BillMax User Manual

Contents

Notices

Documentation for BillMax.

Comments about this documentation may be sent to <*support@billmax.com*>

Preface

The following contains information pertaining to the BillMax Application of BillMax Billing Solutions. It is a work in progress and topics are driven completely by questions from existing customers.

List of Figures

List of Tables

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*
- *BillMax Hardware Requirements*
- *Install BillMax*
- *Other Administrative 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
3. Sign p. 7.
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.

BillMax Hardware Requirements

> **Note:** BillMax requires a dedicated host (server) for its use. This setup enables a secure environment from which to operate as well as greatly reduces complications related to the installation and maintenance of software. BillMax may be installed on bare-metal hardware or on virtual systems (on premise or hosted). BillMax (the company) offers hosting services using Amazon Web Services(AWS)^{SM} as well. Contact BillMax sales (sales@billmax.com) for more information.

The required capabilities of the hardware are highly dependent on the number of subscribers managed by BillMax and to a lesser extent, the number of customer service representatives using the system at any given time.

Two metrics to consider include the acceptable average response time for queries and updates and the time it takes for the nightly (batch) processes to complete. The latter of which is really a consideration for large companies. On these systems, a system without enough resources could cause the nightly batch time to exceed the off-hours time available.

BillMax is primarily a database driven system. As such its performance is enhanced with fast disks and large amount of random access memory. Raw processing power is a secondary concern. With these points in mind, a typical ideal specification for most companies would be:

- 8-16GB RAM
- 2-4 CPUs
- 100GB solid state disk

For more information on sizing, please contact BillMax Support (*support@billmax.com*).

Install BillMax

Due to changes in the distribution of CentOS, BillMax has opted to use Debian as its Operation System. These instructions refer to a new installation of BillMax on Debian 12.x. If upgrading BillMax from CentOS, please see *Converting BillMax to Debian* on page 35.

1. Build and deploy a base instance Debian 12.x.

   Only the **SSH** server and the standard system utilities packages should be installed. Do not install other packages. The BillMax install script will handle all dependencies and preconfigurations.

2. Download the latest BillMax Debian package.
   a) Using a web browser, go to *http://www.billmax.com*.
   b) Click **CUSTOMER PORTAL**. Click **LOGIN**. Contact *support@billmax.com* for the user name and password if you do not know your credentials.
   c) Select Files from the main menu and identify the appropriate **DEB** file.
   d) Alongside each file listing are two links. To download the file to your device (for eventual transfer to the BillMax host), click on the first link (it has a downwards pointed arrow).

e) Otherwise, to get a `wget` command that can be used on the BillMax host to download the file directly, click the second link (rectangular box). This action copies the command to the clipboard. Paste the command and execute it in the command line on the BillMax host as the root user. The command must be used within several minutes of the selection in the portal.

3. Install the BillMax package.

a) Become the **root** user (use `su -`) so `/usr/sbin` is in your path. Note the "-".

b) Execute `cd <install directory>`

c) Execute `apt install ./billmax-YYYY.4.0.RRRR-Debian12.amd64.deb`

> 📝 **Note:** YYYY is the release year and RRRR is a release number; it will vary based on availability and the file you downloaded.

> 📝 **Note:** The relative path above is important. The command will not succeed without the path.

> 📝 **Note:** The command will complain with the message "Download is performed unsandboxed as root as…". This is to be expected and not an error.

d) To begin the second stage of the install, execute `cd /opt/billmax/install`

e) Execute `./BxInstall.pl`.

> 📝 **Note:** You will be prompted to enter the system user that BillMax uses. Accept the default 'billmax'.

> 📝 **Note:** You will be prompted to enter the *hostname*, *username* and *password* to use for access to Subversion (*svn*). BillMax uses Subversion for local code management. Accept the default 'billmax' for *username* and *password*. For *hostname*, accept the default '127.0.0.1'.

> 📝 **Note:** When you are prompted to enter a password for the mariadb root account, hit enter as there isn't one initially.

> 📝 **Note:** Select n for `Switch to unix_socket authentication`.

> 📝 **Note:** Enter a password and remember it for the next installation stage.
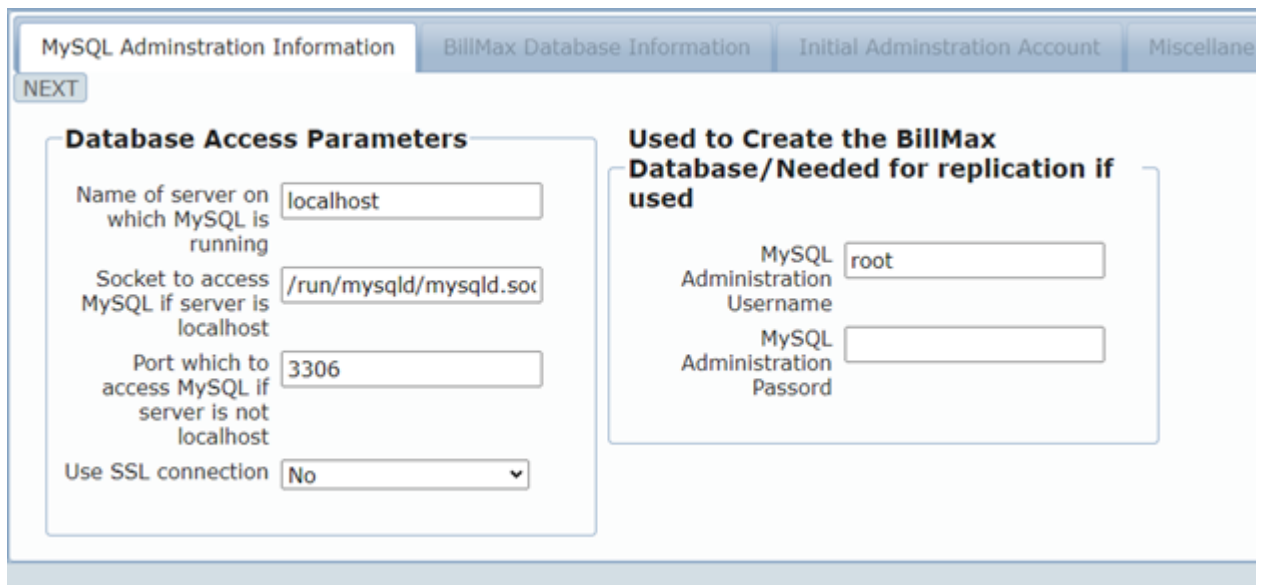
> 📝 **Note:** Accept all other defaults for the *mariadb* configuration.

f) Once the *BxInstall.pl* script finishes, use your web browser to handle the remainder of the install with the URL: `https://<your billmax host ip or name>/cgi-bin/BxConfigure.cgi`

You will get a warning about a insecure connection. This is normal as a legitimate web certificate is not installed yet.

g) Type the password for the mariadb account entered previously.

| MySQL Adminstration Information | BillMax Database Information | Initial Adminstration Account | Miscellane |

NEXT

Database Access Parameters

Name of server on which MySQL is running: `localhost`

Socket to access MySQL if server is localhost: `/run/mysqld/mysqld.soc`

Port which to access MySQL if server is not localhost: `3306`
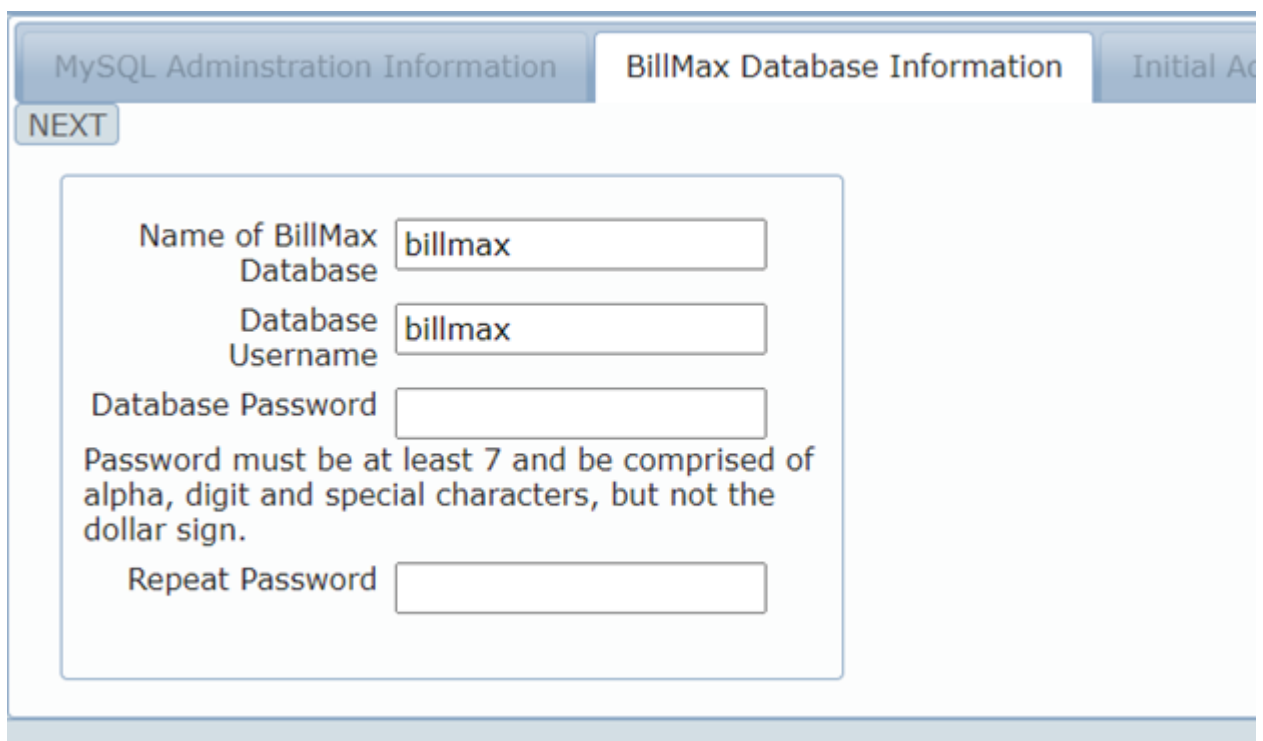
Use SSL connection: `No`

Used to Create the BillMax Database/Needed for replication if used

MySQL Administration Username: `root`

MySQL Administration Passord:

h) On the next screen, enter a password for the 'billmax' database user. - This is the internal user used by BillMax applications.

| MySQL Adminstration Information | BillMax Database Information | Initial Ac |

NEXT

Name of BillMax Database: `billmax`

Database Username: `billmax`

Database Password:

Password must be at least 7 and be comprised of alpha, digit and special characters, but not the dollar sign.

Repeat Password:

i) On the third screen, enter *login id*, *password*, and *email address* for an administrator to use to gain access to BillMax.

MySQL Adminstration Information   BillMax Database Information

NEXT

Initial BillMax
Administrator
Username | Admin

Password |

Password must be at least 7 characters in length
and be comprised of alpha, digit and special
characters.

Repeat Password |

Email |

j) On the fourth page, enter a password to encrypt sensitive data in the BillMax database. Save this value as it will be needed whenever you are required to re-encrypt this data. Also enter the license key. This key can be obtained from BillMax support. In order to request a key, you'll need to supply the shown *Machine ID* and *IP address*.

k) Click COMMIT button.

4. Upon a successful install, you'll be directed to the login screen for BillMax. You can login with the administrator(*Admin*) credentials provided on the third screen.

5. You are ready to configure the system for billing, payment processing, service provisioning, etc. Please see appropriate sections as needed.

Other Administrative Tasks

EMAIL. BillMax relies on the Mail Transfer Agent (MTA) program installed to send email in asynchronous and reliable way. The BillMax installation scripts installs Postfix for this purpose. To send email from BillMax, `Postfix` must first be configured. To send email reliably, a number of spam counter measures have to be implemented. These include `DNS` changes, `SPF`, `DKIM`, and `DMARC`. This setup is most often achieved by relaying email through an already suitably configured mail server that the company maintains. It is also possible with much more configuration to send email directly from the BillMax host. Lastly, BillMax does offer an email service via

AWS SES. Contact BillMax sales for a quote. For guidance on setting up email please contact BillMax support or the many online web resources on this topic.

PRINTING. For printing, BillMax relies on the CUPS printing system. The BillMax installation script installs CUPS but it is not configured. CUPS can be administered via command line tools or a web based console. Once a printer is configured, the printer can be specified within BillMax thereby allowing printing of various documents within BillMax (statements, invoices, receipts). For guidance on setting up printing please contact BillMax support or the many online web resources on this topic.

DNS. You may choose to give your BillMax a host name. A host name is a requirement if you install certificates (see WEB CERTIFICATES).

WEB CERTIFICATES. BillMax installs a self signed certificate. This allows **https** access to the BillMax web interface. Self signed certificates are not trusted by browsers so this use will result in warnings similar to "Your connection is not private". To fully secure the BillMax interface, you should purchase and install a legitimate certificate from a commercial certificate authority or perhaps use the Lets Encrypt certificate authority. If you decide to host the customer portal on the same host as BillMax, installing a legitimate certificate for it is mandatory.

OTHER. Additional tasks that may be necessary:

1. Implement a backup strategy that does off site backups (at a minimum) of /usr/local/billmax, /opt/billmax, /var/www/mysql. Care must be taken when backing up the database. Either the database must be stopped or a utility program like *mariabackup* must be used. See *Backups, Redundancy and Archiving* on page 163
2. Change the system timezone or time on the server.
3. Change the system local host name.
4. Configure system services (Customer portal, REST API, fetchmail, FreeRadius, etc). Documentation on these systems is available elsewhere in the documentation.

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 172.
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*.

Configure Lists

BillMax has many lists to make it easy to customize the system to better meet your local environment. Lists can be drop down lists of items or a set of parameters to meet your business needs.

1. Login to the BillMax Staff Portal using an ID with Administrator privileges.
2. Select **Billing Administration** > **Lists**.
3. Using **chargetypes**(*Charge Type* on page 52) as an example, select the highlighted underlined number in front of **chargetypes**.
4. To add a list item, select **Add List Item**.
5. Enter **Item** as the drop-down choice.
6. Enter **Description** if more explanation is needed or desired.
7. Click **Save**.

Update Lists

We recommend updating this lists before going live with BillMax.

adcodes	List of advertising codes - reasons people found your company
bankaccounts	List of bank accounts that money will be deposited
cancelreasons	List of reasons that customers list your service
chargetypes	List of categories that revenue is earned or QuickBooks® income accounts
nosuspendoverduedates	List of holidays not to suspend accounts. See *Avoid Suspending a Service on Holidays* on page 136
accountclasses	List of Account Classes. Used for financial reports. Also used for product selection.

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

Add Custom Field Definitions

Additional fields may be added to particular database tables to capture unique information for your company.

1. Select **System Administration** > **Custom Fields Definitions**.
2. Click the **New** button on the Context Menu.
3. Enter **Name** and choose the **Table**.
4. Click **Save**.
5. For each Custom Field, click the **Add** button and enter the field details.

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**.

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*.

Configure Taxes

For the internal tax engine, see *Configure Taxes Using the BillMax Tax Engine* on page 184.

Connect POPs/Access Points to Packages/Service

BillMax has several steps to tie a package/service to a POP and Access Point for Fixed Wireless Services.

1. Set up any needed custom fields. See *Add Custom Field Definitions* on page 30

2. Add all POPs. See *Create POP Regions* on page 282

3. Set up all access points for each tower location. See *Create Access Points* on page 283

4. On a Package or Service Definition for Fixed Wireless services, the settings for **POP Required** and **Access Point Required** on the **Provisioning Information** tab must be set to Yes.

