**
3. Click **created and every time it's edited**.
4. Select **formula evaluates to true** and in the formula enter: 1=1. Click **Save & Next**.
5. On resulting page, click **Add Workflow Action** and choose **Select Existing Action**.
6. On resulting page, set Search to **Outbound Message** and select **BillMax Account Action** action.
7. Repeat steps 1-6 for **BillMaxContact**.
8. In Salesforce select **Setup>Process Automation>Workflow Rules** and click **New Rule**.
9. Click **Activate** for both new rules (one for Account and one for Contact).

Edit BillMax Customer Portal Configuration File

1. In Salesforce select **Setup>Company Settings>Company Information** and record the value for **Salesforce.com Organization ID**.
2. In BillMax edit the Customer Portal configuration file located on your public web server at <portal-root>/config/edge.php.
3. Set the value for **saileforce-orgid**.
Create Salesforce CRM User

This user will be used by the API calls. This user should have administrative rights and must have a profile where Send Outbound Messages is turned off.

1. In Salesforce select Setup>Users>Profiles and clone a profile with administrative rights. Name the profile Sys Admin - No outbound
2. Edit the profile and de-select Send Outbound Messages
3. In Salesforce select Setup>>Users and click New User.
   This user will be known as the "hook user". You will need to provide a valid email address where you have access.
4. After the hook user is created, check you email from Salesforce to finish the registration process.
   Retain the password for later use.
5. While logged in as the hook user, click View Profile icon in the top corner and then click Settings.
6. Click Reset My Security Token and complete the instructions.
   Retain the security token for later use.
8. Enter "BillMax for Connected App Name and for API Name. Enter Contact Email. Click Enable Oauth Settings.
9. Enter Callbak URL as https://<your-salesforce-site>/services/oauth2/token and select "Full Access" for Selected OAuth Scopes and click Save.
   Retain the client_id(Consumer Key) and client_secret(Consumer Secret) value for later use.

Configure BillMax Hooks

1. Edit the /usr/local/billmax/local/saleforce.pl script.
2. Note: The @ character in the email address must be preceded by "\".
   Locate the AUTHENICATIONS SETTINGS section and set the values of the parameters that have been retained from Create Salesforce CRM User on page 232.
3. Locate the ENDPOINT section and set the value for APIURL to your Salesforce site URL without any path. Save the salesforce.pl script.
4. As the billmax user, run these commands:

   1. cd /usr/local/billmax/local
   2. ln -s salesforce.pl account_hook
   3. ln -s salesforce.pl user_hook

   It is recommended that these hooks be configured to run asynchronously.
Chapter 27

Reference

Topics:

• Salesforce Field Mappings
• Salesforce Synchronization Log
• Files and Error Messages
### Salesforce Field Mappings

#### Table 8: Accounts

<table>
<thead>
<tr>
<th>BillMax Field</th>
<th>Salesforce Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>importid (Primary Key)</td>
<td>Id</td>
</tr>
<tr>
<td>cfname</td>
<td>BillingFirstName_c</td>
</tr>
<tr>
<td>clname</td>
<td>BillingLastName_c</td>
</tr>
<tr>
<td>company</td>
<td>BillingCompany_c</td>
</tr>
<tr>
<td>addr1</td>
<td>BillingStreet</td>
</tr>
<tr>
<td>city</td>
<td>BillingCity</td>
</tr>
<tr>
<td>statename</td>
<td>BillingState</td>
</tr>
<tr>
<td>email</td>
<td>BillingEmail_c</td>
</tr>
<tr>
<td>zip</td>
<td>BillingPostalCode</td>
</tr>
<tr>
<td>country</td>
<td>BillingCountry</td>
</tr>
<tr>
<td>cphone</td>
<td>Phone</td>
</tr>
<tr>
<td>fax</td>
<td>Fax</td>
</tr>
</tbody>
</table>

#### Table 9: Contacts

<table>
<thead>
<tr>
<th>BillMax Field</th>
<th>Salesforce Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>importid (Primary Key)</td>
<td>Id</td>
</tr>
<tr>
<td>fname</td>
<td>FirstName</td>
</tr>
<tr>
<td>lname</td>
<td>LastName</td>
</tr>
<tr>
<td>addr1</td>
<td>OtherStreet</td>
</tr>
<tr>
<td>city</td>
<td>OtherCity</td>
</tr>
<tr>
<td>statename</td>
<td>OtherState</td>
</tr>
<tr>
<td>email</td>
<td>Email</td>
</tr>
<tr>
<td>zip</td>
<td>OtherPostalCode</td>
</tr>
<tr>
<td>country</td>
<td>OtherCountry</td>
</tr>
<tr>
<td>dataphone</td>
<td>OtherPhone</td>
</tr>
<tr>
<td>nphone</td>
<td>HomePhone</td>
</tr>
<tr>
<td>aphone</td>
<td>MobilePhone</td>
</tr>
<tr>
<td>account (Primary Key)</td>
<td>AccountId</td>
</tr>
</tbody>
</table>

Fields ending in "_c" are custom fields. They must be added to the standard Salesforce Account object.

The `importid` and `Id` mapping is maintained by the integration and is required for processing without errors.
Salesforce Synchronization Log Files and Error Messages

The following is a list of places to check for errors.

Salesforce to BillMax

Outbound Messages

Check the Outbound Message Status in Setup>Environments>Monitoring>Outbound Messages. It will tell you if the "push" failed. If you see no records either the attempt was not made or it was successful.

/usr/local/billmax/logs/salesforce.log
In the BillMax list Salesforce, if the parameter Logfile is set to "1", this log file is always written. If "0" error messages will be written to this file

<portal-root>/storage/logs/laravel.log
This file is on the public Web server or if testing locally it is on the BillMax server.

/etc/php.ini
PHP log file

Possible Error Messages

Data Error
SOAP response was a

DNS Error
The domain name "xxx" was not in the supported set of domain names for the certificate.

BillMax to Salesforce

Errors are typically presented to the CSR if real time provisioning is being used. Details on the error can be found in:

/usr/local/billmax/logs/account_hook.log
This file details the errors pushing Account information to Salesforce Accounts.

/usr/local/billmax/logs/user_hook.log
This file details the errors pushing User information to Salesforce Contacts.
Part VII

Service Availability Determination

Topics:

• Fundamentals
Chapter 28

Fundamentals

Topics:

- Concepts
- How To
Concepts

Overview
Service Availability Determination is the automated process by which BillMax determines whether service may be provided at a location as determined by address information in a User record.


BillMax also supports creating in-house solutions and integration with other providers. An example of how to do this is provided in /usr/local/billmax/src/lib/svcavailexample.

Service Availability Sources
Tests for determining Service Availability are controlled by the BillMax List svcavailsources. The Item value in the List Item references a library in /usr/local/billmax/lib.