5. Before a Package or Service can be added to a User, the **POP** and **Access Point** must be selected from the choices on the User.

6. All Services on an Access Point can be viewed on the **Reports** > **Service By Access Point.** The color of the Access Point will match the color of the Services that it is connected too. All unconnected Services will be black.

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. 📝   **Note:**

```
When using the BillMax tax engine, if no tax is to be collected and
 there
is no requirement to report non taxable sales in your state, set
taxable on the service definition to 'Not Applicable' instead of 'No'
 to avoid
tax columns on billing documents.
```

Enter **Recurring/Usage Taxable**.

7. 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 ⑦ give detailed information about a field.

8. Select the **Provisioning** tab to enter provisioning field information. Documentation for Radius provisioning is included in *BillMax RADIUS Implementation* on page 134 .

9. 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.

10. Select **Save**.

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.

Item	Value
ResultsEmail	**Admin email address**
State	Active
Test	0
UserName	Processor Terminal Id

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

Converting BillMax to Debian

Topics:

- *Concepts*
- *How To*

Chapter

1

Concepts

Topics:

- *Overview*

Overview

Beginning with the 2024 4th quarter release of BillMax, BillMax has provided a CentOS 7 release as well as a Debian 12 release. As of March 31, 2025 BillMax will no longer provide CentOS 7 releases. CentOS 7 End of Life (EOL) was June 30, 2024 and each company is encouraged to transition to using a Debian server. This change was necessitated by changes to the CentOS release cycle where stable enterprise point releases will no longer be maintained by IBM.

For existing BillMax CentOS 7® customers, the upgrade path is different than a normal upgrade where BillMax can be updated in place with an installation/upgrade script. Instead of a upgrade in place, another server running Debian will have to be deployed where BillMax will be installed as a new installation. Then the database files will be migrated and converted from the existing CentOS server to the Debian server. Finally, various conversion scripts are run to update the BillMax database schema.

Switching servers and operating systems involves quite a few extra considerations. In addition to BillMax itself, any external configuration or subsystems that BillMax relies on will need to be addressed. These subsystems include but are not limited to Email, Printing, and service provisioning. These topics are not covered here. Please contact BillMax (*support@billmax.com*) for further assistance.

To convert to Debian, the process is to deploy Billmax on a new server, configure that server as needed, migrate the database as needed. All the while testing the result. Once satisfied that the system operates correctly, then the final step is a suspension of activities on the CentOS server followed by one last data migration. If successful the old server is shut down and operations resume on the new server. If not, operations resume on the CentOS server and another attempt is made once adjustments have been made.

Chapter
2

How To

Topics:

- *Converting BillMax to Debian Installation Steps*

Converting BillMax to Debian Installation Steps

1. Deploy a new server running Debian 12.x. See *Install BillMax* on page 24. Request a new license key making sure to let BillMax support know that this key is for your new Debian host. Remember your choice for the Database Password (on the BillMax information tab). You will need to supply this password later.

2. 📝 **Note:** You will need a license key prior to upgrading the BillMax database schema.

   Login into the new BillMax staff interface. Go to **System Administration** > **License.** Copy and paste (save) the License Key.

3. 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*

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

5. Stop *mariadb*, *httpd* and *edge* on the CentOS 7® server. As root execute: `systemctl stop mariadb httpd bx_edge`

6. On the CentOS 7® server, make a copy of the entire database instance (this includes billmax as well as all other databases). As root execute:
   a) `cd /var/lib`
   b) `tar czf mysql.yyyy-mm-dd.tgz mysql` (where **yyyy-mm-dd** is the current date).

7. Restart *mariadb*, *httpd* and *edge* on the CentOS 7® server. As root execute: `systemctl start mariadb httpd bx_edge`. Do not restart if this is your final data migration.

8. Copy the mysql.yyyy-mm-dd.tgz file to the Debian host: `scp mysql.yyyy-mm-dd.tgz admin@<Debian ip` or `host name>:`

9. 📝 **Note:** The rest of these steps are on the Debian server.

   Stop *mariadb*. As root execute: `systemctl stop mariadb`

10. Move the existing *mysql* instance out of the way. As root execute: `cd /var/lib; mv mysql mysql.original`

11. Restore the database files from the CentOS 7® server. As root execute:
    a) `cd /var/lib`
    b) `tar xzf /home/admin/mysql.yyyy-mm-dd.tgz`

12. Restart *mariadb*. As root execute: `systemctl start mariadb`

13. 📝 **Note:** You will need the root password from the CentOS server for this step. You can retrieve this password from `/usr/local/billmax/local/billmax.conf` (MYSQL_ADMIN_PASSWD setting).

    Fix a few *mariadb* settings. Execute:
    a) `mysql -u root -p mysql`
    b) CREATE USER *'mariadb.sys'*;
    c) ALTER USER *root@localhost* IDENTIFIED VIA unix_socket;
    d) FLUSH privileges;
    e) quit

14. Run *mariadb* command for upgrading databases to a new version. As root execute: `mysql_upgrade --force`

15. Restart mariadb. As root execute: `systemctl restart mariadb`

16. Check the system log for any *mariadb* logs on startup: `journalctl -xeu mariadb`

17. Restore license key. Copy the license key to the clipboard. Paste key when prompted. As the billmax user execute: `/usr/local/billmax/bin/license_install`

18. 📝 **Note:** Be sure you have tuned the database before converting it.

Convert BillMax tables to *InnoDb* and *UTF8* character set. As the billmax user execute: `/usr/local/billmax/bin/convert2InnoDB -p 15`

19. Update BillMax database schema and run upgrade scripts. You will be prompted to enter the BillMax version that is running on the CentOS server. You can get this value via the **Help** > **About** menu. Supply the full version string (4.0.XXXXXX). As the billmax user `cd /usr/local/billmax/pkg` and execute: `/usr/local/billmax/pkg/upgrade_db_only.pl`

20. Verify that the database has no errors. As the billmax user execute: `/usr/local/billmax/bin/dbck`

21. Make sure that the nightly processes are not configured to run. Go to **System Administration** > > **Batch Processing** and disable all active batch groups. If performing the final migration, re-enable the same batch process groups. This process ensures that the `crontab` is updated properly.

22. While testing, you should change credit card processor to the dummy processor called 'Loopback'. Set the State value for all active Credit Card and Bank Account lists to 'Not Active'. Set the State value for the Loopback lists to 'Active'. Also you should disable printing and email by setting the appropriate commands in the Global Options to `/bin/true`. These steps should not be done for the final data migration.

23. Consider using BillMax test tools to run the system forward to test batch processing. Contact support for details.

24. Copy the runtime files from your CentOS server to the Debian server. Run time files include any data file, custom configuration files, provisioning scripts and so on. The following commands copy the required (bare minimum) files. Depending on your use, you may need additional files. For example, you will want any customizations you have made like changes to templates, source files, etc. On the CentOS server. As root:

    a) `cd /usr/local/billmax`
    b) `tar czf runtime.yyyy-mm-dd.tgz backups documents efp local logs portalv2/config/edge.php uploadedfiles` (where *yyyy-mm-dd* is the current date)
    c) `scp runtime.yyyy-mm-dd.tgz admin@<Debian ip or host name>:`

25. Redo any local customizations that were noted in `upgradeddifferences.dat`.

26. On the Debian server as root:

    a) **Note:** This step omits the local directory. For the local directory, extract it separately to a work directory. Then selectively copy the needed customized files.

    `cd /usr/local/billmax`
    b) `tar xzf /home/admin/runtime.yyyy-mm-dd.tgz backups documents efp logs portalv2/config/edge.php uploadfiles` (where *yyyy-mm-dd* is the current date)

27. Review, configure and test all subsystems (backups, email, printing, provisioning, etc) that BillMax relies on.

28. Once satisfied that everything is working correctly and all subsystems are configured and tested, schedule the final data migration. Repeat steps 3-18 but do not enable processing on the CentOS server. If everything goes to plan, be sure to disable all services on the CentOS server. Make DNS changes as needed, etc.

Part

II

Fundamentals

Topics:

- *Automated Processing*
- *Billing*
- *Bookkeeping*
- *Broadband Labels*
- *Customer Organization*
- *Customer Communications*
- *Email Processing*
- *Financials*
- *Inventory Management*
- *Payments and Refunds*
- *Products*
- *Provisioning*
- *Remote Applications*
- *Staff Portal*
- *System Administration*
- *Taxes*
- *Reference*
- *Disable BillMax System*

Disable BillMax System

1. Choose **System Administration** > **Global Options**. Change **Path to the sendmail Program** to `/bin/true` to disable email notifications.

2. Change **Path to lpr Program** to `/bin/true` to disable printing.

3. To suspend all payment processing, edit the lists for your payment processor - one for credit card and one for e-checks. Change **State** to Not Active. If you have more than one merchant account, make sure you modify additional lists for those accounts.

4. To disable **Customer Portal**

   a) As root, run `systemctl stop bx_edge; systemctl disable bx_edge`

   b) Take down hosting or access to portal (client side) web pages.

5. Disable batch processing. Choose **System Administration** > **Batch Processing**. Select every group that is active to **Disabled**, Review listing to make sure all groups show as "DISABLED".

Chapter

1

Automated Processing

Topics:

- *Concepts*
- *How To*

Concepts

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 Group**s are available. When BillMax is installed, no **Batch Process Groups** are enabled.

The current list of **Batch Process Group**s installed with BillMax are:

- **nightly** - enable this when putting BillMax into production. See *nightly Typical Batch Processes* on page 239.
- **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 Group**s and may also add their own **Batch Process Group**s.

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

2

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.

Billing Models

BillMax allows for multiple recurring billing models in addition to one time sales and general sales.

1. 1st of Month Billing - bill everything of the current month, i.e. 1/1-1/31.

    a. Settings

        a. Set Account Profiles->Billing Day of Month/Prorate Day to 1.
        b. Account Profiles-Billing Timing should have a Monthly Bill Sending option set.
        c. All Accounts should have Billing/Prorate Day of Month to 1.

          d. All Package/Service Definitions should have Prorate Dates/Amount set to Yes.
    b. Effect

          a. Customers will get a bill once a month on the 1st.
          b. Any Additional Recurring Services added mid-month will be pro-rated.
    c. Options

          a. If Post Pending Transactions Daily set to No, mid-month billing not added until the 1st. If Post Pending Transactions Daily set to Yes, mid-month billings added to balance.
          b. To send bill xx days early, set Service Offset Days to xx. The billing date will float due to number of days in a month.

2. Anniversary Billing

    a. Settings

          a. Set Account Profiles->Billing Day of Month/Prorate Day to "Date of Account Creation".
          b. Account Profiles-Billing Timing should have a Monthly Bill Sending option set.
          c. All Accounts should have Billing/Prorate Day of Month to 1.
          d. All Package/Service Definitions should have Prorate Dates/Amount set to Yes.
    b. Effect

          a. If Service is Pending, the day the Account is billed will be modified when the first Package/Service is opened.
          b. Accounts billed on the 29/30/31 will be billed on that day or on the last day of the month.

3. Asynchronous Billing - no prorating, bill once a month

    a. Settings

          a. Set Account Profiles->Billing Day of Month/Prorate Day to "Date of Account Creation".
          b. Account Profiles-Billing Timing should have a Monthly Bill Sending option set.
          c. All Accounts should have Billing/Prorate Day of Month to 1.
          d. All Package/Service Definitions should have Prorate Dates/Amount set to No.
    b. Options

          a. With Service Offset Days set to 0, billings that are on the same day as the billing date are prebilled. Billings that are after the billing date are partially prebilled and partially post billed.
          b. With Service Offset Days set to 1 and Offset Day of Month selected, all Accounts will be billed on the Billing/Proate Day of Month for all Services that have a billing date less then the next billing date. Accounts with Billing/Prorate Day of month of 29-31 will be billed on the 28th.
          c. With Service Offset Days 2-28 and Offset Day of Month selected, the actual day of billing will depend on the Billing/Prorate Day of Month.

4. Daily Billing - no prorating, bill multiple times a month

    a. Settings

          a. Account Profiles-Billing Timing should have a Daily Bill Sending option set.
          b. Posting Pending Transactions should be set to Yes.
          c. All Package/Service Definitions should have Prorate Dates/Amount set to No.
    b. Effect

          a. Billings occur multiple times a month if there are multiple services with different service periods.

                Note: On the Bill Timing tab, click Billing Date Examples to verify your selections will produce the billings as expected.

Charge Type

A Charge Type is a classification of how revenue is made. This classification plays an important role in how the BillMax system reports sales data. When setting up packages and services, the charge type is chosen from a drop down list. Charge types are defined in the **chargetypes** list .

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 168 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.

📝 **Note:** When both a set amount and a percentage are specified on a Service or Package, BillMax computes the percentage amount then applies the least of the computed percentage amount and the set amount. The set amount provides a discount "ceiling".

Fees

BillMax allows for a variety of fees. Below is a chart of fees available:

Table 1: Fees

Name	Fee/ Percentage	Recurring/ One Time	Setup Location	When Charged
Credit Card Decline Fee	Flat Fee	As Required	**Account Profiles**	based on **Account Profile** settings
Document Fee	Flat Fee	Recurring	**Account Profiles/ Documents**	added to each printed bill based on **Account Status**
Invoice Fees	Percentage Based	Recurring	**Billing Administration** menu item	added to each invoice one time based on the positive sum of transactions and the Virtual Company
Late Fees	Either	As Required	**Account Profiles**	based on **Account Profile** settings

Name	Fee/ Percentage	Recurring/ One Time	Setup Location	When Charged
One Time Fee	Flat Fee	One Time	**Billing Administration** menu item	can be part of a **Package Definition**, a **Bundle**, or added to an Invoice
Overdue Fee	Flat Fee	As Required	**Service Definition**	charged in addition to any Late Fees assessed
Recurring Service Fees	Flat Fee	Recurring	**Billing Administration** menu item	specified on a **Package/Service Definition**, can be based on a **Selectable Fee Group**
Universal Service Fees	Percentage Based	Recurring	**Tax Item/Fee**	specified as part of a **Tax Group**

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. The **Late Fee** provides a late fee "floor".

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**.

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.

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.

Usage Based Billing

Usage Billing is based on statistics. Pricing is applied to the statistics based on what is specified in a Tierplan. A tier plan outlines what measurements and what computations will be used to calculate the usage based computation result. A tier plan may contain zero or many tiers.

There are two steps in Usage Based Billing. First, BillMax uses data contained in the stats table to calculate the single value (the usage based computation result) of the specified mathematical computations on the applicable statistics. Next, the value is used in determining what should be billed. Both steps are configurable.

Bandwidth Usage Billing using Preseem®'s QOE Platform

A byproduct of Preseem's bandwidth management is the capture of bandwidth (bytes transferred) usage by an ISP's end customer. This usage is made available to billing applications via an Application Programmer Interface (API).

Using this API, BillMax can be configured to receive and process bandwidth usage in near real time. This processing includes a number of actions or abilities, each of which is optional. These actions include:

- Notifications to customers via email one certain thresholds are reached
- Periodic usage reports via email
- Usage overage billing (linear or tier pricing)
- Suspension once certain usage threshold is reached
- Customer Portal reports

Here is an outline of the process flow:

Poll Preseem periodically (every X minutes) for usage data for a specified time period. A typical scenario is to poll every 15 minutes for the preceding 15 minute time interval (for example at 10:30 retrieve usage for the time period 10:00 till 10:15).

Data returned includes the BillMax service number and bytes downloaded and uploaded. Data returned is reformatted as Radius accounting detail dataset and then loaded into BillMax via the *radparse* program.

For each service, a record is written to the *datacall* table provided that service exists in BillMax. If the service is associated with a usage billing plan (see Tier plans) then notifications and suspension (if applicable) takes place.

If a usage billing plan is used, BillMax will bill usage where applicable each billing period.

To implement, first check with Preseem support to ensure this feature is available in your Preseem instance. If available, they will provide a API key to be used when making the necessary API calls.

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:

   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.

Bill Usage from Preseem®

What to do when billing a customer for bandwidth usage based on data from Preseem's QOE Platform.

1. First check with Preseem support to ensure this feature is available in your Preseem instance. Note; at a minimum your services must be known to Preseem (e.g. bandwidth management is already in place).

2. Edit the Preseem list (**Billing Administration** > **Lists**). Set the *Userid* value to the *API Userid* (also may be called Key) provided by Preseem. Set the *Delay* (interval) in minutes as desired.

3. Configure one or more *Usage Tier Plans* as needed.

4. Configure Recurring Packages and Service plans as needed. For each specify the appropriate *Usage Pricing Plan* under *Tierplan Pricing* created in step 3.

5. Copy preseem script to *bin* directory: *cp /usr/local/billmax/local/preseem.pl /usr/local/billmax/bin*.

6. Configure *cron* job for polling Preseem usage at **System Administration** > **Batch Processing** > . Create a new batch with timing consistent with the Delay list setting. For example for 15 minutes, enter: Minute = 0,15,30,45 Hour = * Day of Month = * Month = * Day of week = *. In the new batch, create a new process. Specify `Process = preseem.pl` and `Args = -a getusage`.

Commissions

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 have to rebuild the Services if Provisioning is involved.

Configure Tier Plan Definition

Describes how to setup up a tier plan.

1. Click **Billing Administration** > **Usage Tier Plan** .
2. Click **New** or click the **Number** of an established tier plan to **Clone** the settings of an existing plan as a starting point.
3. Enter **Description** for the name of the usage tier plan.
4. Choose **Method to Compute Value** from the following:

   • **Precentile** - All the samples for a given period are sorted, then the top x% are discarded. x is defined as 100 minus *percentile*. The largest remaining sample is the usage based computation result.

   Sample values are 1,2,4,7 and 20. Percentile is 80% The usage based computation result is 7.0.
   • **Average** - The usage based computation result is the average of the samples.

   Sample values are 1,2,4,7 and 16. The usage based computation result is 6.0.
   • **Max** - The usage based computation result is the maximum of the samples.

   Sample values are 1,2,4,7 and 16. The usage based computation result is 16.0.
   • **Min** - The usage based computation result is the minimum of the samples.

   Sample values are 1,2,4,7 and 16. The usage based computation result is 1.0.
   • **Sum** - The usage based computation result is the sum of the samples.

   Sample values are 1,2,4,7 and 16. The usage based computation result is 30.0.
5. Choose **Direction** for the recorded statistics if using a comma delimited list of samples stored in the form *name=value*. Values can be stored in the stats table as *in='value', out ='value'*or both. Choose *Not Applicable* when using the **Sum** computation method.
6. Choose an answer for **Can Go Down**. If the usage based billing can ever be billed at a lower priced tier after having been billed at a higher priced tier, select *Yes*.
7. Enter a unit for measure for **Sampling Units**.

This is for descriptive purposes only. Specifying units such as MB and capturing the data in Mb will cause incorrect billing and confusion.

8. Choose an answer for **Units are International System**.

   Presently this option is used only when loading states data from the *datacall* table and in *datacall* reports.

9. Enter the **Usage Display Precision** to specify the decimal precision to customers. It does not effect some values displayed in the staff interface.

10. Choose **Sample Type**. The setting links the type of sample specified with statistics entered in the stats table using *enter_stat* with the *stype* parameter set to the value specified here.

11. Enter **Purge Samples Frequency**/

    The applicable statistics are purged even if they have not been used for a usage based billing result.

12. Enter **Email Report Frequency (days)**. A usage report for a **Service** will be mailed to the email address of the **Account** at this interval.

13. Enter the **Email Warning Cost** to send an email alert to the customer whenever the difference between the current usage cost and the previous usage cost reaches this amount.

    The email alert is available only for MIN, MAX, and SUM methods. If the value is set to 0, this option is disabled.

14. Enter the **Suspension Amount** to suspend the customer for the remainder of the service's usage cycle whenever the usage computed value reaches this value.

    Suspension is available only for MIN, MAX, and SUM methods. If the value is set to 0, this option is disabled.

15. Enter the **Suspension Cost** to suspend the customer for the remainder of the service's usage cycle whenever the usage cost value reaches this value.

    Suspension is available only for MIN, MAX, and SUM methods. If the value is set to 0, this option is disabled.

16. Fill in any other data.

17. Click **Save**.

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.

Create One Time Fee Using a Selectable Fee Group

One Time Fees can be added and billed with a Package or Bundle. When the Package is billed the first time, the One Time Fee is billed also. Some One Time Fees are limited by the Selectable Fee Group that they belong to. Example: POP

1. Select **Billing Administration** > **Lists**.

2. Select **feeselectgroups**.

3. Click **Add** and enter the name in the **Item** and what to **Match on**.

4. If the item has to match for the fee to be charged, click **Match Required.** Matches are made on **User** information.

5. Select **Billing Administration** > **One Time Item/Fee**.

6. Click **New** and specify the **Name, Billing Display, Charge Type, and Price** and the **Selectable Fee Group**.

7. Enter the criteria for the **Selectable Fee Group**.

8. Specify the other fields as desired and click **Save**.

9. To add the **One Time Fee**, bring up the Package/Service definition. On the **Package Elements** tab, click **Add**. Choose the **One Time Fee Group**. **One Time Fees** can be added a **Bundle Element** also.

> Example: List item "Install Fee" matches on POP. Create a One Time Fee for each POP needed or a default "Match Any" for all POPs.

Recommendation: Consider creating a default One Time Fee to avoid errors if an item has to be matched when a package is created. Otherwise, leave **Match Required** unchecked and the item will be matched if possible.

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 Amount**s are credited only to **Open** or **Suspended Account**s.
   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 Recurring Service Fees Using a Selectable Fee Group

Recurring Service Fees are flat recurring fees that are added to a Package or Service Definition. When the Package or Service is created, the Recurring Service Fee is created also. Some Recurring Service Fees are limited by the Selectable Fee Group that they belong to. Example: County

1. Select **Billing Administration** > **Lists**.
2. Select **feeselectgroups**.
3. Click **Add** and enter the name in the **Item** and what to **Match on**.
4. If the item has to match for the fee to be charged, click **Match Required.**
5. Select **Billing Administration** > **Recurring Service Fee**.
6. Click **New** and specify the **Name, Billing Display, Charge Type, and Price** and the **Selectable Fee Group**.
7. Enter the criteria for the **Selectable Fee Group**.
8. Specify the other fields as desired and click **Save**.
9. To add the **Recurring Service Fee**, bring up the Package/Service definition. In the Fees section, click **Add**. Choose the **Selectable Fee Group**.

> Example: List item "County Recovery Fee" matches on County. Create a Recurring Service Fee for each County needed or "*" for all counties.

Recommendation: Consider creating a default Recurring Service Fee to avoid errors if an item has to be matched when a package or service is created. Otherwise, leave **Match Required** unchecked and the item will be matched if possible.

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

Setup a Payment Plan for a One Time Item

1. View the Account.
2. Select **Sale** > **Payment Plan** .
3. Choose **Item** from the list of one time items.
4. Enter **Default recurring amount**, the date to begin billing - **Next Date**, and the **Number of Times** the amount will be charged. Change the **Interval** if necessary. The default is each month.
5. Click **Generate Plan**.
6. Review the **Total** (under the buttons). For an installment plan it should be the same as the Item but it can be more or less if a charge is incurred for the payment plan or the price was negotiated.
7. To edit the payment plan, click **Delete Plan** and make adjustments or alter the dates and amounts in the payment plan.
8. Update any other fields necessary and click **Save** to generate the **Pending Transactions** that will be placed on upcoming bills as time passes.

Chapter

3

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 fund by type of tender (cash, check, Visa®, MasterCard®, electronic check, etc.). These are automatically set by BillMax.
2. The actual bank account to which funds are deposited. An example would be a business checking account at a local bank. These are created using the BillMax list *bankaccounts* with the **Item** field as the name of the bank.

When an IIF file is created, Bank Deposits, created in the Staff Portal, 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.

When a CSV file is created for NetSuite®, Bank Deposits are used to create a deposit transaction into the bank account specified on the BillMax Bank Deposit using the Description. The funds for the NetSuite® deposit transaction come from the BillMax tender bank accounts created and funds populated into the NetSuite® CSV 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
- Sales Tax
- Sales Tax Reversal
- Credit Tax
- Credit Tax Reversal/Write off

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. If NetSuite® is used for accounting, then a CSV file may be downloaded for import for the Balance Forward A/R.

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 Bookkeeping report is the basis for creating an 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 provides a list for mapping the Item, Income Account and Class values to those desired in QuickBooks.

The data displayed by the Bookkeeping report can also be the basis for creating a CSV file suitable for importing into NetSuite®. Lists allow for custom setup for NetSuite® income account mapping and exports.

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 **movepayhistcustomerof**.
4. Make edits.
5. Click **Save**.
6. Choose **movepayhistprofile**.
7. Make edits.
8. Click **Save**.

Setup NetSuite® in BillMax

These steps will ensure that NetSuite® can be updated with a file created by BillMax.

1. Choose **Billing Administration** > **Lists** to review and update the lists in BillMax for NetSuite®.
   a) 📝 **Note:** Accounts can be classified in BillMax by Charge Type (how revenue is made), Service Class or Service Definition. Charge Type classification is recommended.

   Select the list named **chargetypes** and add a **List Item** for each revenue category. The first six are reserved by BillMax.
   b) Select the list named **netsuite** and update the **Values** as needed. These items are the default accounts in NetSuite®.
   c) 📝 **Note:** The first time you run the Bookkeeping Report after your initial entries into this list, BillMax will give you a list of missing mappings to add.

   Select the list named **netsuiteincomeaccount** and add a **List Item** for each item that should be mapped to a revenue account. The **Item** is the Account Classification by grouping, each separated by

a colon. The **Value** is the NetSuite Account number. The **Description** is the name of the Account in

List Items

	Add	Select	Toggle Portal Usage				
Number		**Item**	**Value**	**Description**	**Omit in Portal?**	**Discontinued?**	**Select**
1432		BillMax:Recurring:Internet Service	40204	Internet Sales	No		
1433		BillMax:One Time:Internet Service	40204	Internet Sales	No		
1434		Billmax:Recurring:VOIP Service	40205	VOIP Sales	No		
1435		BillMax:One Time:VOIP Service	40205	VOIP Sales	No		

NetSuite.

   d) Select the list **bankaccounts** and add a **List Item** for each NetSuite bank account. The **Item** is the name of the bank. The **Value** is greater than or equal to 1. The **Description** is the bank account number in NetSuite.

2. As each Package and Service Definition is created, choose the appropriate **Charge Type**.

3. Choose **Reports** > **Bookkeeping** to see the report and create the NetSuite CSV file.

   a) Set the **Start Date** and **End Date** .

   b) Choose **Date Grouping**.

   c) Choose **Companies** to report on.

   d) Choose **Type of Sales Report** and **Classification**.

   e) Choose **Account Classification**.

   f) Click and sort the Grouping options so that the order of the options match the grouping options in **netsuiteincomeaccount** for each item.

   g) Click **Run Report** to preview the information and click **Download NetSuite GL** to verify the mappings appear correctly by editing the resulting file.

   h) Create an **Option Set** to retain these settings for each file creation in the future.

4. Import the resulting CSV file into a test instance of NetSuite to verify that all the mappings come across as expected.

Setup Quickbooks® in BillMax

These steps will ensure that Quickbooks® Desktop can be updated with an IIF file created by BillMax.

1. Choose **Billing Administration** > **Lists** to review and update the lists in BillMax for Quickbooks®.

   a) 📝 **Note:** Accounts can be classified in BillMax by Charge Type (how revenue is made), Service Class or Service Definition. Charge Type classification is recommended.

Select the list named **chargetypes** and add a **List Item** for each revenue category. The first six are reserved by BillMax.

   b) 📝 **Note:** The first time you run the Bookkeeping Report after your initial entries into this list, BillMax will give you a list of missing mappings to add.

Select the list named **qbincomeaccount** and add a **List Item** for each item that should be mapped to a revenue account. The **Item** is the Account Classification by grouping, each separated by a colon. The **Value** is the Quickbooks Income Account.

List Items

	Add	Select	Toggle Portal Usage					
Number		**Item**	**Value**	**Description**	**Omit in Portal?**	**Discontinued?**	**Select List Item**	
1432		BillMax:Recurring:Internet Service	40204	Internet Sales	No		☐	
1433		BillMax:One Time:Internet Service	40204	Internet Sales	No		☐	
1434		Billmax:Recurring:VOIP Service	40205	VOIP Sales	No		☐	
1435		BillMax:One Time:VOIP Service	40205	VOIP Sales	No		☐	

   c) Select the list **bankaccounts** and add a **List Item** for each Quickbooks bank account. The **Item** is the name of the bank. The **Value** is greater than or equal to 1.

2. As each Package and Service Definition is created, choose the appropriate **Charge Type**.

3. Choose **Reports** > **Bookkeeping** to see the report and create the IIF file.

     a) Set the **Start Date** and **End Date** .

     b) Choose **Date Grouping**.

     c) Choose **Companies** to report on.

     d) Choose **Type of Sales Report** and **Classification**.

     e) Choose **Account Classification**.

     f) Click and sort the Grouping options so that the order of the options match the grouping options in **qbincomeaccount** for each item.

     g) Click **Run Report** to preview the information and click **Download IIF** to verify the mappings appear correctly by editing the resulting file.

     h) Create an **Option Set** to retain these settings for each file creation in the future.

4. Import the resulting CSV file into a test instance of Quickbooks to verify that all the mappings come across as expected.

Chapter

4

Broadband Labels

Topics:

- *Concepts*
- *How To*

Concepts

Broadband Labels Overview

The FCC is requiring broadband labels based on nutrition labels to help consumers quickly and easily compare options and increase the transparency in the industry. For entities serving 100,000 customers or more the labels must be completed by April 10, 2024. For entities serving less than 100,000 customers labels are required on October 10, 2024.The labels must disclose important information about broadband prices, introductory rates, data allowances, and broadband speeds. They must also include links to information about network management practices, discounts (if needed) and privacy policies.

To support this new government mandate, BillMax has implemented a way to create a label for each stand-alone mass-marketed broadband access service plan. These labels are created in Draft mode so that changes can be applied and labels can be previewed. The labels are published when complete into a CSV file as well as a complete file of labels in HTML. The CSV file is to meet the requirement to make the information in the labels machine-readable to enable third parties to easily collect and aggregate data for the purpose of creating comparison-shopping tools for consumers. The HTML file will help enable the labels to be added to a website. BillMax also has a REST API endpoint to retrieve the label for a certain service so the label can be retrieved for an individual customer in an online portal.