If Item is svcavailexample, then the library /usr/local/billmax/lib/libsvcavailexample.so will be used.

The List Item must also be marked Available. Sources marked a Discontinued will not be consulted.

When Service Availability is determined using a Service Availability Source library, the User setting Service Available? is set to Yes or No. If Yes, the process that determines Service Availability will set the User setting Technology to the appropriate setting. The User Technology setting combined with the Service Definition Technology may limit the products available to a specific User.

In addition, library may also create other objects such as Tickets.

Service Availability No Sources
If there are no Service Availability Sources, there is a way to trigger default settings for new Accounts. This includes the automatic creation of an installation Ticket.

Configuring the List ServiceAvailability and selecting Check Service Availability for the Account Technology setting will cause the following when adding a new Account via the Staff Portal:

• The Technology for the User will be set to Technology in the List.
• A Ticket will be created using the Queue and Issue in the List.

In addition, if an Account is created using the Customer Portal prequalify page, then the VirtualCompany and Profile settings will be used if specified.

TowerCoverage Concepts

Integration Overview
TowerCoverage® is a service by which a potential customer may view a Wireless/Fiber Service Provider's coverages/service areas, select their location, and request availability. BillMax is integrated with the TowerCoverage End User Submission (EUS) service as well as its Pre qualification service. See http://www.towercoverage.com.

There are multiple ways to access the TowerCoverage integration. One is through the Staff Portal, Others are initiated by the customer using the BillMax Customer Portal prequalify REST endpoint and the TowerCoverage supplied iframe.

Staff Portal/Customer Portal Interaction
Whether initiated by creating a new Account in the Staff Portal or through the Customer Portal, the basic process is the same.

1. General information is submitted including the address for service.
2. If through the Customer portal, a map of the location is displayed giving the customer the option to adjust the location.

3. An Account, User and Ticket are created. A Service Availability request is made to TowerCoverage. If BillMax has been configured to know a positive TowerCoverage Pre qualification response always indicates a certain service, BillMax returns immediately after receiving the Pre qualification response. Otherwise there may be a delay up to 30 seconds while the system wait for a TowerCoverage EUS response which will indicate fiber, wireless, or both.

4. Data resulting from the EUS request such as link analysis and fiber availability is entered into BillMax and associated with the Ticket as a message. The following steps are Customer Portal only.

5. If the customer cannot get service, then a message is displayed and the customer interaction is complete.

6. If the customer can get service, then additional information may be gathered including a user-name/password for logging the customer into the Customer Portal.

7. The customer may then be logged into the Customer Portal and placed on the Shop page where products listed are filtered by the technology available, fiber or wireless. In addition, filtering may be done by Account Class.

**iframe Interaction**

An alternative interaction with TowerCoverage and BillMax is with an iframe supplied by TowerCoverage. During the iframe interaction, the following happens:

1. Customer submits general information including address.

2. A map of the location is displayed giving the customer the option to adjust the location.

3. An EUS request is made and a message displayed to the customer. The customer interaction is complete.

4. Data resulting from the EUS request such as link analysis and fiber availability is entered into BillMax creating an Account, User and Ticket.

The created Account and User hold the prospective customer's contact information and location information. The created Ticket holds link and fiber analysis details in Messages.

**Processing "How Did You Hear About Us?"**

The following processing happens for the How Did You Hear About Us? data from TowerCoverage.

1. If the BillMax List `TCHowDidYouHear2BXAdcode` exists and an entry mapping the TowerCoverage value to an entry in the BillMax List `adcodes` by Item exists, the Account adcode will be set to the value from the adcodes List.

2. If the mapping does not exist, the TowerCoverage value will be added to the BillMax adcodes List and the Account adcode will be set to the value of the new entry.

**Account Creation**

When an Account is created from a TowerCoverage EUS submission, the following happens:

- The Virtual Company is set to the TowerCoverage List entry Company if it exists and is not zero. Otherwise it is set to the Company setting of the Remote Application.
- The Account Profile is set to the TowerCoverage List entry Profile if it exists and is not zero. Otherwise it is set to the Customer Profile setting of the Remote Application.
- First Name, Last Name, Address, City, State, Zip/Postal Code, Country, Phone and Email come from TowerCoverage data.
- The Adcode comes from TowerCoverage data but may be mapped to a BillMax value from the adcodes List. See Processing "How Did You Hear About Us?" on page 241
- Taxable is set to Yes.
- Method is set to Cash/Check.
- Default Tax Region is set to the Default Tax Region of the Account Profile.
- If a comment is supplied, it is added as an Account Note.
Logging and Error Handling

Logging may be enabled by setting the TowerCoverage List entry `LogFile` to "1". Logging information is written to `/usr/local/billmax/logs/towercoverage.log`.

When logging is enabled, the XML data is always written to the log file. If successful, the numbers of the Account, User and Ticket are written to the file. If unsuccessful, the error message is written to the file and an email is sent to the Virtual Company Administrative contact.

If logging is not enabled, successful attempts are not logged. However, error handling occurs in the same manner as if logging was enabled.

Message Creation

Messages are created only if link analyses are available. A Message is created for each link analysis. The text of the message contains the following:

- Test about link strength and fiber availability.
- `CustomerDetailsID`
- `Sitename`
- `Azimuth_ST`
- `Azimuth_ET`
- `Tilt_ST`
- `Tilt_ET`
- `Distance`
- `Received Signal`
- Coverage graphics associated with the link analysis are embedded as images in the Message.

Requirements

- A BillMax Customer Portal license. Contact sales@billmax.com for details.
- A Premium TowerCoverage subscription.

Ticket Creation

When Tickets are created from a TowerCoverage EUS submission, the following happens:

- **Phone, Email, Contact Method, Contact Time** come from TowerCoverage data.
- If link analysis data is available and the signal meets or exceeds the TowerCoverage List entry `SignalThreshold`:
  1. The Queue is set to the TowerCoverage List entry `Queue` if it exists and is not zero. Otherwise it is set to the Queue numbered "1". If the Queue specified does not exist, it is an error.
  2. The Issue is set to the TowerCoverage List entry `Issue` if it exists and is not zero. Otherwise it is not set. If the Issue is specified in the List, it must be an Issue from the Queue otherwise it is an error.
- If link analysis data is not available or the signal does not meet or exceeds the TowerCoverage List entry `SignalThreshold`:
  1. The Queue is set to the TowerCoverage List entry `QueueNoCoverage` if it exists and is not zero. Otherwise the List entry `Queue` is used.
  2. The Issue is set to the TowerCoverage List entry `IssueNoCoverage` if it exists and is not zero. Otherwise List entry `Issue` is used.
- If fiber is available:
  1. The Queue is set to the TowerCoverage List entry `QueueFiber` if it exists and is not zero. Otherwise the List entry `Queue` is used.
  2. The Issue is set to the TowerCoverage List entry `IssueFiber` if it exists and is not zero. Otherwise List entry `Issue` is used.
- Additional data may be captured. If it is, it will be stored as a Ticket Note. This is the same data as the User Location Description.
• If SLAs are in effect (see *Service Level Agreements (SLAs)* on page 215), SLA related fields will be populated.