If a Service Definition is updated in such a way that a new label is necessary, a new Draft of the label will need to be created. Once the label is published, the original published label remains as an Archived Label to meet the requirements of keeping labels for two years to be given to the FCC if notified or the customer in an online customer portal. Labels will remain in BillMax retrievable based on dates, service definitions, and Virtual Companies.

Broadband Labels API Overview

Included in the Broadband Label functionality from BillMax are two new REST APIs. One API returns a broadband label based on Package/Service Definition number. The other is based on a Package or Service number. If the User has a Package or Service that has been created after the Broadband Label functionality was implemented then a label of the User's Package or Service is returned. The FCC did not require broadband labels for customers' service before the law went into effect.

Figure 1: Broadband Label Based on Package/Service Number

Figure 2: Broadband Label Based on Package/Service Definition

How To

Getting Started with Broadband Labels

```
The following steps assume you have multiple mass-marketed broadband services
 defined within BillMax with the Service Type marked as Facilities Based
 Broadband under the FCC Classification.
```

1. Check your **Package Definitions** and **Service Definitions**. Find more information in *Update Package/Service Definitions Data* on page 77
2. Update required fields needed to create the label. See*Update List bblabel* on page 77
3. To work with the broadband labels, choose **Billing Admininstration** > **Broadband Labels**.

4. If a different set of broadband labels is needed for a particular Virtual Company, *Create a Tag* on page 77.

5. *Draft New Labels* on page 77 will create a set of broadband labels to review.

6. 📝   **Note: Invoice Fees** are added automatically to all broadband labels based on a **Virtual Company** tag.

   📝   **Note: Other One Time Fees** and **Recurring Fees** are also added automatically if they are set on the **Package/Service Defintion**.

   To add additional fees, click *Add Fees* on page 78.

7. Review the drafts by clicking the Broadband Label number. Review and edit any fields for a particular label. Click the tab **View Label** to preview the resulting label.

8. When the review is complete, choose *Publish Labels* on page 78 for a complete set of labels.

Update Package/Service Definitions Data

To create the broadband labels, data is pulled from the definitions. Checking key fields will ensure labels are complete.

1. There is a new field for **Latency** under **FCC Classification**. **Download and Upload Rate** are also pulled for the speeds.

2. **Contract Length** under General is the number of months that the definition is under contract.

3. If there is a **First Price** followed by a **Second Price**, then the label will show the **First Price** as the **Intro Price** and the **Number of Times to Bill** as the **Time the introductory rate applies**. If there is only one price, it will be listed as the **Monthly Price** on the label.

4. The **Fees** that are connected to a **Package/Service Definition** will be automatically added as **Provider Monthly Fees** on the broadband label.

5. The **Early Termination Fee** under Other Pricing will be added as the **Early Termination Fee** on the broadband label.

6. **One-Time Purchase Fees** will come from the **Setup Fee** and any **One-Time Items or Fees** that are listed under **Package Elements** on a **Package Definition** with a **Minimum Required** of 1.

Update List bblabel

To create the broadband labels, data is pulled from the settings in **bblabel**.

1. There is a new field for **Latency** under **FCC Classification**. **Download and Upload Rate** are also pulled for the speeds.

2. **Contract Length** under General is the number of months that the definition is under contract.

3. If there is a **First Price** followed by a **Second Price**, then the label will show the **First Price** as the **Intro Price** and the **Number of Times to Bill** as the **Time the introductory rate applies**. If there is only one price, it will be listed as the **Monthly Price** on the label.

4. The **Fees** that are connected to a **Package/Service Definition** will be automatically added as **Provider Monthly Fees** on the broadband label.

5. **One-Time Purchase Fees** will come from the **Setup Fee** and any **One-Time Items or Fees** that are listed under **Package Elements** on a **Package Definition** with a **Minimum Required** of 1.

Create a Tag

Tags are completely optional and may not be needed. A Virtual Company may have fees or plans that apply to a particular Virtual Company only and a separate set of labels is desired. To create broadband labels for a particular Virtual Company, create a tag for that Virtual Company.

1. Enter the **Tag Description**.
2. Choose **Virtual Company** from the drop down list. Click **Create**.

Draft New Labels

To create the broadband labels, data is pulled from the settings in **bblabel**.

1. There is a new field for **Latency** under **FCC Classification**. **Download and Upload Rate** are also pulled for the speeds.

2. **Contract Length** under General is the number of months that the definition is under contract.

3. If there is a **First Price** followed by a **Second Price**, then the label will show the **First Price** as the **Intro Price** and the **Number of Times to Bill** as the **Time the introductory rate applies**. If there is only one price, it will be listed as the **Monthly Price** on the label.

4. The **Fees** that are connected to a **Package/Service Definition** will be automatically added as **Provider Monthly Fees** on the broadband label.

5. **One-Time Purchase Fees** will come from the **Setup Fee** and any **One-Time Items or Fees** that are listed under **Package Elements** on a **Package Definition** with a **Minimum Required** of 1.

Add Fees

Most fees are added automatically when the broadband label is created. For additional fees, follow these steps.

1. Choose **Billing Admininstration** > **Broadband Labels**.

2. Choose **Add Fees.**

3. Enter the **Tag** .

4. Choose **BillMax Service Definition** from the drop down list for each **One Time Fee** or **Recurring Fee**. Click **Add Fee**.

5. After choosing the fees, click the button in the lower right called **Add Fees** to add these new fees to all drafts with the **Tag** selection.

Publish Labels

1. To publish a set of broadband labels to review, choose **Billing Admininstration** > **Broadband Labels**.

2. Choose **Publish Labels** from the menu.

3. Choose the **Tag** from the drop down list. For the first time, choose **All Plans** to include all definitions set as **Facilities Based Broadband**.

4. Choose **Yes** On **Output Spreadsheet** for a CSV file to be created and **Output Labels** for an HTML file of labels to be created.

5. Click **Publish** to create and use your new labels.

Chapter

5

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 125.

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.

• Scheduling when Billing Documents (Billing Statements,

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. The **Late Fee** provides a late fee "floor".

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
3. Viewing documents.

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

How To

Configure POP/AP Outage Banner for Accounts

1. Select **System Administration** > **POPs** or **System Administration** > **Access Points** from the Top Menu.
2. Choose the **Number** of the **POP** or **Access Point** with the issue.
3. Choose the **Operational Status** that applies.
4. Bring up any **Account** on that **POP** or **Access Point** and a banner is displayed based on the Status to alert the CSR.
5. To discontinue the banner, change the **Operational Status** back to **Normal**.

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

6

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 XSL: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 XSL: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:

1. Modifying `/usr/local/billmax/cfg/xsltfiles/html/documents.xsl` and `/usr/local/billmax/cfg/xsltfiles/fo/documents.xsl` for minor changes.
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 sent 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.

Emails

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 303 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 templates can be written with text or html. These include but are not limited to:

- Overdue notices
- Credit card decline notice
- Customer Portal password reset notice
- Payment confirmations

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. Multiple late notices allows escalation in the messaging. An example for overdue3 appears below.

1. Select **Correspondence** > **System 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.

```
To: $TO
From: $FROM
Subject: Account ${account.number} is Past Due - Third Notice
${account.company}
${account.cfname} ${account.clname}
${account.addr1} ${account.addr2}
${account.city}, ${account.statename}, ${account.zip}
${account.cphone}
According to our records your account is overdue for payment
 in the amount of $AMOUNT, with a total amount due of $BALANCE.
 Please remit payment to:
                Accounts Receivable
                Widgets Inc.
                ${config.addr1} ${config.addr2}
                ${config.city}, ${config.statename}
 ${config.zip}

Your service may be interrupted on or after
 ${account.projectedsuspdate}.

If you have any questions about your account, please contact the
 billing department at at ${config.bcemail} or ${config.phone}.
Thank you for your business!
<html>
<body>
<img src="/html/images/logos/logo.png"/>
<pre>
${account.company}
${account.cfname} ${account.clname}
${account.addr1} ${account.addr2}
${account.city}, ${account.statename}, ${account.zip}
${account.cphone}
According to our records your account is overdue for payment
 in the amount of $AMOUNT, with a total amount due of $BALANCE.
 Please remit payment to:
                Accounts Receivable
                Widgets Inc.
                ${config.addr1} ${config.addr2}
                ${config.city}, ${config.statename}
 ${config.zip}

If you have any questions about your account, please contact the
 billing department at at ${config.bcemail} or ${config.phone}.
```

```
<span style="font-weight: bold; font-size: larger">Your service
 may be interrupted on or after ${account.projectedsuspdate}.</
span>

Thank you for your business!

</pre>
</body>
</html>
```

Add Printed Late Notices

BillMax supports printing overdue notices for Accounts without email or those that receive a printed bill.

1. Select **Billing Administration** > **Lists** from the Top Menu. Select *systemstate*.
2. Change the Value of *AutomatedOverduePrint* to 1.
3. *Add Additional Late Notices* on page 89 as needed to escalate the verbiage.
4. To view printed overdue notices received by the customer, go to the `Account/Billing/Document History` tab and click the *number* of the overdue notice.

If you want a fulfillment partner to send out printed late notices for your company, please contact them with the format and support@billmax.com to help connect it in the system.

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.

   ℹ **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 reset 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"/> <!--
 This is link -->
    <img src="/images/logos/logo.jpg"/> <!-- This will include
                                      /usr/local/billmax/
images/logos/logo.jpg
                                      as an embedded image
 -->
    <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.

1. Copy `/usr/local/billmax/cfg/xsltfiles/fo/documents.xsl` locally. Make local edits.
2. Copy `/usr/local/billmax/cfg/xsltfiles/fop.conf` locally. Edit file and change the strings *file:cfg* to *file:/usr/local/billmax/cfg*.
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.

1. Copy `/usr/local/billmax/cfg/xsltfiles/html/documents.xsl` locally. Make local edits.
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:

- copy edited `documents.xsl` to `/usr/local/billmax/cfg/xsltfiles/html/documents.xsl`.
- implement a new *documentclass*

  1. Copy edited `documents.xsl` to `/usr/local/billmax/cfg/xsltfiles/html/newnamedocuments.xsl`.
  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 237

   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

Reprint Documents from BillMax

There are 3 options depending on circumstances to reprint documents from BillMax.

1. Choose **Correspondence** > **Combine PDFs** if you want to print all documents from a time period.

   a) Select **Virtual Company**.
   b) Enter the **Start Date** and **End Date**.
   c) Choose **Download PDF** and the file will be downloaded to your machine for local printing.

2. Choose **Correspondence** > **PDF Print Files** if you want to print created PDF files from a day within the last week. Generally used to print any batch files.

   a) Click **Get** on the date/time desired and the file will be downloaded to your machine for local printing.

3. Choose **Correspondence** > **Requeue Batch Documents** if you want to limit the documents to a particular delivery method or date.

   a) Choose **Refine Search** and add filters. Click **Apply**. Example below:

   b) Review the results and make adjustment if needed.
   c) Choose **Requeue Documents for Delivery**. The documents will print tonight.
   d) To print the documents immediately, choose **System Administration** > **Batch Processing**.
   e) Click the number for the "nightly" batch job.
   f) Click the number for "Send Billing Documents".

g) Click **Run Process** and the documents will be printed.

Resend Documents from BillMax

1. Choose **Correspondence** > **Requeue Batch Documents** to limit the documents to a particular delivery method or date.

2. Choose **Refine Search** and add filters. Click **Apply**. Example below:

3. Review the results and make adjustment if needed. Use the side bar showing how many in the list as a reasonableness check.

4. Choose **Requeue Documents for Delivery**. Filter on today's date with a Status of QUEUED to see the results. The documents will be sent tonight during batch processing.

5. To send the documents immediately, choose **System Administration** > **Batch Processing**.

6. Click the number for the "nightly" batch job.

7. Click the number for "Send Billing Documents".

8. Click **Run Process** and the documents will be sent.

Email Customers in Bulk from BillMax

The functionality allows for customers to be notified quickly by email using a chosen System Template. It was added in BillMax 2022 Q3.

1. Update or create a System Template.

   a) Select **Correspondence** > **System Templates**.

   a) Select **New** and enter the **Name** or click the System Template number.

   b) Enter **Description**.

   c) Enter **Template Class**.

   d) Enter **Template**. The following is an example for sending an maintenance notice.

> **Note:** HTML presentation may be sent as well as text presentation.

> **Note:** Variables in the form of ${*tablename.fieldname*} may be used in the template for the Account, User, and Virtual Company. $NAME is a manufactured variable that combines Company, First and Last Name from the Account. $*SERVICEADDRESS* is also a manufactured variable.

> **Note:** TO and FROM are required to be part of email template. The use of Subject is a best practice.

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

Routine maintenance will be conducted on FILLINDATE beginning
 at FILLINTIME.
We expect this to take FILLINLENGTH hours.  During this time you may
 experience
connectivity issues.

We apologize in advance for any inconvenience this may cause.

Thank you for your business!
<html>
  <body>
    <p>Routine maintenance will be conducted on FILLINDATE beginning
 at FILLINTIME.
We expect this to take FILLINLENGTH hours.  During this time you may
 experience
connectivity issues.</p>
<p>We apologize in advance for any inconvenience this may cause.</p>
<p>Thank you for your business.</p>
<p>${config.billingname}</p>  </body>
</html>
```

e) Select **Save** from the Context Menu.

f) 📝   **Note: Test** will only work for templates that are delivered from BillMax or a template beginning with 'operational' in the Name.

Select **Test** to a User for information and an email to send the template to. Click **Test** and verify the results until satisfied.

2. Send a **System Template**.

a) Select **Correspondence** > **Email Customers in Bulk**.

b) Enter **FROM Override** if email not coming from the Virtual Company Billing Email Address.

c) Enter **Send Immediately**, **Template Class**, and **Template** from above.

d) 📝   **Note:** The list of selections will be limited to the items that have an active Service.

In the **Combine** selections, choose one or more groups to receive the email. Some selections can be specified specifically. In the drop-down field, enter the first two characters for a list of selections. Choose one or more.

e) Restrict the email group further by specifying the **Account Class**, **Service Class**, **Virtual Company**, or **Account Profile**.

f) Click **Test** and the results of what would occur will be shown in a new window. Correct any errors and run the test again.

g) Go back to the original window and click **Send**.

h) If **Results** is clicked, the latest email results are displayed. To see previous results, click the **Options** tab and select the **Batch ID** with the email template and date sent.

Send An Email for Outages from BillMax

Notify customers on a particular tower, pop, or access point when an outage occurs or a maintenance window is planned.

1. Select **System Administration** > **POPs** or **System Administration** > **Access Points** .

2. Choose the **Number** of the **POP** or **Access Point**.

3. Choose the **Email Template** that needs to be sent.

4. Update the wording with the date or time as needed. Click Send and confirm.

⚠️   **Important:** Do not change the lines beginning with To or From. These are variables filled in by the BillMax system with appropriate values.

⚠️   **Important:** These Email Templates have a text and a html version of the same template in the window. Make sure to update both versions.

Send Payment Reminders from BillMax

Payment Reminders are emailed by Profile to Accounts with an outstanding balance a designated number of days before the due date.

1. Setup **Payment Reminders**.
   a) Select **Billing Administration** > **Account Profiles**.
   b) Select the **Account Profile**.
   c) In the Overdue Processing section, enter the number of days before the due date to send the reminder in **Payment Reminder**.
2. Update the Email template:
   a) Select **Correspondence** > **System Email Templates**.
   b) Select **paymentreminder**.
   c) 📝   **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.

          DUEDATE

            The date the balance is due.

          DAYS

            The number of days before the balance is due.

          BALANCE

            The monetary amount that is outstanding.

          config.*

            fields from the Virtual Company record to which the Account belongs. See *config Table (Virtual Company)* on page 237

Modify the **Template**.

Chapter

7

Email Processing

Topics:

- *Concepts*
- *How To*

Concepts

Background

In general, BillMax relies on the operating system's Mail Transfer Agent (MTA) for sending email. Depending on the configuration either *Sendmail* or *Postfix* is used. These systems can be configured to send email directly from the BillMax server or email may delivered through a SMTP relay (typically the mail server for the organization) server.

For performance reasons, direct use of a SMTP connection is not used by BillMax software for sending email. Instead BillMax relies on *Sendmail* and *Postfix* MTAs queuing mechanisms to submit batches of emails in an asynchronous manner.

As anti spam techniques have evolved, sending email properly to ensure delivery has become increasingly difficult to get right. There are quite a few requirements including proper DNS setup, SPF alignment, **DKIM** authentication, and **DMARC** reporting. In addition, content rules, IP/domain sending reputation and IP blacklisting also come into play.

Complaints about sent emails not being received by end-customers are almost always due to the email sending infrastructure is configured incorrectly in some way.

Configuration

How to configure your email sending infrastructure won't be covered in detail here. There are many sources for this information on the Internet. The following content does however provide some overall guidance and overview.

Depending on your version there are three separate systems for sending email:

1. *Sendmail*
2. *Postfix*
3. Amazon's Web Services (AWS) Simple Email Service (SES)

Sending Email from BillMax

Postfix is a Mail Transfer Agent (MTA). While other MTAs can be used (*sendmail* for example), *Postfix* is generally easiest to configure. It can be configured to send email directly from the BillMax server or through a SMTP relay..

Beginning with the Debian release of BillMax, the preferred and installed MTA is *Postfix*.

Regardless of which MTA is used, the Sendmail Path in **System Administration** > **Global Options** must be `/usr/sbin/sendmail` as this is an alias to the actual command used.

Configuring the MTA to use a SMTP relay or send email directly dictates what steps are needed to setup email sending. For example, reverse DNS lookup for the BillMax server is needed if sending email directly, but generally is not needed for SMTP relays.

Using a SMTP relay is preferred over direct sending as this technique utilizes a mail server that meets (hopefully) all the requirements for sending email. However, relays generally require authentication and thus the storage of email credentials on the BillMax server. This can be problematic administratively if the mail server requires password changes over time.

Email configuration for *Postfix* is done in the `/etc/postfix` directory. See *https://www.postfix.org/* for specific instructions.

Note regardless of email system used, it is not necessary to configure inbound email from external hosts for BillMax to operate. The BillMax ticketing system, if configured, uses the *fetchmail* program to receive such email.

AWS Simple Email Service (SES®)

Beginning with BillMax version 2022 Q3, BillMax can be configured to use AWS SES service to send email. Using AWS SES to send email has many advantages over sending email through an MTA and associated mail system and configuration. These include:

1. Greatly improved delivery success rate

2. Reduced administrative burden
3. Status Tracking (delivered, bounced, opened, …)
4. Scalable to 100's of thousand emails
5. Reputation management

With AWS SES, emails are more likely to be received by mail servers and not bounced or placed into SPAM folders. This is accomplished by proper setup of: DNS, SPF alignment, **DKIM** signatures. In addition, emails can be configured to originate with dedicated IP addresses so bad practices of other senders won't adversely affect your sending domain's reputation.

Administratively, there's less to figure out and maintain. For example, *SendMail* and *Postfix* can be configured to sign emails but the process doubles the amount of setup over that of non-signed emails. And this of course means more to maintain as well.

In addition to sending email through AWS SES®, BillMax can be configured to receive and process notices of various events related to the email sent. The events hold the status of the email delivery. Possible status values include but not limited to: Delivered, Bounced, Rejected, Opened.

Aside from proper configuration, AWS SES® also includes tools for managing your sending domain's reputation. For example, SES maintains suppression lists where previously problematic email addresses are quarantined instead of being sent to a mail server. Such techniques can help to improve your bounce rate and reputation, thus improving overall deliver-ability.

BillMax offers an optional service where BillMax will host and maintain the AWS SES services to send email for your BillMax instance. While your domain or sub-domain can be used in this service, the use of a BillMax domain for sending is recommended as this allows for complete outsourcing of the email sending infrastructure. Contact *sales@billmax.com* for full details.

How To

Configure BillMax to use AWS Simple Email Service (SES®)

Contact sales to purchase the BillMax Email Service or complete the following tasks to configure BillMax to use SES:

1. As root, down load and install AWS CLI. Consult *https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html*. Follow instructions for Linux (curl download). Then run the install command as described.
2. Configure the AWS CLI credentials. Obtain the credentials from BillMax support. Then edit the `/root/.aws/credentials` file by substituting *??????* with the provided values:

   - [default]
   -
     ```
     aws_access_key_id=??????
     ```
   -
     ```
     aws_secret_access_key=??????
     ```
   -
     ```
     region=??????
     ```

3. Edit the `/usr/local/billmax/local/billmax.conf` file. Set the value of BX_AWS_SES_CONFIGSET parameter to the value supplied by BillMax support.
4. In the staff interface, change the Global Configuration setting '*Path to the sendmail Program*' to '`/usr/local/billmax/bin/awsses.pl`'.
5. In the staff interface, for each Virtual Company, set the email addresses for contacts using the appropriate email address from the sending domain provided by BillMax support.

Once configuration is complete, validate a successful configuration by sending a statement to an email address you have access to. First confirm receipt of email. Then under the Account from which the statement was sent, view the Document History tab. The status of the delivery will be shown in the Delivery Status column of the document table.

Email delivery status for non document-based emails (for example overdue notice) can be seen on the Notes table on the Account.

> **Note:** Complete details on AWS SES are not provided here. See *https://aws.amazon.com/ses/* for details.

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 Reversals 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

Inventory Management

Topics:

- *Concepts*
- *How To*

Concepts

Inventory Management Overview

Inventory management is the process of overseeing and controlling the flow of goods and materials in a business, ensuring that the right quantity of the right products is available. In BillMax, Assets are tracked to assist in inventory management.

Assets are defined by Asset Definitions and displayed in the Assets list. Assets can be physical inventory items or IP Addresses and Assets are categorized by Class. Classes are used to link a group of assets to Billing Definitions. Available Classes are specified by the *'assetclasses'* list. Each Asset also has an Asset Key that is unique within the Asset Class. Available keys can be specified by the *'assetidentifiers'* list. There are several other fields available on the Asset Definition for informational purposes only at this time.

Assets are loaded into BillMax via a CSV file import. The CSV file's first record must contain the appropriate asset table column names. All columns are optional except the column defined as the Asset Key. Here is a list of available columns:

```
acquiredate  date (YYYY-MM-DD)
ponumber  char(30)
state  int
reason   int
digest   char(100)
serial   char(100)
macaddr  char(20)
ipaddr   char(64)
account  int
user   int
service  int
place   int
cogs   int
flags   int
filename  char(255)
contenttype  char(255)
batchid  char(16)
misc01-05  int
misc06-10 char(50)
```

How To

Manage IP Addresses

Setup to manage IP Addresses require several steps.

1. Create an **AP** with a Network/Netmask in CIDR format.

2. Create an **Asset Definition** for **IP Addresses**. **Assets** for **IP Addresses** associated with the **IP Address Asset Definition** are automatically generated.

3. On the **Package** or **Service Definition**, add the **IP Address** as a **Provisioning Variable**.

4. When the **User** is created, choose the **AP**.

5. When the **Package** or **Service** is created, choose the **IP**. The **IP** range is listed from the **AP** that the customer is attached to. Type in the first two numbers and a range of **IPs** will be listed to choose from.

6. The **IP Address** is removed from inventory and attached to the **Service**.

Assets Index

Menu

New
Refine Search
Clear Search
Spreadsheet
Print Friendly (PDF)
Paging
First
Next
Previous
Last

Page [1] out of 8
Records 1 - 35 of 262 shown

Option Sets

[ ] **?**

Save Options
Delete Option Set

Search Results

Number	Asset Key	Asset Definition	Status	Service	Access Point
294	192.168.195.127	IP addresses	AVAILABLE		AP 3
467	192.168.195.54	IP addresses	AVAILABLE		AP 3
396	192.168.195.219	IP addresses	AVAILABLE		AP 3
332	192.168.195.161	IP addresses	AVAILABLE		AP 3
435	192.168.195.254	IP addresses	AVAILABLE		AP 3
428	192.168.195.248	IP addresses	AVAILABLE		AP 3
364	192.168.195.190	IP addresses	AVAILABLE		AP 3
415	192.168.195.236	IP addresses	AVAILABLE		AP 3
351	192.168.195.179	IP addresses	AVAILABLE		AP 3
281	192.168.195.115	IP addresses	AVAILABLE		AP 3
480	192.168.195.66	IP addresses	AVAILABLE		AP 3
500	192.168.195.84	IP addresses	AVAILABLE		AP 3
383	192.168.195.207	IP addresses	AVAILABLE		AP 3
313	192.168.195.144	IP addresses	AVAILABLE		AP 3
448	192.168.195.37	IP addresses	AVAILABLE		AP 3
461	192.168.195.49	IP addresses	AVAILABLE		AP 3
402	192.168.195.224	IP addresses	AVAILABLE		AP 3
338	192.168.195.167	IP addresses	AVAILABLE		AP 3
268	192.168.195.103	IP addresses	Not Available	26	AP 3
493	192.168.195.78	IP addresses	AVAILABLE		AP 3
513	192.168.195.96	IP addresses	AVAILABLE		AP 3
434	192.168.195.253	IP addresses	AVAILABLE		AP 3
370	192.168.195.196	IP addresses	AVAILABLE		AP 3
300	192.168.195.132	IP addresses	AVAILABLE		AP 3
421	192.168.195.241	IP addresses	AVAILABLE		AP 3
357	192.168.195.184	IP addresses	AVAILABLE		AP 3
287	192.168.195.120	IP addresses	AVAILABLE		AP 3
474	192.168.195.60	IP addresses	AVAILABLE		AP 3
389	192.168.195.212	IP addresses	AVAILABLE		AP 3
325	192.168.195.155	IP addresses	AVAILABLE		AP 3
319	192.168.195.15	IP addresses	AVAILABLE		AP 3
442	192.168.195.31	IP addresses	AVAILABLE		AP 3
408	192.168.195.23	IP addresses	AVAILABLE		AP 3
344	192.168.195.172	IP addresses	AVAILABLE		AP 3
274	192.168.195.109	IP addresses	AVAILABLE		AP 3

Chapter

10

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 2: Payment Processing Settings

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

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 116.

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 121
*   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 121

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 120.

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 120

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 120.

One Time Payments

The staff and portal interfaces both support the notion of one time payments where the payment info is provided but not retained as a payment method or stored elsewhere. This option is available on a processor (more specifically a specific processor, *paytype* and *virtual company*) basis. The option is activated if *HasPointofSale* is set to 1 in a CC or echeck list.

In the staff interface, if activated, a staff member can use 'Payment > Process Credit Card|Electronic Check' even when no Payment method is on file:

The BillMax Customer Portal offers a similar option for customers.
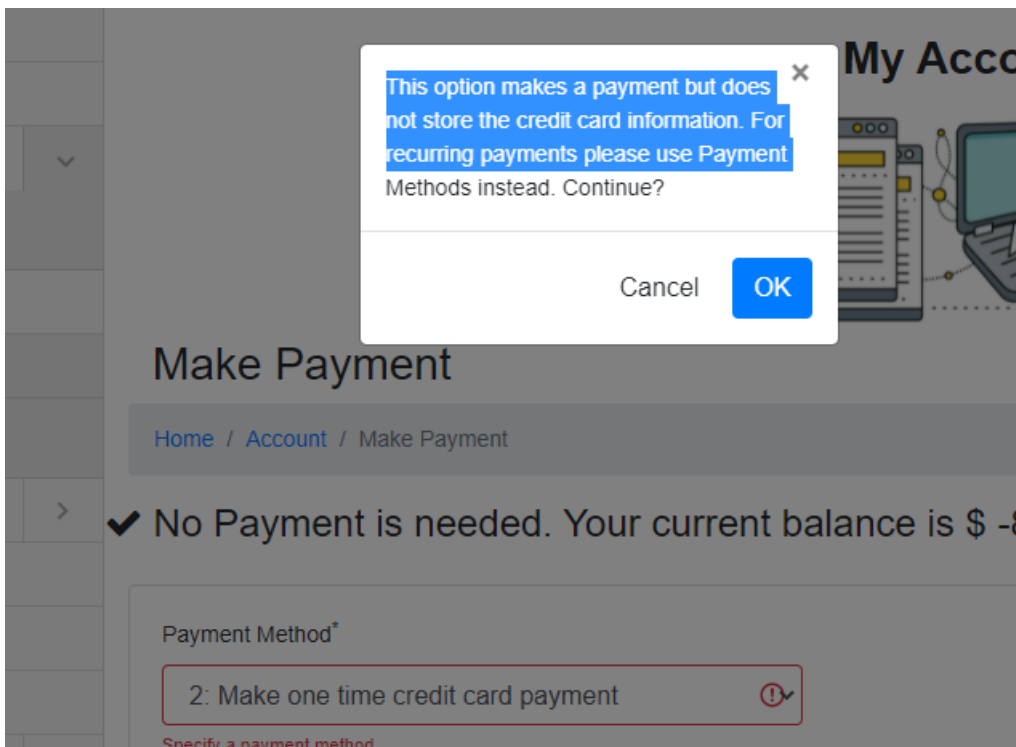
> ![note] **Note:** This option does write the encrypted card or bank account number to the transaction table (*efpstate*) but the information is removed at the end of the transaction.

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 119

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 119

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] **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.

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

Activate IPPay® Account Updater

Account Updater provides updated cards numbers and expiration dates from lost, stolen or reissued cards, to ensure uninterrupted recurring payments. Account Updater electronically retrieves the most current card holder information and updates the information on file. BillMax customers can sign up for the service to take advantage of these benefits.

1. Account Updater is a billable add on service at IPPay®. Complete the service addendum to sign up.
2. Receive the Merchant ID (different from the payment processing MID) and password from IPPay®. Update the *MerchantID* and *Password* entries on the IPPay list.
3. Change the *cardupdater* batch job to *Active*.

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.

   Enter **Ending Post Date** of the transactions for the deposit.
   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 120.

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 is it 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 120.
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 120.
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 119 or *Book Payments for Multiple Accounts* on page 119.
4. Optionally create a Billing Statement .
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 117.

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 3: IPPay e-check List entries

Item	Value	Description
DisableEcheckAutoPay	2	After two declines, disable auto-pay. When disabled, a ticket is created if the ticket option is used.
EcheckReturnFeeDefinition	0	Zero indicates no fee will be assessed. Otherwise, the value is the item definition number.
EcheckTicketQueue	2	Zero indicates no ticket will be created. Otherwise, the value is the ticket queue number.

Item	Value	Description
EcheckTicketProblem	3	Zero indicates no specific issue is set. Otherwise, the value is the queue issue number to associate with the created ticket.
EcheckRetryDays	3	Specifies the interval in days for the next auto-payment attempt.

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 Payments Using a Nacha File

How to book echeck payments and upload them to a bank via a Nacha file for processing.

1. > 📝 **Note:** When payments are created and applied in BillMax, BillMax assumes that the e-check payment is going to be good. The customer, if getting a billing statement or payment confirmation, will see a payment has been made.