**User Creation**

When a User is created from a TowerCoverage EUS submission, the following happens:

• *First Name, Last Name, Address, City, State, Zip/Postal Code, Country, Phone, Email, Latitude and Longitude* come from TowerCoverage data.
• If a U.S. address, census data will be populated.
• Additional data may be captured. If it is, it will be stored as the User *Location Description*.
• If wireless is available, User *Technology* is set to wireless. User *Technology* is used by the Customer Portal to determine what products are available to the customer.
• If fiber is available, User *Technology* is set to fiber. This supersedes wireless if both fiber and wireless are available.

**How To**

**Configure Service Availability**

1. If the Customer Portal REST endpoint *prequalify* is to be used, set up a Remote Application for the Customer Portal. See *Setup the Customer Portal* on page 188.
2. If, when creating an Account using the Staff Portal, it is desired Service Availability Determination be done by default, set the Account Profile setting *Check Service Availability* to *Yes*.
3. At least one Queue must be created. See *Ticketing* on page 211.
4. Edit the BillMax *ServiceAvailability* List and specify the numeric values for:

   **VirtualCompany**
   
   The Virtual Company to which new Accounts will be added. If not specified, the Virtual Company of the Remote Application for Service Availability will be used.

   **Profile**
   
   The Account Profile which will be assigned to new Accounts. If not specified, the Account Profile of the Remote Application for Service Availability will be used.

   **Queue**
   
   If no Service Availability Sources are defined, the number of the Queue to which a new installation Ticket will be assigned.

   **Issue**
   
   If no Service Availability Sources are defined, the number of the Issue of a new installation Ticket. If not specified, no Issue will be set.

   **Technology**
   
   If no Service Availability Sources are defined, the number from the List *infrastructuretechs* for the User Technology setting.

5. Edit the file `/usr/local/billmax/portal.v2/config/edge.php` and set value for the following:

   **SA_previewservices**
   
   Names and prices of Service Definitions with a both *Sell Online* and *Preview Online* will be shown if the customer can get service.
<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA_additionaldata</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>Additional form fields to capture data will be displayed if the customer can get service. Data that has the attribute <code>data-note</code> will be captured as a Ticket Note, User Location Note and Account Note. To modify what data is captured, edit the blade <code>/usr/local/billmax/portal.v2/resources/views/BillMax/Portal/TC.blade.php#coverage</code>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA_landing</td>
<td>The Customer Portal REST endpoint after a successful registration. Typically this will be <code>shop</code>.</td>
</tr>
</tbody>
</table>

## TowerCoverage How To

### Configure EUS

1. Login to TowerCoverage.
2. Click Account.
3. Click API.
4. Fill in the following:

   - **API Key**, **Username** and **Password** will be used
   - **Push Url** is the URL for the receiver on the public web server that hosts the Customer Portal software, i.e. `https://mywebserver/TCEUSResponseReceive`. 

Configure BillMax for TowerCoverage

1. Edit the entries in the BillMax List *TowerCoverage*.

   **mcid**
   
   A value from the TowerCoverage web site that is the multi-coverage map to be used for EUS submissions. This is the *mcid* value if viewing data from the *iframe* link generated by Tower Coverage.

   **account**
   
   This is the *ACCT ID* from Tower Coverage on the Account->API display.

   **key**
   
   This is the *ACCT KEY* from Tower Coverage on the Account->API display.

   **apikey**
   
   The *API Key* from *Configure EUS* on page 244.

   **apiusername**
   
   The *Username* from *Configure EUS* on page 244.

   **apipassword**
   
   The *Password* from *Configure EUS* on page 244.

   **testmode**
   
   See *Testing TowerCoverage Setup* on page 245.

   **Queue**
   
   The Queue to which the User Ticket for a new Installation will be assigned. If not specified, "1" will be used.

   **Issue**
   
   The default Issue for the new Ticket. Specified using the Issue entry. If not specified, no Issue will be selected.

   **QueueFiber**
   
   The Queue to which the User Ticket for a new fiber Installation will be assigned. If not specified, "Queue" will be used.

   **IssueFiber**
   
   The default Issue for the new fiber Ticket. If Issue is non-zero and *QueueFiber* is non-zero, this must be non-zero.

   **QueueNoCoverage**
   
   The Queue to which the User Ticket with no service availability will be assigned. If not specified, the value for *Queue* will be used.

   **IssueNoCoverage**
   
   The default Issue for the new Ticket with no coverage. If Issue is non-zero and *QueueNoCoverage* is non-zero, this must be non-zero.

2. Edit the List *svcavailsources* and add the List Item *towercoverage*. If you ever want to disable TowerCoverage, set the List Item *Availability* to *Discontinued*.

Testing TowerCoverage Setup

How to test local setup for Queues, Tickets, List settings, shopping etc.

The Customer Portal must be configured for TowerCoverage and *edge* is running. Local server CGI programs and scripts are used to mimic TowerCoverage responses.

1. **CAUTION:** When done, set the value back to *false*. 
Edit the List *TowerCoverage*. Set the value for *testmode* to *yes*. Then either set the values for *apikey*, *apiusername* and *apipassword* to *test123*, *myuser* and *mypassword* respectively, or edit `/usr/local/billmax/bin/TCEUSResponse` to match the values that are already set in the List for these List Items.

2. Access [https://yourbillmaxserver/testportal/prequalify](https://yourbillmaxserver/testportal/prequalify)

3. Enter data. The following in Phone field triggers the following behavior:

<table>
<thead>
<tr>
<th>Phone</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>Pre qualification returns no coverage.</td>
</tr>
<tr>
<td>F</td>
<td>Pre qualification returns fiber available only.</td>
</tr>
<tr>
<td>W</td>
<td>Pre qualification returns wireless coverage available only.</td>
</tr>
<tr>
<td>FW</td>
<td>Pre qualification returns both fiber available and wire coverage.</td>
</tr>
</tbody>
</table>

### Deployment

1. Verify the following in `/usr/local/billmax/portal.v2/config/edge.php`

   The following settings comes from account information at the *TowerCoverage* website.

   **TCmulticoverageid**
   
   The multi coverage map id.

   **TCaccount**
   
   *Acct ID* found on API page.

   **TCkey**
   
   *Acct KEY* found on API page.

   **TCAPIKey**
   
   *API Key* found on API page.

   **TCEUSResponseUsername**
   
   *Username* found on API page.

   **TCEUSResponsePassword**
   
   *Password* found on API page.

Make sure the following are set

**TCLanding**

*The Customer Portal landing page when the user logs in after success in determining service availability. Set to *shop*.**
2. On the public web server deploy the Customer Portal if not already done. If already done, make sure `edge.php` is updated with the correct settings.

Part VIII

VOIP/Telecommunication

Topics:

- Concepts
- How To
- Reference
Chapter 29

Concepts

Topics:

• CDR Billing
• Taxes, Fees and Universal Service Funds (USF)
CDR Billing

BillMax VOIP/Telecommunication Overview

BillMax supports billing for VOIP/Telecommunication services that range from very simple billing practices to complex scenarios. Examples of simple to complex are:

1. Billing one price the service, usage included.
2. Billing one price the service with additional billing for usage.
3. Billing for both outgoing and incoming calls.
4. Billing for specialty numbers such as incoming Toll Free numbers and premium (900,976) numbers.
5. Billing for combined usage from multiple lines for corporate and family plans.

BillMax also supports managing Universal Service Fund (USF) assessments, both the Federal and State portions. This includes support for reporting to the managing entities such as the Universal Service Administrative Co. (USAC) and passing assessments through to the end customer.

Activating the CDR Billing module in BillMax requires a license with CDR Billing activated. Very simple service only billing for VOIP/Telecommunication may be done without the CDR Billing module activated.

Call Detail Record (CDR) Association

CDRs are processed and associated with BillMax Packages and Services for billing purposes. The association is done on both incoming and outgoing calls and the CDR data is associated by comparing the CDR data to the Package or Service d01 field.

Each CDR is associated with up to two Packages or Services based on the specified association criteria specified for inbound and outbound attributes of the CDR. Typical association is made by comparing the source of the call or the destination of the call to the Package or Service d01 field. Alternatively, a secondary set of criteria may be used such as source channel or destination channel may be compared to the d01 field. If both a primary and a secondary set of criteria are specified, both are attempted to make an association to the correct Package or Service.

**Attention:** Only Packages or Services associated with a Service/Package Definition with a CDR Billing Plan plan will be considered for CDR association.

CDR Rating

CDR rating is the classification of CDRs and the assignment of one or more billing amounts to a CDR. BillMax supports classification of CDRs via NPANXX tables and rated via per minute amounts. In addition, BillMax supports both pre classified (classification for USF purposes) and pre rated (rated for long distance amounts) CDRs.

BillMax separates CDRs into outbound and inbound CDRs. A CDR is considered outbound when the source number or channel matches the Package or Service d01 field. A CDR is considered inbound when the destination number or channel matches the Package or Service d01 field. Note that from the BillMax perspective, a CDR may be both outbound and inbound.

BillMax classifies outbound CDRs as one of four types of calls:

- **Plan Calls** - "local" calls for which the sum of the minutes of the call is included for price for a specified duration. A typical plan is so many minutes per month. If the caller exceeds the specified Plan minutes, more billings may occur.
- **Long Distance Calls** - Any call that is not a Plan, N11 or Toll Free call.
- **N11 Calls** - (211, 311, ...) - N11 calls designated as such during CDR preprocessing. Allows for non North America users of BillMax.
- **Toll Free Calls** - In North America, number where the area code begins with 800, 844, 855, 866, 877 and 888 - Toll Free calls designated as such during CDR preprocessing. Allows for non North American users of BillMax. Any billings associated with Toll Free calls such as connect fees are classified as interstate for USF purposes. Premium calls (900,976) are classified as intrastate.
In addition, in the United States both Plan calls and Long Distance calls may have an additional classification of International, Interstate and Intrastate. This may be used to apply a Federal Universal Service Fund charge to the end customer.

**Rate Decks**

Rate Decks are used to determine whether an outbound CDR is a Plan Call or a Long Distance call and what per minute billing should occur. If selected, BillMax will use data from [http://www.localcallingguide.com/](http://www.localcallingguide.com/) to determine the local calling area for a particular number. If the called number falls in the local calling area, then the CDR is considered a Plan Call. If data from [http://www.localcallingguide.com/](http://www.localcallingguide.com/) is not used or the call number is determined to be a Long Distance call, BillMax may consult a Rate Deck if specified. If the billing for the called number in the second Rate Deck is 0, then the call is considered a Plan Call, otherwise it is considered a Long Distance call and is subject to the billing in the Rate Deck and the Long Distance billing parameters.

**Attention:** If data from [http://www.localcallingguide.com/](http://www.localcallingguide.com/) is not used and no Rate Deck is specified, all calls are considered to be N11, Toll Free and Plan calls. None are considered Long Distance.

**Attention:** If data from [http://www.localcallingguide.com/](http://www.localcallingguide.com/) is used the primary source search key is used for lookup and must be a 10 digit phone number. This does not imply that the 10 digit number be the d01 field in the BillMax database. As an example, if the primary source search key is 8174467776, the secondary source search key is BillMax and the d01 value in the BillMax database is BillMax, then the CDR will have an association base on the BillMax value but the look up in the data from [http://www.localcallingguide.com/](http://www.localcallingguide.com/) will be done using 8174467776.

The Rate Deck specification for BillMax may be found here *BillMax CDR Rate Deck* on page 266.

BillMax supports the following billing models:

**Plan Calls**

BillMax supports the following billing parameters for Plan Calls:

- A limit on the number of minutes before extra billing occurs or an unlimited plan.
- Optional inclusion of minutes from inbound calls in the total minutes for billing/display.
- An optional connect fee for every completed outbound call

**Long Distance Calls**

BillMax supports the following billing parameters for Long Distance Calls:

- A default rate for calls not found in Rate Decks.
- An option to include both outbound and inbound minutes in the Plan minutes calculations.
- An optional connect fee for every completed outbound call.

**N11 calls**

BillMax supports the following billing parameters for N11 calls:

- Selection for each N11 number where it will be included in Plan minutes calculations.
- A default rate.
- A default connect fee.
- Selection for each N11 number as to whether it will be billed the fee or rate. For example, no billing for 911 calls may be desired.

**Inbound Toll Free and Premium call**

BillMax supports the following billing parameters for Toll Free/Premium calls using Plan parameters:

- Billing for input minutes.
- Billing using other Plan parameters.
CDR Preprocessing

CDRs may be generated from a multitude of sources resulting in a variety of formats. As such CDRs are preprocessed and loaded into the cdr table before being associated to a Package/Service and before being rated. Making preprocessing easily modifiable supports billing outside the NANPA domain.

Preprocessing is done via a program or scripts that output CDRs in a format expected by BillMax. Preprocessing scripts are located in /usr/local/billmax/local and the file name is pre-pended with string preprocesscdr_. The following CDR formats are supported by default:

- Asterisk®
- PBXWare®
- Frontier®
- NetSapiens® Advanced
- Voyant® EMI CDRs.