   Create an echeck in BillMax manually *Manually Process an Electronic Check* on page 119 or from a scheduled payment via batch processing.

2. > 📝 **Note:** The payment entered makes an entry in the current transaction file. No money has been collected. No payment have been processed. Bookkeeping entries only have been made in BillMax.

   When ready to process payments, choose **Billing Administration** > **Create Bank EFP**.

3. Click **Create Nacha File** to close the current transaction file and create a file in Nacha format.
4. Choose the Bank Files tab and select the file to download.
5. Click **Download Selected**. Upload the file to the bank to have the payments processed.
6. If the bank report the payment not actually collected, *Book a Payment Reversal* on page 120

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 121.

3. Follow the step in *Manually Process a Credit Card Refund* on page 122 or *Manually Process an Electronic Check Refund* on page 122

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 58

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 67.

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 58

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 58. 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 121.

3. Follow the step in *Manually Process a Credit Card Refund* on page 122 or *Manually Process an Electronic Check Refund* on page 122

4. If necessary, reprocess the Credit Card or Electronic Check for the correct amount.

Chapter
11

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 329 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**.

3. Choose **Future Startdate** for **Reason**.

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.

Change Package or Service

To upgrade or downgrade a package or service for a customer, complete a service change.

1. Click the wrench beside the **Package** or **Service** number. Choose **Change Package** or **Change Service**.

2. 📝   **Note:** The **Resources** and **Service Type** on the definition must match for the product to be a choice in the drop down list of new definitions.

   Enter the **Change Date** and select the new definition.

3. Update any pricing or settings needed on the new definition.

4. Click **Save** for the change to occur on the date requested during batch processing. Click **Process Change** for the change to occur immediately.

The **Status** will remain as **Scheduled** until the change date has passed. When the change date arrives, the service change occurs during batch processing and the **Status** will be set to **Completed**. If the **Package** or **Service** is closed before the scheduled change date, then the **Status** is changed to **Cancelled**. The list of all changes is found on **Package Changes** tab on the **Package** or **Service.**

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 Amount**s are credited only to **Open** or **Suspended Account**s.
    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

12

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 134.
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.

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
- efpstate
- 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 136 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.

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 an 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 126, *Service Definition* on page 127

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 identify 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 135

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 pre-configured 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 does not 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).

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.
```
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:

```
#           ----- 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

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

Avoid Suspending a Service on Holidays

BillMax provides a list so that customers will not be suspended on entered holidays.

1. Login to the BillMax Staff Portal using an ID with Administrator privileges.
2. Select **Billing Administration** > **Lists**.
3. Select **nosuspendoverduedates**.
4. To add a list item, click **Add**.
5. Enter **Item** as the name of the holiday.
6. Enter **Value** as the date (YYYY-MM-DD).
7. Enter **Description** if more explanation is needed or desired.
8. Click **Save** or **Save and New**.

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:

   Table 4:

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 pre-configured 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 **System Administration** > **Attribute Groups**
   b) Click the Attribute Group number that needs provisioning or Select New
   c) Add an Attribute 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:

Attribute	Operator	Value	Type
Mikrotik-Rate-Limit	=	*2M/4M*	*reply*

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

Attribute	Operator	Value	Type
Mikrotik-Rate-Limit	=	*56k*	*reply*

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

id	groupname	attribute	op	value
3	2MB Up/4MB Down	Mikrotik-Rate-Limit	=	2M/4M
4	Suspended	Mikrotik-Rate-Limit	=	56k

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 5:

Field	Value	RADIUS Data
d01	*User Name*	*username*
d02	*Password*	*password*
d03	*Plan*	*plan identifier*

   - 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 134
   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`

```
                                       my $cmd = "$BILLMAXDIR/local/
bx_provision_radius.pl";
                                       $SERVICES{1}{name} = "radius";
                                       $SERVICES{1}{provision} = "yes";
                                       $SERVICES{1}{groupswitch} = "-g";
                                       $SERVICES{1}{passwordswitch} = "-
p";
                                       $SERVICES{1}{data} = [ "-s",
  "{NEW_state}", "-n", "{NEW_d01}",
```

```
                                              $SERVICES{1}{passwordswitch},
"{NEW_plaintextpass}",
                                              $SERVICES{1}{groupswitch},
"{NEW_d03}" ];
                                              $SERVICES{1}{cmd}{create} = "$cmd
-a create " ."
                                              $SERVICES{1}{passwordswitch} " .
escL($ENV{NEW_plaintextpass}) . "
                                              $SERVICES{1}{groupswitch} " .
escL($ENV{NEW_d03}) . " " " .
                                              escL($ENV{NEW_d01});
                                              $SERVICES{1}{cmd}{delete} = "$cmd
-a delete " .
                                              escL($ENV{NEW_d01});
                                              $SERVICES{1}{cmd}{update} = "$cmd
-a update";
                                              $SERVICES{1}{remotecmd} = "";
```

> **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:

Field	Value
User Name	*user*
Password	*p@ssw0Rd*

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

username	groupname	priority
user	2MB Up/4MB Down	1

   `SELECT * FROM radcheck WHERE username='user'` will result in:

id	username	attribute	op	value
4	user	Cleartext-Password	:=	p@ssw0Rd
5	user	Max-All-Session	:=	0

   b) Suspend the Service
   `SELECT * FROM radusergroup` should not show:

username	groupname	priority
user	Suspended	1

⚠️ **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.

Manage Asynchronous Hook Processing

1. Select **Billing Administration** > **List** from the Top Menu.
2. Select the **asynchooks** 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 (CentOS 7)*
2. Add the FreeRADIUS® Remote Application.
   a) Select **System Administration** > **Remote Applications**
   b) Select **New** and do the following:
      1. Enter the following:

Field	Value
Name	*RADIUS*
Description	*RADIUS*
Type	Select **FreeRADIUS**
Company	Select a company
Application Login Name	Specify a string that is unique between Remote Applications.
Client IP	*127.0.0.1*

      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

-

Field	Value
Description	Primary Server
Database Name	*billmax*
	⚠️ **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

Field	Value
Subscriber Username	*ping*
Subscriber Password	*w3e4r5t^y&u8*
Shared Secret	*testing123*
Type	*Authorization*

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:

Field	Value
SessionID	*RADIUS0123*
Elapsed Time	*33*
Elapsed Band Width	*123*

e) Select **Test** on the menu. if successful an `Accounting-Request` message will display

```
Sent Accounting-Request Id 193 from 0.0.0.0:51285 to 127.0.0.1:1813
  length 62
  User-Name = "ping"
  NAS-IP-Address = 127.0.0.1
  Acct-Session-Id = "RADIUS0123"
  Acct-Status-Type = Start
  Acct-Session-Time = 33
  Acct-Output-Octets = 123
Received Accounting-Response Id 193 from 127.0.0.1:1813 to 0.0.0.0:0
  length 20
```

Configure a Network Access Server (NAS)

1. Choose **System Administration** > **Network Access Servers**

2. Select **New**

3. Enter the following:

Field	Value
Name	IP address of NAS
Secret	key value specified in NAS and RADIUS configuration
RADIUS Column Type	Select *Terminal IP Address*
RADIUS Column Value	IP address of NAS
Resources	Select the RADIUS check box

> 📝 **Note:** The other NAS settings are rarely used. The Service Lookup fields are used to identify the service associated with the usage reflected in the Radius accounting request (e.g. the *datacall* table). The lookup provides a means to identify services where **Usernames** are not unique as many occur when multiple Radius non-synchronized servers are deployed.

4. Repeat for additional Network Access Servers
5. Select **System Administration** > **RADIUS Servers**
6. Select the applicable RADIUS server
7. Select **Reload Configuration**

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).

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:

```
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.

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**.

- RESOURCE 2
- Saisei

General

Number	2 ▾
Name	Saisei
Description	Saisei
Availability	Available ▾
Type	Not Applicable ▾
Sort order	0.0

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

Field	Value
Pop Required	Yes
Access Point Required	Yes
d01	IP Address
d02	
d03	Plan
	⚠ **Important:** On the **d03** row set **Constraint or List** to **'Saiseigroups'** and check **Uses List**.

e) Click **Save** and page will refresh. On the **d03** row, set **Default** to the appropriate plan and Click **Save**.

SERVICE DEFINITION 1
Saisei

Provisioning Information

			Resources	Radius ☐ ?
Tower Required	Yes ▼ ?			Saisei ✔
Access Point Required	Yes ▼ ?			
Provisioning Variable Conflict	Never OK ▼ ?			
Auto-Generate Password	No ▼ ?			
Save Plain Text Password	No ▼ ?			
Uses MAC Authentication	No ▼ ?			
Use User Name as Service Name	Use User Login ☐ ?			
	Use User Password as Service Password ☐ ?			
Service Class	Unspecified ▼ ?			

Provisioning Variables

Variable ?	Label ?	Asset Class ?	Constraint or List ?	Default ?	Required ?	Passwor
d01	IP Address	None ▼			☐	
d02 ?					☐	☐
d03	Plan	None ▼	Saiseigroups	gold1 ▼	☐	☐
d04		None ▼			☐	☐

4. Create credentials for API use in the Saisei application.

   a) Select **System** > **Adminstrators** 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 hast 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
13

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:

   - BillMax Customer Portal. See *Customer Portal* on page 243.
   - 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

Generate a Session ID

1. Login to the BillMax Staff Portal using an ID with Configuration privileges.
2. Select **System Administration** > **Remote Applications**.
3. Enter a **Name** and **Description**.
4. 📝   **Note:** To be able to generate a session id, the **Type** must be **Staff**.

   Enter **Type** and **Portal URL**. Choose the **Company** and **Customer Profile**.
5. Enter **Application Login Name** and **Application Password**.
6. Click **Generate Session ID** and use it as needed.

Chapter
14

Staff Portal

Topics:

- *Concepts*
- *How To*

Concepts

Lists

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.

General

- LIST DEFINITION 2
- adcodes

Menu

Save
Index
Other   >

General

Name	adcodes
Description	Values for the "Adcodes
Constraint	^[0-9]+$
Values must be unique in list	Yes
Sort Order	0.0
Enable Location	No
Remote Applications May Use List	No

Modify Permissions

Read	☐
Write	☐
Billing Read	☐
Billing Write	☐
Administration	☑

List Items

Add   Select   Toggle Portal Usage

Number	Item	Value	Description	Omit in Portal?	Discontinued?	Select List I
12	None	0		No		☐
13	Radio	1		No		☐
14	TV	2		No		☐
15	Magazine	3		No		☐
16	Newspaper	4		No		☐
17	Phonebook	5		No		☐
18	Billboard	6		No		☐
19	UseNET	7		No		☐
20	Search Engine	8		No		☐
21	Friend	9		No		☐
22	Web	10		No		☐
23	Seminars	11		No		☐
24	Direct Mail	12		No		☐

- a set of needed parameters for an operation such as payment

processing

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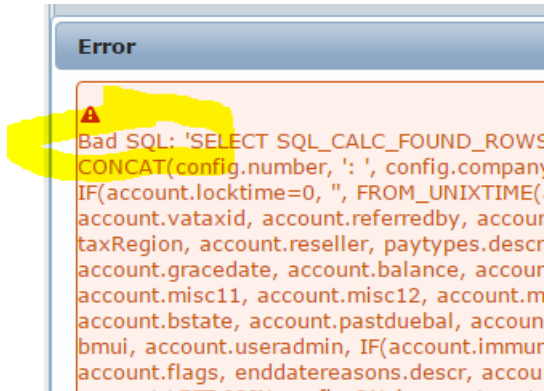
Although it looks ominous, simply click the **OK** and fix the Filter statements. Clicking the trashcan icon deletes a statement.

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 Setsare 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 only be shared among other users of BillMax if they are set to "Public". Go to **My Items** > **Options Sets** to find the setting. See the video "How to Conduct a Refine Search" on *http://www.billmax.com* under **Resources** > > **Video Tutorials** or the link below.

.

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. Note: Closing the Package or Service must handled manually. Since billing cannot continue unless the Service is being provided the assumption is that provisioning is manual (i.e. suspended only affects billing not the service itself) and therefore cessation of billing will be manual as well.

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

Sales Return When Opening [No Sales Return ▼] (?)

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

Date to Reopen/Resume Billing [ ]/[ ]/[ ] 📅 (?)

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

General

Re-Open Users/Services [No ▼] (?)
Billing Catch Up Days
Date to Resume Billing [ ]/[ ]/[ ] 📅 (?)
Sales Return for non-Open Period [No Sales Return ▼] (?)

[Continue] [Cancel]

Status Open

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>
```

becomes

```
<li>
    <label for="accountMisc01">Address Line</label>
    <select id="accountMisc01" name="misc01">
        <option value="1"
 @TEST(##Fmisc01#,1,selected="selected",)>Yes</option>
        <option value="0"
 @TEST(##Fmisc01#,0,selected="selected",)>No</option>
    </select>
</li>
```

Changing which Data are Required

Customize your system to edit required fields for the Staff Portal.

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 prepending with the "#" symbol.

Making the Account Second Address Line Required

Change

```
required.push('#accountCfname');
required.push('#accountClname');
required.push('#accountAddr1');
required.push('#accountCity');
required.push('#accountZip');
```

to

```
required.push('#accountCfname');
required.push('#accountClname');
required.push('#accountAddr1');
required.push('#accountCity');
required.push('#accountZip');
required.push('#accountAddr2');
```

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 **searchs** List is in view.
3. Choose the **searchs** List.
4. Chose 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&sortingt`
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**.
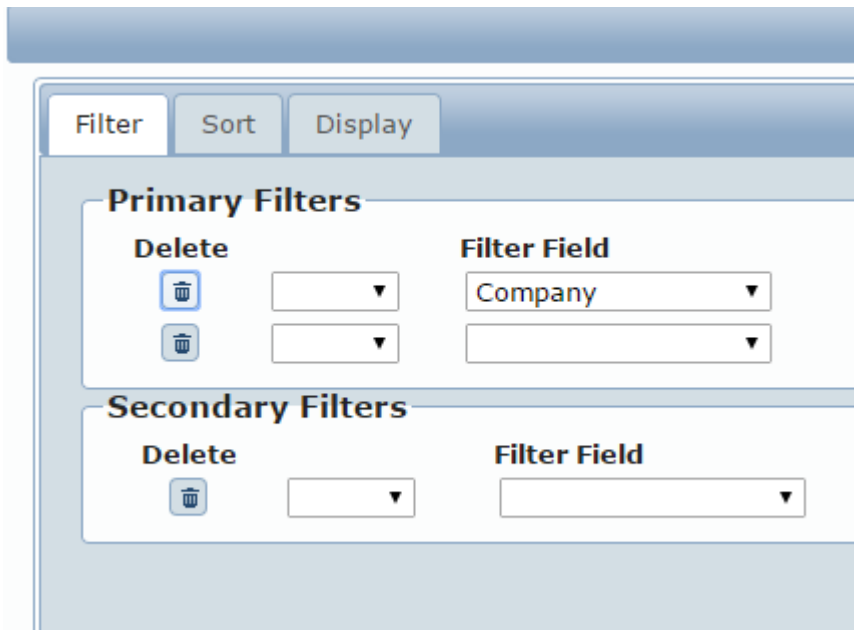
Add Staff Portal Validation

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 157

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
15

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 OFF-SITE!!!** 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 ANSII 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.

⚠ **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 157
- *How To* on page 89 (Customer Communications)
- *Modify a Server-side Binary* on page 170
- *Automated Processing Overview* on page 46

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 trough 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 trough 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 number 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 171.

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*.

The following elements are set to PCI Data Security Standards 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 173. 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.

Permissions

Permissions are entries in the permission table. These entries represent all activities available in the BillMax system. Each activity is associated with one or more **authlevels** that can perform that activity. An **Authorized User** is assigned certain **Authorization Levels** to be able to perform the activities they need.

   ⚠ **Caution:** It is strongly recommended that **Permission** changes are tested on a development server and involved support from the BillMax support team. A mistake can easily result in loss of access to the interface or the ability to run programs.

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 - Configuration
- 32 - Billing Administration
- 64 - Super User

Each Authorized User needs a minimum of Read access to login into the BillMax system. The first 5 levels build on one another. Here is an example of what each level can do. It is not an exhaustive list.

Table 6: Auth Level access

	Access
Read	Login
	See Full Account

	See Billing Transactions
	See Payment Transaction data
	See Electronic Transaction data
	See Document History but not individual items
	See User
	See Tickets
	See some Lists
	Access Help Menu items
Write	Add/Edit Account
	Suspend Account
	Add/Edit Account, User, Service Notes
	Add/Edit Payment Method
	Process Payments
	Add/Edit Users
	Add/Edit Services
	Add/Edit Tickets
	Add/Edit Appointments
	Add/Edit Employee Schedules
	Add an Option Set
	Customize Display Settings (My Items)
Billing Read	Add/Edit Service Definitions
	Add/Edit Service Definition Notes
	Add/Edit Package Definitions (except Package Elements)
	Run Reports except FCC and CDR Billing
Billing Write	Add Package Elements
	Schedule/Process Price Changes
	Add/Edit Bundles, One Time Items, Recurring Fees
	Add/Edit Promotions
	Add/Edit Invoice Fees
	Add/Edit Sales Items
	Add/Edit Invoices and Billing Statements
	Add/Edit Sales Returns
	Add/Edit Refunds
	Add/Edit Service Level Agreements
	Run FCC and CDR Billing Reports

Configuration	Add/Edit Virtual Companies
	Add/Edit Account Profiles
	Add/Edit Lists
	Add/Edit Federal USF reporting
	Add/Edit Tax/Fee Items
	Add/Edit Tax Group Items
	Add/Edit Issue Classes
	Add/Edit Ticketing Queues
	Add/Edit Batch Processing
	Add/Edit POPs/Access Points
	Add/Edit Authorized Users

Billing Administrator is a place holder to create access based on special needs. Currently the Super User level is reserved for special use cases for BillMax support.

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.

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.

How To

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

```
Machine Information
BillMax Host Information. Fri Mar 11 09:01:57
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 inet6: error fetching interface
information: Device not found
```

is highlighted.

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

   ```
   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
   inet6: error fetching interface information: Device not found
   ```
   highlighted.

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

Permissions

Add Authorization Level to Permissions

1. Login to the BillMax Staff Portal using an ID with Super User privileges.
2. Select **System Administration** > **Permissions**.
3. Choose **Refine Search** and add a filter based on the **Object** or **Source** if the **Object** is blank.
4. Click the number of the **Permission** to change.
5. Select the check box for the **authlevel** that has been added to change or an existing one to add the permission.
6. Click **Save**.

Add an Authorization Level

1. Login to the BillMax Staff Portal using an ID with Super User privileges.
2. Select **System Administration** > **Authorization Levels**.
3. To add a list item, click **Add** under **List Items**.
4. Enter **Item** as the drop-down choice.
5. Enter **Description** if more explanation is needed or desired.
6. Click **Save**.

Set Minimum Permission to Login and Logout

These steps will implement an Authorization Level to login and logout and land on the main page and be able to do nothing else.

1. Login to the BillMax Staff Portal using an ID with Super User privileges.
2. Select **System Administration** > **Permissions**.
3. Click the number for these Permission items.

Search Results

Select	Number	Source	Object	Status	Activity	Permission	Suggested	Description	Permission (bit)
☐	3343	bin/licenseinfo/licenseinfo.c		Active	u5xKy7O7zaINawvOmg0OMFS0Cnk71L	Billing Read,myauth2	Billing Read		68
☐	4066	cgi-bin/ua/ua.c	login	Active	VIEW FORM	Read,myauth2	Read		65
☐	4067	cgi-bin/ua/ua.c	logout	Active	VIEW FORM	Read,myauth2	Read		65
☐	4070	cgi-bin/ua/ua.c	main	Active	ALERTS	Read,myauth2	Read		65
☐	4071	cgi-bin/ua/ua.c	main	Active	VIEW FORM	Read,myauth2	Read		65

4. Check the **authlevel permission** that was created.
5. Click **Save**.

See Add Permissions to add additional actions to the **authlevel** as needed.

Add Permissions
Create a Log File

1. Login to the BillMax Staff Portal using an ID with Super User privileges.
2. Select **System Administration** > **Authorization Levels**.
3. To add a list item, click **Add** under **List Items**.
4. Enter **Item** as the drop-down choice.
5. Enter **Description** if more explanation is needed or desired.
6. Click **Save**.

Update Permissions to Add Actions

1. Login to the BillMax Staff Portal using an ID with Super User privileges.
2. Select **System Administration** > **Permissions**.
3. 📒   **Note:** Consider each item with a Permission > 1 if each Authorized User has a foundation of Read access. Otherwise, all items must be updated.

    View the log file to determine items to change.

4. Add the new **Authorization Level** to the **Permission** for each **Source** or **Object** listed. To update the **Permission**
    a) Choose **Refine Search** and add a filter based on the **Object** or **Source** if the **Object** is blank.
    b) Click the number of the **Permission** to change.
    c) Select the check box for the **Authorization Level** that has been added to change or an existing one to add the permission and click **Save**.

> Example: Adding the ability to edit Tickets to a User with Read Permissions and a new *authlevel* of Tickets would look like the following:
>
> **Search Results**
>
> | Select | Number | Source | Object | Status | Activity | Permission | Suggested | Description |
> |--------|--------|--------|--------|--------|----------|------------|-----------|-------------|
> | ☐ | 285 | cgi-bin/enter_ticket/enter_ticket.c | | Active | OVERRIDE TICKETING CHECK | Administration | Administration | |
> | ☐ | 286 | cgi-bin/enter_ticket/enter_ticket.c | | Active | RUN PROGRAM | Write,Tickets | Write | |
> | ☐ | 287 | cgi-bin/enter_ticketprob/enter_ticketprob.c | | Active | Ibw93qNBZOpfQoeuctpLEMdH956G2h | Write,Tickets | Write | |
> | ☐ | 288 | cgi-bin/enter_ticketres/enter_ticketres.c | | Active | neCQnj90CdRKokrFzLBISW26YPbcUY | Write,Tickets | Write | |
> | ☐ | 289 | cgi-bin/enter_ticketsubscriber/enter_ticketsubscriber.c | | Active | OVERRIDE SUBSCRIBER CHECK | Administration | Administration | |
> | ☐ | 290 | cgi-bin/enter_ticketsubscriber/enter_ticketsubscriber.c | | Active | RUN PROGRAM | Write,Tickets | Write | |

Update Permissions to Limit Actions

1. Login to the BillMax Staff Portal using an ID with Super User privileges.
2. Select **System Administration** > **Permissions**.
3. Add an **Authorization Level**.
4. Add the new **authlevel** to the **Permission** for each **Source** or **Object** that need the **Permission** leaving off those that the **Permission** has been taken away. To update the **Permission**
    a) Choose **Refine Search** and add a filter based on the **Object** or **Source** if the **Object** is blank.
    b) Click the number of the **Permission** to change.
    c) Select the check box for the **Authorization Level** that has been added to change or an existing one to add the permission and click **Save**.

> Example: Add an **Authorization Level** *NoCreditCards*. Add *NoCreditCards* to all items that have Write skipping the Sources and Objects that deal with credit cards.

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 212

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.

1. Register for your own USPS account at *https://www.usps.com/business/web-tools-apis/welcome.htm*
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** > **Manage Edge Process.** .
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**
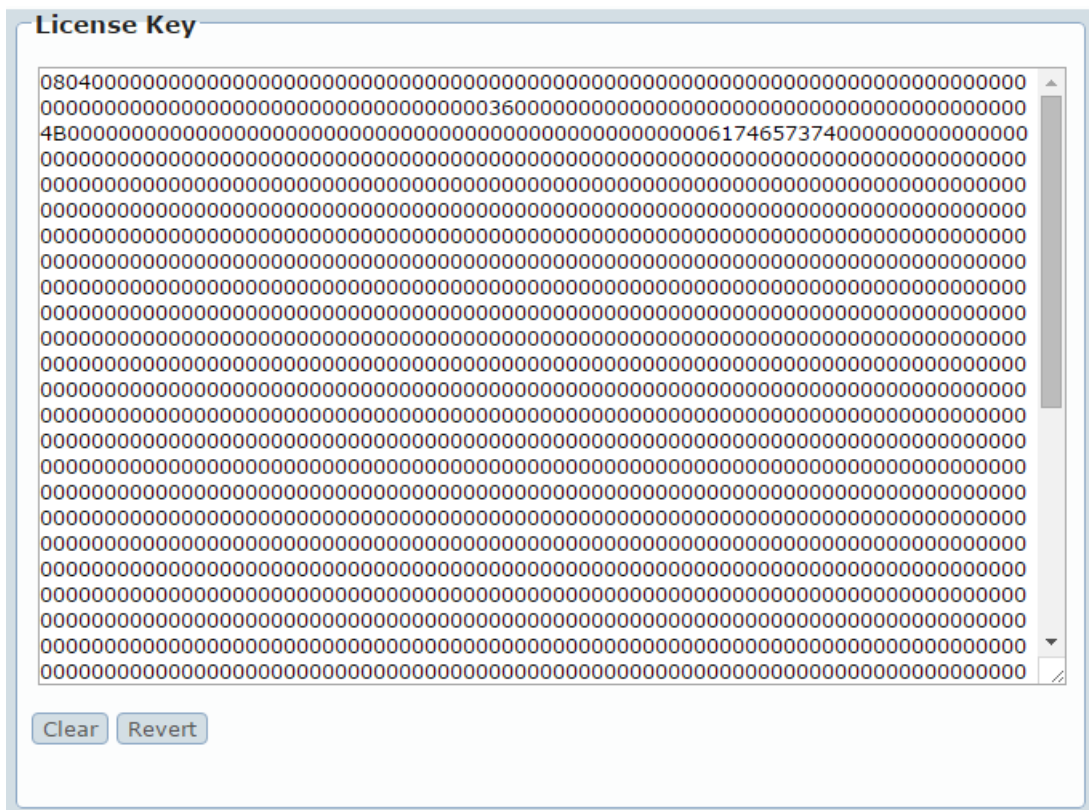
4. Click **Clear**.
5. Paste the license into the **License Key** field.

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 166

1. Select **System Administration** > **Security Setting**.
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 on a Debian Server

1. Download the BillMax Debian package.
   a) Using a web browser, go to *http://www.billmax.com*.
   b) Click **LOGIN**. Contact support@billmax.com for the user name and password if you don't know it.
   c) Select **Files** in the Main menu and identify the appropriate DEB file.
   d) Alongside each file listing are two links. To download the file to your device (for eventual transfer to the BillMax host), click on the first link (it has a downwards pointed arrow).
   e) Otherwise, to get the file location used in a `wget` command that can be used on the BillMax host to download the file directly, click the second link (rectangular box). This action copies the command to the clipboard. Paste the command and execute it in the command line on the BillMax host as the root user. The command must be used within several minutes of the selection in the portal.
2. 📝 **Note:** Example- `systemctl stop apache2 bx_edge mariadb radiusd`

   Login as root and stop all BillMax related services.
3. Make backups. At a minimum `/usr/local/billmax`, `/opt/billmax`, `/var/www/mysql` must be backed up. See *Backups, Redundancy and Archiving* on page 163
4. Making sure that no one is using the system, execute `systemctl start mariadb apache2` to start the web server and database.
5. Commit any local edits. As root execute:

a) `cd /usr/local/billmax`

b) `svn commit -m "commit information"`Update BillMax Debian package (the file must include a path)

6. To update BillMax Debian package (the file must include a path), execute `apt upgrade ./<file name of BillMax Deb package>`

7. Run the root upgrade scripts. As root execute:

a) `cd /opt/billmax/install`

b) `./BxInstall.pl`

c) Supply "n" if you see "Continue printing (Y/n)?

d) Supply "F" if you see "Enter two indexes for each column to rename, (R)elist, or (F)inish:"

e) Solve any inconsistencies. There should not be any if no customizations have been made. After resolving an inconsistency, you may need to issue the command resolved filename where filename is the name of the file that had the inconsistency.

f) ./BxUroot-4.0.pl

8. To run the billmax upgrade script, execute:

a) `cd /opt/billmax/install`

b) ./BxUpost-4.0.pl

9. To verify the installation, execute

a) Execute `/usr/local/billmax/bin/dbck`

b) Login into the system and conduct various task to ensure the system is operational.

c) If fall back is needed, restore the system from the backups in step 3.

Chapter
16

Taxes

Topics:

Avalara® Integration

Concepts

Integration Overview

Avalara® is a Software As A Service(SAAS) tool for tax calculations. The BillMax Avalara integration provides taxes for BillMax financial transactions. BillMax sends the transaction information along with a T/S pair and a place code. The T/S pair identifies the transaction type and service type to tie to Avalara product mappings. The place code is the location of the User/Service location where billing will occur. The tax is returned from Avalara.

This integration requires an agreement with Avalara that includes the necessary **username**, **password**, and **Client Id** to use Avalara's API calls and is available only in BillMax 4.0.8800 or versions released since. A new license key with Avalara enabled is required.

Note: As testing and implementation begins, if an invoice is posted without the tax service and then the tax service engine is enabled, the invoice will not be able to be deleted. The converse is also true.

Creating pending transactions under a tax service engine and then disabling the tax service engine may keep the invoice from being able to be posted. The transactions will need to be fixed or regenerated. The converse is also true.

How To

Configure Taxes Using the Avalara® Integration
Taxes are charged based on a **P Code** derived from the User's location and the **T/S Pair** of the item or service.

1. To begin *Set Avalara Parameters* on page 177 .
2. *Configure Avalara T/S Pairs* on page 176 for each package or service that needs to be assessed a tax.
3. *Set Virtual Company Parameters* on page 178.
4. *Set Place Codes for Users* on page 177.
5. City taxes can be affected if a User is Inside or Outside the City Limits. The default comes from Avalara but can

be overridden on the User. Make changes as necessary.
6. *Update Pending Transactions* on page 179.
7. Review *Setup for Accounts for Avalara* on page 178 and *Setup an Avalara Tax Exemption* on page 179 to complete your setup.

Configure Avalara® T/S Pairs

1. Login to the BillMax Staff Portal using an ID with Configuration privileges.
2. Select **System Administration** > **Avalara Tax Service** > **Download T/S Mappings** and choose **All Products** or **Products without T/S Pairs**. Select **Spreadsheet**.
3. 📝 **Note:** . See *https://knowledge.avalara.com/bundle/qvv1656594440497/page/mapping_resources.html#swl1658957553261* for the Avalara worksheet.

> 📝 **Note:** See *https://knowledge.avalara.com/bundle/qvv1656594440497/page/ Transactions_and_Service_Types.html#pmv1658957553197* for mapping pairs.

Cut and paste the columns representing the SKU, Product Name, and Product Description into the Avalara Product Mapping Spreadsheet columns that match.

4. 📝 **Note:** The T/S Pair (0,0) is for Packages and Services with no tax.