These formats are itemized in the List cdrsources.

You add processing of a CDR format by adding a CDR Source and specify the CDR format. The CDR Source also specifies how CDRs are to be classified:

- **No**
  - CDRs are not classified with respect to USF and will not be.

- **Use NPANXX Table**
  - Classify the CDRs for USF purposes using the NPANXX tables.

- **Pre Classified**
  - The CDRs are pre classified for USF purposes.

The CDR Source also specifies how CDRs are to be rated:

- **No**
  - CDRs are not to be rated. By definition this means CDRs will be associated but not billed in any way. Use this if you want to capture CDRs for informational purposes such as traffic studies.

- **Yes**
  - Rate the CDRs according to the CDR Billing Plan.

- **Pre Rated**
  - CDRs are pre rated and the amounts are submitted with the CDRs.

The CDR preprocessing script is responsible for the following:

1. Eliminating calls that should not be considered such as those never connected.
2. Removing non-numeric characters from the source and destination numbers.
3. Eliminating calls from consideration based on length of number dialed or other considerations such as number being a long internal extension.
4. Flagging N11, international call, Operator assisted calls, Toll Free calls and Premium calls as such. In addition, if needed, flagging calls as Local calls.
5. Outputting the necessary source, destination and Rate Deck keys to match Package/Service d01 values and values in the Rate Decks.
6. Depending on the source of the CDRs, the CDRs may be tagged as inbound, outbound or indeterminate. If indeterminate and the source is in BillMax it is considered outbound. If indeterminate and the destination is in BillMax, it is considered inbound. If indeterminate and both the source and destination numbers are in BillMax, it is considered both inbound and outbound and a additional CDR for the inbound portion is created.

The output format from CDR preprocessing may be found here BillMax CDR Input Format on page 266.
CDR Classification - NANP

BillMax optionally implements classifying calls according to the North America Numbering Plan (NANP). Data for classification are stored in the BillMax npanxx table. This table contains the source and destination NPANXX values and various other data used to classify data in a specific CDR.

Data for the npanxx table may inserted from various sources. Out of the box, BillMax is integrated with Local Calling Guide [https://localcallingguide.com](https://localcallingguide.com). BillMax consults with the Local Calling Guide as needed with an automatic refresh every 30 days. However, other sources may be integrated. An example library /usr/local/billmax/src/lib/bxtnpanxx is supplied as an example.

Using the Number Plan Area (NPA) data and the Central Office exchange code (NXX), data from a CDR may be classified in the following manner:

1. Local
   a. Optionally Interstate
   b. Optionally International
2. Long Distance
   a. Optionally Interstate
   b. Optionally International

If a rate deck is used and the rate for a Long Distance is 0.00, then the Long Distance CDR will be treated as a Plan CDR for billing purposes.

CDRs and the Federal Universal Service Fund

In many cases BillMax customers want to pass through to the end customer the amount needed to pay into the Federal Universal Service Fund (FUSF). The Federal USF is based on end user revenues for interstate and international calls. The contribution rate for FUSF is set by quarter by the FCC.

BillMax allows three methods to calculate and report this information on the FCC 499 report: Safe Harbor, Traffic Study, and actual billing minutes.

BillMax customers may pay into the FUSF a Safe Harbor amount and bill the end customer their share of the Safe Harbor amount. Different service types have different Safe Harbor percentages from the FCC.

A second methodology is a traffic study. A quarterly traffic study on your network that you document with the FCC/USAC may lend itself to a lower amount needed to pay in. Currently, BillMax does not have a way to conduct a traffic study, but will allow the interstate and international percentages discovered from a traffic study to be entered.

The third method in BillMax is to add a FUSF charge to a customer's bill based the amount billed using CDR data. If a customer has long distance CDRs that are interstate or international, the billing displayed to the customer is separated from the billing for other long distance calls. In addition, the FUSF charge based on the interstate/international calls is added to the customer's bill.

Billing for Plan interstate/international calls is slightly different as typically a customer has a set amount of free minutes and only the overage is billed. If this is the case, the percentage of overage billing considered as interstate/international is the overall percentage of interstate/international calls for all Plan calls.

<table>
<thead>
<tr>
<th>Plan FUSF example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Assume an FUSF assessment of 20%.</td>
</tr>
<tr>
<td>2. A plan has 1000 Plan minutes free with 10 cents a minute overage.</td>
</tr>
<tr>
<td>3. A customer uses 2000 Plan minutes resulting in an overage billing of 1000 minutes, or $100.00.</td>
</tr>
<tr>
<td>4. Based on CDR data, of the Plan minutes 65% were not interstate/international and 35% were interstate/international.</td>
</tr>
<tr>
<td>5. The customer would see 3 charges on their bill:</td>
</tr>
<tr>
<td>a. The $65 for overage Plan minutes, not interstate/international.</td>
</tr>
<tr>
<td>b. The $35 for overage Plan minutes that were interstate/international.</td>
</tr>
</tbody>
</table>
c. The FUSF charge of $7.00 (.2 * $35).

**CDRs and the State Universal Service Fund**

In many cases BillMax customers want to pass through to the end customer the amount need to pay into the State Universal Service Fund (SUSF) in addition to the Federal USF. The State USF is based on end user revenues for intrastate calls.

**Note:** If the SUSF is to be passed through, the state/province two letter abbreviation must be used. Contact support@billmax.com to change your data or control the data through a drop down.

What is required for the SUSF varies widely by service type and state. The State USF is optional in BillMax. SUSF amounts can be found on the Sales Tax report broken out by state and shows the revenue and pass through fees that have been collected.

**CDR Billing Plans**

A CDR Billing plan contains the information necessary to bill using data from CDRs. Data from the CDRs may be used to rate Long Distance calls, sum Plan Minutes, bill for N11 calls, etc. All this is governed by the CDR Billing Plan.

A CDR Billing Plan is attached to a Package or Service Definition. By virtue of this attachment, a Package or Service is considered during CDR Association.

If a Package or Service is based off a CDR Billing Package or Service Definition, a customized CDR Billing plan may be added to the Package or Service. This allows for special CDR Billing pricing for a particular Package or Service without having to create additional Package or Service Definitions.

A CDR Billing plan may be marked as "Billing Suspended". This is useful for when first implementing a new CDR Billing plan. Automated billing is suspended.

**Timezones and Service Locations**

The timestamps in the CDRs after pre-processing may be either UTC or a local timezone. If UTC, each service location will need to be assigned a timezone. A default list of timezones is stored in the BillMax List timezones. If a new timezone is added to the list, it must be supported by the underlying operating system. If no timezone is specified, times are considered local.

A default timezone may be specified on the Virtual Company. An Account created under the Virtual Company will use the Virtual Company timezone as the default. The timezone specified on the Account is used as the default for a User under the Account.

Packages/Services added to a User are automatically assigned the User record as a location for timezone and mapping purposes. A Package/Service may be configured to have a different location than the parent User by specifying an alternate User belonging to the Account. This allows for one User to be used for tax purposes and default map location and timezone while containing Packages/Services that may be in a different location but should be taxed at the primary location.

**Taxes, Fees and Universal Service Funds (USF)**

**Fees**

Flat rate fees, such as E911, may be assessed using a Recurring Service Fee. Functionality supported is

1. No prorating
2. Fee per service
3. Marking the Fee as Government Surcharge. This is required for USF reporting.
4. Reporting is done through the Sales report. Although not technically revenue, it is treated as such.
Taxes

Assessing taxes on VOIP/Telecommunications products is done in the same manner as any other class of product. Typically a new Tax Class other than "Sales" is used. Taxes may be used to recover amounts for USF purposes even for non-taxable accounts. The calculations will include non-taxable amounts if the setting is made to pass the fee through to accounts. In addition, the Tax engine is not suitable for flat fee assessments such as E911.

The Tax Designation is set on the Account Profile and inherited by the Account. This field specifies the type of customer for tax purposes. Currently it is applicable only if using a tax service or assessing USF. Some customer types cannot be charged USF.

The Tax Region can be specified on the Account Profile and inherited by the Account as well if there is a common enough region to be a default. Otherwise, the Tax Region will be set on the Account and inherited by the User. The User's Tax Region is used in billing the Services and Packages at the User's location.

Reporting on an Accrual or Cash/Credit basis is done using the Sales Tax Report.

See Taxes on page 129.

Universal Service Fund (USF)

Universal Service Fund management is split into the following categories:

1. Reporting revenue for the Federal USF.
2. Recovering amounts contributed to the Federal USF from the end customer.
3. Reporting revenue for the State USF if applicable.
4. Recovering amounts contributed to the State USF if applicable.

Unlike taxes, reporting of USF revenues and passing through contributions to the end customer are disconnected. USF reporting may be on taxable or taxable plus non-taxable amounts whereas taxes only involve taxable amounts. In addition, contributions are based on intrastate, interstate and international portions of billings.

BillMax computes amounts for various revenue lines for the FCC 499A and 499Q reports while creating an audit trail. In addition, it will generate reports in support of State USF reporting.

Reporting USF Revenue

BillMax supports reporting USF revenue using the following categorizations:

1. Type of service based on technology. Examples are:
   a. Interconnected Voice over Internet Protocol (VOIP) Service.
   b. Wired or Wireless Local Exchange Telephone Service.
2. State in which billing occurred.
3. Interstate, intrastate, international billings. If no CDR data available, BillMax supports Safe Harbor or Traffic Studies, by state if necessary. If CDR data is available and is rated/billed, BillMax supports using Traffic Studies if needed on the recurring billing and current USF rates on the CDR billings.
4. Taxable vs Taxable plus Nontaxable billings.
5. Association with broadband or not in the case of VOIP.

Support is available for both the FCC 499Q and FCC 499A filings. Data for State USF may be derived from these reports and can be found in the Sales Tax report. In some cases separate reports may be available or created.

Recovering USF Contributions

BillMax supports recovering USF contributions in several ways:

1. Add Tax Items to a Tax Group and classifying the Tax Items as a Fees. Specify whether they are Federal USF or State USF with the state notated. The Fee can also be marked to be passed to non-taxable accounts unlike normal taxes.
2. Use the Federal USF table to specify the method that will be used for reporting: Safe Harbor, Traffic Study, or Minutes from CDRs. The method helps differentiate the amount of intrastate, interstate, and international traffic that is being billed.
Chapter 30

How To

Topics:
- CDR Billing
- Taxes, Fees and Universal Service Funds (USF)
CDR Billing

Getting Started with CDR Billing

The following assumes you have a source of CDRs in CSV format and that you know what the data in each field is. Some of the early steps

1. Create a Package or Service Definition for billing the Service portion of the VOIP/Telecommunication plan. Set the FCC classification to the appropriate value.

2. If passing through USF fees, create the FUSF and SUSF as needed. See Create a Federal Universal Service Fee on page 264 and Create a State Universal Service Fee on page 264.

3. If using a rate deck, upload the rate deck into BillMax. See BillMax CDR Rate Deck on page 266 and Upload a CDR Rate Deck on page 260.


5. Add a CDR Billing plan to a Package or Service Definition. See Add a CDR Billing Plan to a Package/Service Definition on page 261.

6. Add a new CDR Source. See Create a CDR Source on page 261. If adding a new cdrformat, insert it as a List Item.

7. If using an existing cdrformat, execute "cp preprocesscdr_cdrsource-dist preprocesscdr_cdrsource". Make sure the new file is owned by the BillMax system user and is executable.

8. If adding a new cdrformat, using the data from the CSV file, create a script in /usr/local/billmax/local that takes the file as input and outputs a CSV file in a format that BillMax expects. Output the lookup keys in the same format as what is stored in BillMax. See Call Detail Record (CDR) Association on page 252, CDR Preprocessing on page 254, Create Package/Service Association for a CDR on page 261 and BillMax CDR Input Format on page 266. There are also examples in /usr/local/billmax/local.

9. Test - the following will be iterative

   a) Upload the CDR file. See Upload CDR File on page 262.

   b) Execute Reports > CDR Keys Not Associated. If there are unassociated keys that need to be associated:

      1. Rework the new script if a new script was created. If so, delete the uploaded CDRs using Reports > CDR Files.

      2. Modify/add data to the VOIP services such that they match the data in the uploaded CDRs and execute Associate CDRs from Reports > CDR Keys Not Associated.

   c) Examine results by executing Reports > Future CDR Billings. Make changes to the CDR Billing plan if needed. From Reports > CDR Listing run ReRate Unbilled from the Context Menu.

   d) DANGER: The following will bill all usage for an Account, not just CDR usage. If the Account is billed for more types of Usage than just CDRs, open the VOIP/Telecommunication Service and on the Additional Usage Billing tab, click Bill for the CDR Billing plan.

      Select an Account and execute Other > Bill Usage. The result will be Pending Transactions. Deleting them will revert the CDRs to a not billed state.

   e) View an Invoice or Billing Statement to examine results of billing, CDR display, etc.

10. Put into production:

   a) Automate CDR uploads. See Insert CDRs in Batch Mode on page 263.

   b) Edit CDR Billing Plan setting Billing Suspended to No.

Upload a CDR Rate Deck

To make a Rate Deck available for use in CDR Rating

1. Select Billing Administration > CDR Rate Deck from the Top Menu.

2. Select New from the Context Menu.
3. Enter into **Name** a descriptive name.
4. Enter any other desired information
5. Select **Save** from the Context Menu.
6. Click **Choose File** to specify the **Rate Deck CSV File**.
7. Select **Upload Rate Deck** from the Context Menu.

**Create a CDR Billing Plan**

Needed to do any CDR usage billing and to use a CDR Billing Rate Deck.

If a CDR Rate Deck is to be used, the CDR Rate Deck should already be specified.

1. Select **Billing Administration > CDR Billing Plan** from the Top Menu.
2. Select **New** from the Context Menu.
3. Enter into **Name** a descriptive name.
4. Enter any other desired information
5. Select **Save** from the Context Menu.

**Add a CDR Billing Plan to a Package/Service Definition**

1. Select **Billing Administration > Recurring Package** or **Billing Administration > Recurring Service** from the Top Menu.
2. Select the desired Package/Service Definition.
3. Select the **Additional Usage Billing** tab.
4. Choose the **Usage Module CDR Billing**.
5. Choose the **Plan**.
6. Click **Save**.

**Create a CDR Source**

Each CDR Source will specify the format, classification mechanism, and how the CDRs are to be rated.

1. Select **Billing Administration > CDR Sources** from the Top Menu.
2. Select **New** from the Context Menu.
3. Enter the **Name** of the new CDR source.
4. Choose the **CDR Format**.
5. Enter any other desired information
6. Select **Save** from the Context Menu.

**Create Package/Service Association for a CDR**

1. **Note**: Some types of associations:
   - DID only
   - Channel only
   - Channel then DID

   Decide on what type of association will be made.

2. **Note**: The simplest convention is to store the DIDs as 10 digit numbers without any "punctuation" 8174467776 instead of (817)446-7776 as an example. The delivered CDR preprocessing script outputs dialed numbers in this format.

   **Note**: If using channels, make sure the CDR preprocessing script outputs the channel identifier in the same format as stored in the Package/Service d01 fields.
Decide on an convention for data stored in a Package/Service d01 field and what will be supplied by the CDR preprocessing step.

3. Populate the Package/Service d01 fields with the same data/search keys that will come from the CDR preprocessing step.

Upload CDR File

1. Select Billing Administration > Upload CDR Data from the Top Menu.
2. Upload the RAW CDR file
   a) Select the CDR Source.
   b) Click UPLOAD FILES.
   c) Click Choose Files and select one or more files.
   d) Click Upload Files.
      The raw CDR file is now on the BillMax server.
3. Process the CDR file:
   a) Select the Source CDR File.
   b) Click PROCESS CDR File.
      The CDR file has been run the preprocessor and the results are ready to be loaded into the BillMax database.
4. Load Processed CDR File:
   a) Select BillMax CDR File.
   b) Click LOAD CDRs.
      The CDRs are loaded into the BillMax cdr table and the CDRs are associated with Packages and Services. Run Reports > CDR Keys Not Associated to identify any CDRs that were unable to be matched.

Associate CDRs

1. Select Reports > CDR Keys Not Associated from the Top Menu.
2. Enter the Start Date and End Date to be considered.
3. Select Keys.
4. Click Run Report from the Context Menu
5. Add listed keys to the appropriate Packages or Services.
6. Click Associate CDRs from the Context Menu.
7. Repeat if needed.

View CDR Usage

1. Select Reports > CDR Billing Usage from the Top Menu.
2. Select Type
   Package/Service - Not Billed Not billed CDR record. Specify number of Package or Service for Search Value.
   Sales Transaction Billed CDR records. Specify the number of the Sales transactions for Search Value.
   Billing Statement Billed CDR records associated with a Billing Statement. Specify the Billing Statement number (1-2 for example) for Search Value.
   Invoice Billed CDR records associated with an Invoice. Specify the Invoice number (1-2 for example) for Search Value.
3. Select any other options.
4. Click **Run Report** from the Context Menu.

**Insert CDRs in Batch Mode**

There are many ways this can be done from downloading files to get the data to reading databases to get the data. Below is an example of what can be done. This assumes that the CDR file is on the BillMax server and may be renamed and a new one will be created by the data gathering process.

2. Add a step to the nightly Batch Processing
   - If desired to gather CDRs before billing, insert step before nightly processes. Decide if failure is okay or not for the CDR step.
   - Program name will be `processCDRFile`.
   - Arguments will be `-f CDRfilename`

**Override a CDR Billing Plan for a Package/Service**

1. Select the relevant Package/Service for an Account.
2. Select the **Additional Usage Billing** tab.
3. Click **Add Plan** for the CDR Billing Usage Plan.
4. Choose the **Plan**.
5. Click **Save**.

**Remove CDRs from Documents**

Removing the display of CDRs on Invoice and Billing Statements in generally controlled by settings on the CDR Billing Plan. Making the following edits will accomplish the same goal.

1. Edit `/usr/local/billmax/cfg/xsltfiles/fo/documents.xsl` to make customizations
2. Where applicable edit the following:

   **Note:** Before

   ```xml
   <xsl:if test="noteSet">
     <fo:block>
       <xsl:if test="noteSet/note">
         <fo:block>
           <xsl:value-of select="text"/>
         </fo:block>
       </xsl:if>
     </fo:block>
   </xsl:if>
   </xsl:for-each>
   </xsl:if>
   <xsl:if test="not(//phoneUsageTables)">
     <fo:block id="endofdoc"/>
   </xsl:if>
   <fo:flow/>
   </fo:page-sequence>

   **After**

   ```xml
   <xsl:if test="noteSet">
     <fo:block>
   ```
Taxes, Fees and Universal Service Funds (USF)

Create a Federal Universal Service Fee

If a FUSF is to be passed through to a customer, a Federal USF must be created and updated quarterly.

1. Select **Billing Administration > Tax/Fee Items** from the Top Menu.
2. Select **New** from the Context Menu.
3. Choose **Fee**.
4. Choose the **Type** for Federal USF.
5. Enter any other desired information.
6. Select **Save** from the Context Menu.
7. Add the Federal USF fee to the appropriate **Tax Groups**.
8. **Attention:** The FCC updates the Contribution Factor for the FUSF each quarter.

   To update the FUSF quarterly, select **Billing Administration > Federal USF**.
9. Select the **Number** from the latest Federal USF entry and click **Add Next Quarter** from the Context Menu.
10. Select **Billing Administration > Tax/Fee Items** from the Top Menu. Enter the **Start Date** and **New Rate**.

Create a State Universal Service Fee

If a SUSF is to be passed through to a customer, a State USF must be created for each state.

1. Select **Billing Administration > Tax/Fee Items** from the Top Menu.
2. Select **New** from the Context Menu.
3. Choose **Fee**.
4. Choose the **Type** for State USF.
5. Choose the **StateName**.
6. Enter any other desired information.
7. Select **Save** from the Context Menu.
8. Add the State USF fee to the appropriate **Tax Groups**.
Chapter 31

Reference

Topics:

- BillMax CDR Rate Deck
- BillMax CDR Input Format
- VOIP Glossary
BillMax CDR Rate Deck

The BillMax CDR Rate deck is a Comma Separated Value (CSV) file used to determine what calls are Plan calls and to rate Long Distance calls. The header line "Description, Destination, Rate, Minimum Duration, Increment, Connect Fee" is required.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Descriptive value of destination string.</td>
</tr>
<tr>
<td>Destination</td>
<td>A string representing the destination of the call. For numbers part of the North American Number Plan Administration (NANPA), all strings should begin with &quot;1&quot;. Examples are:</td>
</tr>
<tr>
<td></td>
<td>• 1817 - Fort Worth, Texas</td>
</tr>
<tr>
<td></td>
<td>• 1905 - Ontario, Canada</td>
</tr>
<tr>
<td></td>
<td>• 1817446 - further refinement of Fort Worth, Texas</td>
</tr>
<tr>
<td>Rate</td>
<td>Rate per minute. Specified in fractional minimum monetary units. Cents for USA. Support for up to 10 decimal places.</td>
</tr>
<tr>
<td>Minimum Duration</td>
<td>For every answered call, at least this duration in seconds will be used for billing purposes. As an example, if the Minimum Duration is 30 and a call is only 10 seconds long, a thirty second call will be billed. A value of zero will cause the use of the Long Distance Minimum Duration parameter.</td>
</tr>
<tr>
<td>Increment</td>
<td>The step increment to use in seconds when computing the duration for billing. As an example, if a call is 53 seconds and the Increment is 5, a 55 second call will be billed.</td>
</tr>
<tr>
<td>Connect Fee</td>
<td>The amount, specified in dollars and cents, that will be assessed for every completed call to this Destination.</td>
</tr>
</tbody>
</table>

BillMax CDR Input Format

The BillMax CDR file input format is a Comma Separated Value (CSV) file. If there is no header line, the order documented below is assumed. Processing of original CDRs should result in this format. Fields are optional within reason, but use a header line if not specifying every field. Processing of the header line is case insensitive.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>The original source/inbound of the call.</td>
</tr>
<tr>
<td>Destination</td>
<td>The original destination/outbound of the call.</td>
</tr>
<tr>
<td>Source Channel</td>
<td>The source/inbound channel or path of the call.</td>
</tr>
<tr>
<td>Destination Channel</td>
<td>The destination/outbound channel or path of the call.</td>
</tr>
<tr>
<td>Call Start</td>
<td>The time stamp of the start of the call - UTC</td>
</tr>
<tr>
<td>Call End</td>
<td>The time stamp of the end of the call - UTC</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Billable Seconds</td>
<td>must be non-zero.</td>
</tr>
<tr>
<td>Source Search Key 1</td>
<td>The first source/inbound value to try to match to the Package/Service d01 value.</td>
</tr>
<tr>
<td>Source Search Key 2</td>
<td>The second source/inbound value to try to match to the Package/Service d01 value.</td>
</tr>
<tr>
<td>Destination Search Key 1</td>
<td>The first destination/outbound value to try to match to the Package/Service d01 value.</td>
</tr>
<tr>
<td>Destination Search Key 2</td>
<td>The second destination/outbound value to try to match to the Package/Service d01 value.</td>
</tr>
<tr>
<td>Destination Rate Deck Key</td>
<td>The destination/outbound value to match to a value in a Rate Deck.</td>
</tr>
<tr>
<td>Flags</td>
<td>A bit mask denoting what type of call the CDR represents:</td>
</tr>
<tr>
<td></td>
<td>2  Operator assisted</td>
</tr>
<tr>
<td></td>
<td>4  International</td>
</tr>
<tr>
<td></td>
<td>8  N11 call</td>
</tr>
<tr>
<td></td>
<td>16 Toll Free</td>
</tr>
<tr>
<td></td>
<td>32 Local Call</td>
</tr>
<tr>
<td></td>
<td>1024 Premium Call</td>
</tr>
<tr>
<td>Unique ID</td>
<td>A unique id for the record. Combined with Start of Call in the BillMax database to create a unique key across sources.</td>
</tr>
<tr>
<td>Status</td>
<td>Status of call.</td>
</tr>
<tr>
<td></td>
<td>0  Call is in progress.</td>
</tr>
<tr>
<td></td>
<td>1  Call is complete.</td>
</tr>
<tr>
<td>Time Call Answered</td>
<td>The time stamp of when the call was answered - UTC</td>
</tr>
<tr>
<td>Record Number</td>
<td>Record number of CDR in current file/dataset.</td>
</tr>
<tr>
<td>Direction of Call</td>
<td>Not specified in CDR. BillMax will derive it.</td>
</tr>
<tr>
<td></td>
<td>1  Call is outbound from Customer.</td>
</tr>
<tr>
<td></td>
<td>2  Call is inbound to Customer.</td>
</tr>
</tbody>
</table>

**The following may be used for pre-rated and pre-classified CDRs.**

<p>| Rated Amount                  | Rated amount based on duration of call.                                    |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call Type</td>
<td>0  Not specified in CDR. BillMax will derive it. Typically used for location less numbers such as Toll Free and Premium numbers.</td>
</tr>
<tr>
<td></td>
<td>1  Process as a Plan call.</td>
</tr>
<tr>
<td></td>
<td>2  Process as a Long Distance call.</td>
</tr>
<tr>
<td></td>
<td>5  A local call processed as an N11 call.</td>
</tr>
<tr>
<td>Connect Fee</td>
<td>Connect Fee to bill.</td>
</tr>
<tr>
<td>Billed Seconds</td>
<td>Billed Seconds for the supplied Rated Amount.</td>
</tr>
<tr>
<td>Source Description</td>
<td>Description of the Source of the call.</td>
</tr>
<tr>
<td>Destination Description</td>
<td>Description of the Destination of the call.</td>
</tr>
<tr>
<td>USF Region</td>
<td>The region/state to which the call/revenue/minutes should be attributed for Federal USF and State USF purposes.</td>
</tr>
<tr>
<td>USF Class</td>
<td>0  Not relevant for USF purposes. In general outbound toll free, premium numbers and non billed inbound calls should have this value.</td>
</tr>
<tr>
<td></td>
<td>1  Intrastate.</td>
</tr>
<tr>
<td></td>
<td>2  Interstate.</td>
</tr>
<tr>
<td></td>
<td>3  International.</td>
</tr>
</tbody>
</table>

**VOIP Glossary**

**Call Detail Record**

See [https://en.wikipedia.org/wiki/Call_detail_record](https://en.wikipedia.org/wiki/Call_detail_record)

Call Detail Record (CDR)

**CDR**