   📝 **Note:** To upload the file into BillMax, it must be saved as a CSV file.

   Populate the **Transaction Type** and the **Service Type** columns and save file.

5. Select **System Administration** > **Avalara Tax Service** > **Upload T/S Mappings**.

6. Choose the saved file and click **Upload Files**.

7. If Late Fees are assessed go to **System Administration** > **T/S Pairs**. Click the **T/S Number** and select the **T/S Pair** for **Late Fee Transaction/Service Pair** and enter a description to be used in billing. Click **Save**.

Configure Packages with All-in-One Services

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

2. There are 3 options to generate correct taxing:

   - Create a T/S pair at Avalara designating the percentage of recurring amount to one of their T/S pairs. For example, if you have $200 for VOIP/Internet where in your view $150 is VOIP and $50 is Internet, you can create T/S Pair 1000/1000 that is 19/6 - 75%, 5/6 - 25%.
   - Make use of Packages. There is a setting that rolls up the billings underneath, so create a $0.00 Package with priced Services and with appropriate T/S Pairs set on each.
   - Break into different Services with appropriate pricing and T/S Pairs on each.

Set Avalara® Parameters

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

2. Select **System Administration** > **Avalara Tax Service** > **General**.

3. 📝 **Note:** During the time of transition, taxing will continue with the internal BillMax taxing module so tax settings are present as well as the Avalara settings for setup. When the Tax Engine is Active, the internal BillMax settings will not be seen.

   Enter *Not Active* for **Tax Engine** to begin Setup. Enter *Active* when the system is ready to be tested or go live.

4. Enter **Username**,**Password**, **Client ID** provided by your Avalara agreement.

5. Enter *Sandbox* for **End Point** when testing begins until the system is ready to "Go Live" and then enter *Production*.

6. Enter *Disabled* for **Taxing** during setup. Enter *Enabled* to begin testing.

7. Enter the values for other fields as needed and click **Save**.

Set Place Codes for Users

1. Run **System Administration** > **Batch Processing** > **avalarageobatchapi** to retrieve a **Place Code** for all User/ Service locations.

2. 📝 **Note:** When a **Place Code** is retrieved, the **County** and **Country** are also retrieved and updated on the **User** in BillMax.

   After the initial run, set the batch process to run nightly for new **Users/Service Locations. P Codes** are also refreshed monthly 2 days before billing occurs.

3. 📝 **Note:** Closed Users are ignored. Only errors are reported.

Check **System Administration** > **Log Files** > **Other** > **avalara_xxxx-xx-xx.log** for errors. *xxxx* representing a date.

4. Run **System Administration** > **User Place Codes** for a list of **Users** with **Place Codes** missing.

5. Check to see if the **User** has a valid **Longitude** and **Latitude** or a valid address to get a valid **Place Code**.

Set Virtual Company Parameters

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

2. Select **Billing Administration** > **Virtual Companies** and choose one.

3. The **Client Profile ID** should remain zero unless you have created a profile at Avalara. Profiles are used to enable customizations for a request to the AFC tax engine.

4. Enter the **Company Identifier**. This field identifies the **Virtual Company** to Avalara if you use more than one for reporting purposes.

5. The next several fields establish the type of company that is billing with this **Virtual Company**. Enter the appropriate answers.

6. 📝 **Note:** Taxes can be presented to your customer by category or individual tax. Categories might show as Sales and Use Taxes, Connectivity Charges, and Regulatory Charges. Default is By Category.

   Select the **Tax Presentation to Customer**.

7. Enter **Lat/Long Required for Place Code**. The default is No.

8. Select the reporting mechanism for VoIP and Cellular.

9. Click **Save**.

Setup for Accounts for Avalara
A few additional settings on **Accounts** are needed to use Avalara effectively.

1. In the **Taxes/Fees** display box on the **General** tab, choose the **By Invoice**. Default is **Single Invoice** and it is recommended unless taxing does not seem correct and then try **By Individual Transactions** so tax brackets, caps, and thresholds are applied at the individual line item.

2. Choose **Tax Designation**. The default comes from the Account Profile setting with the same name.

3. Choose the **User** for the Place Code for the Account.

4. Add any items not passed to the customer for Avalara by setting **Tax/Fee No Pass Through**.

5. Click **Save**.

Setup for Discounts for Avalara
Discounts are available at the **Account**, **User** or **Package/Service** level. The *Discount Type* is required for each discount if using Avalara.

1. 📝 **Note:** We are not tax experts. Discuss settings with Avalara representative.

   In the **Discounts** display box , choose a **Discount Type**. Default is **None** if no Discount is entered.

   - Retail Product - discount from normal price to get the sale
   - Manufacturer Product - discount given if a kickback is to be received from manufacturer
   - Subsidized - ACP, Lifeline, etc.
   - Good Will - trade for services, possibly discount for outage

2. Click **Save**.

Setup an 800 Number

1. Set the **T/S Pair** as 19/6 on the **Package/Service Definition**.

2.

Prices Include Tax | No

T/S Pair | 19/6 VoIP/Access Cl

T/S Pair Lines/PBX Trunk/Toll Free DID | 19/635 VoIP/Toll-Fre

T/S Pair PBX Outbound Channel | Not Set

T/S Pair PBX Extension | Not Set

Add 19/635 to objects that have a DID specified.

3. Create a **CDR Billing Plan** that does both outbound/inbound minutes to use on the **Additional Usage Billing Module** connected on the **Package/Service Definition**.

Setup an Avalara Tax Exemption

1. Bring up an **Account** and click the **Add Avalara Tax Exemption** tab.

2. Choose the **Status** as **Active**.

3. Choose the **Place Code** for the location that the exemption applies.

4. 📝 **Note:** A Tax Exemption can be for a category or a particular tax type. Only one selection is allowed per exemption.

Choose the **Category Id** if the location is exemption for all taxes and fees of that category.

5. Choose the **Tax Type** if the location is exempted for a particular tax. See *https://knowledge.avalara.com/bundle/ qvv1656594440497/page/AFC_Tax_Type.html* for more information.

6. Select a combination of one or more values for **Scope** of the tax exemption. **Scope** determine where the exemption is to be applied.

7. Select the **Domain**. The **Domain** is the taxing jurisdiction that determines when an exemption is applied.

8. The **Type** is defaulted to **Tax Examption**. Some **Accounts** may have negotiated a tax or fee removed. If the tax or fee must still be paid by your company, mark the **Type** as **No Pass Through**. The item will not be displayed on the customer's bill.

9. **Exempt Non-billable** is defaulted to No. If non-billable items should be exempted for this location, change it to Yes.

10. **Force Level Exemptable** is defaulted to No. For more information contact Avalara.

Update Pending Transactions

1. Select **System Administration** > **Pending Transactions**.

2. Select **Run Report** to see what transactions exist in the system that need a T/S pair during the transition period.

3. Select **Update Pending Transactions Using Product Definitions** to have BillMax update all transactions it can based on the T/S pair of the Production Definition it is associated with.

4. Select **Run Report** again and see if you have any remaining transactions. These transaction may have issues with missing the User Place Code or the T/S pair. Add the T/S pairs on the definitions and check the User for a valid address.

5. ⚠ **Notice:**

If desired, as a convenience feature, **Update Pending Transactions Set To 0/0** may be used. It will set all Pending Sales/Credits not associated with a Product Definition (i.e. General Sales) to the 0/0 T/S pair, if the 0/0 T/S pair is in the system.

Run Report again until you have no transactions listed.

Internal BillMax Taxing Module

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

Setting up taxes for Sales Tax, VoIP Tax and Internet Tax

Assume the following Tax Items:

• State Sales Tax
• County Sales Tax
• City Sales Tax
• State VoIP Tax
• County Internet Tax
• USF Tax

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 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.

Tax Specification on the Package or Service

A product may be deemed to be subject to taxes or not or Not Applicable. Not Applicable is for revenue or locations (Oregon which has no sales tax is an example), where the sales tax code is not applicable. No tax is assessed and no record is made in the tax detail table. This revenue is intended to not be included when reporting sales taxes.

In Texas, you report both taxable and non-taxable sales, but not all transactions a company may book in BillMax are subject to the sales tax code, so it is a mechanism to know the amounts to exclude from the filing.

It differs from the Taxable - No setting in that the revenue for this may be included in the filings depending on local requirements. Revenue is booked in the *taxdetail* table.

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

1. Select **Billing Administration** > **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%.

Add Tax Exemptions

Depending on whether the customer is exempt from all taxes or an individual tax, will determine the setup.

1. For a customer to be exempt from all taxes:
   a) Bring up an Account.
   b) On the General tab under Taxes/Fees, choose **No** for **Taxable**.
   c) Enter **Tax Exempt ID** provided by the customer.
   d) Click **Files** tab and click **Add Files**.
   e) Click **Choose Files** to select the tax exempt certificate provided by the customer and click **Open**.
   f) Click **Upload Files** to save a copy of the certificate with the Account.
2. For a customer to be exempt from a single tax:
   a) Bring up an Account.
   b) Click **Tax Exemptions** tab and click **Add Tax Exemption**.
   c) Choose the **Tax Group**.
   d) Choose the **Charge Type** and **Reason**.
   e) Click the checkbox for each **Tax Item** that should be exempt and click **Save**.

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 183 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 183 for each tax or fee that needs to be assessed.

3. Repeat *Add A Tax Group* on page 182 for each combination of **Tax Class**, **Tax Region**, and **Tax Items** needed. For an example, see *Tax Groups* on page 180

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 to alter the amount of the Sale to be taxed, a *Tax Rule* can be created to be used on individual **Service** or **Package Definitions.**. See *Tax Rules* on page 182.

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**.

3. Select **Run Report** from Context Menu.

Chapter

17

Reference

Topics:

API

REST APIs

Our BillMax Core REST API allows seamless integration with our billing solutions, offering developers a standardized way to access and manipulate customer data, billing information, and service provisioning. From the *billmax.com* website, explore our Swagger page to interact with our API endpoints in a user-friendly environment, where you can test requests, view responses, and understand how to integrate our robust billing functionalities into your applications effortlessly.

Instructions for setup can be found on your server at: `/usr/local/billmax/restApis`.

Currently there are two supported modes using the REST API. These are controlled by the USER_AUTH variable in the `/usr/local/billmax/restApis/billmaxCoreApi/v1/.env` file.

Currently supported are

```
USER_AUTH="NONE"
```

- only the API key is needed, all POST/PATCH actions are logged as "billmax".

```
USER_AUTH="USERPASS"
```

- the API key is needed. In addition, the */login* endpoint is used to pass back a *BXSESSIONID*, and this value should be passed in the http header as a Bearer token for subsequent calls to the API. Example:

```
api_key: asdf
Content-Type: application/json
Authorization: Bearer 14741E3548AF080F59A18B64BC5AE1E8
```

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.

Name	Type	Description	Required
srchnumber	Integer	If non zero, the number of an existing Account.	Yes
number	Integer	The number to use for the Account. Recommended to use **0** for a new Account.	Yes
state	Integer	Status of the Account.	Yes if a new Account.

Name	Type	Description	Required
		0	
		Open	
		1	
		Closed	
		2	
		Suspended	
		4	
		Collections	
startdate	Date		Yes if a new Account
enddate	Date		
adcode	Integer	Advertising Code	
		0 or a value from the *adcodes* List.	
cancelreason	Integer	Cancel Reason	
		0 or a value from the *cancelreasons* List.	
referredby	String		
cfname	String	First Name	
cmname	String	Middle Name	
clname	String	Last Name	
company	String		
cphone	String	Phone Number	
fphone	String	Fax Number	
email	String		
textmessage	String	Email address equivalent for a cell phone.	
taxable	Integer	**-1**	Yes if a new Account.
		Tax Not Applicable	
		0	
		Tax Exempt for all taxes	
		1	
		Taxable	
taxregion	Integer	**-1**	Yes if a new Account.
		Tax Region Not Applicable	
		>0	
		A reference to the number field in the *taxregion* table.	
reseller	String		

Name	Type	Description	Required

paytype	Integer	Default Payment Method
		
		1
		   Cash/Check
		2
		   Credit
		3
		   Electronic Check
		4
		   EFT
		5
		   Credit Once, then change to Cash/Check
			Yes if a new Account.

invmethod	String	Automated Billing Methods. This is contextual depending on the Account Profile. A "0" indicates no automated method is selected. Otherwise, a comma separated list such as "1,2,4,16..." 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.	Yes if a new Account.
addr1	String		
addr2	String		
addr3	String		
addr4	String		
city	String		
statename	String		
zip	String		
country	String		
county	String		
invday	Integer	Billing/Prorate Day of Month. If 0, then the setting from the Virtual Company will be used.	
gracedate	Date	Date an overdue Account has until Overdue processing is started.	
misc01	Integer		

Name	Type	Description	Required
misc02	Integer		
misc03	Integer		
misc04	Integer		
misc05	Integer		
misc06	String		
misc07	String		
misc08	String		
misc09	String		
misc10	String		
misc11	Date		
misc12	Date		
misc13	Date		
misc14	Date		
misc15	Date		
misc16	Time		
misc17	Time		
misc18	Time		
misc19	Time		
misc20	Time		
customerof	Integer	Virtual Company. A reference to the number field in the *config* table.	Yes if a new Account.

generation	Time stamp	**Blank**
		  Used for new Account
		YYYY-MM-DD HH:MM:SS
		  The time stamp of the existing Account record.
			Yes

reason	Integer	The reason an Account is not open. 0 or a value from the *reasons* List.	
title	Integer	0 or a value from the *titles* List.	

bmui	Integer	Portal Access.
		0
		  Portal Access disallowed.
		1
		  Portal Access allowed.	

Name	Type	Description	Required

useradmin	Integer	Account Administrative Portal User
		
		0
		    No User specified
		-1
		    Set to the first User created under this Account
		>1
		    An existing User of this Account.	

immune	String	Comma separate list of values, each value a power of 2.
		
		1
		    Immune from automated suspension
		2
		    Immune from Late Fees.
		4
		    Will not receive overdue emails	

profile	Integer	Account Profile. A reference to the number field in the *profile* table.	
vataxid	String	Value Added Tax ID	
datetoreopen	Date	If an account is not Open, the date in the future to reopen the account automatically.	

flags	String	Comma separate list of values, each value a power of 2.
		
		1
		    Issue Sales Returns for services billed in the future when Closing the Account.
		4
		    Issue Deposit Returns when Closing the Account.
		8
		    Reopen Users and Services that where	

Name	Type	Description	Required
		Suspended or Closed when the Account was Suspended or Closed.	
		16	
		Issue Sales Returns for period billed but Account was Suspended or Closed.	
enddatereason	Integer	Reason that the End Date has been set.	
		4	
		The End Date was set because the Account is closing. Normal Suspend then Close will take place.	
		8	
		The End Date was set at the customer's request. The Account will be Suspended indefinitely until further action is taken.	
autoeft	Integer	Automated Payment Processing modifiers.	
		Comma separate list of values, each value a power of 2.	
		1	
		Disable all automate Collections and Refunds.	
		2	
		Disable Automated Refunds	
		4	
		Allow one Automated Refund, then disable.	
importid	String	If imported from another billing system, the key in the old system for the account	
pctdiscount	Decimal	Percent discount that all non-Usage Sales to the Account will get. Implemented as a Store Credit for each Sale.	

Name	Type	Description	Required
		Supply "2.5" for a 2.5 percent discount.	
pctdiscountexpdate	Date	Date that the discount expires.	
usagepctdiscount	Decimal	Percent discount that all Usage Sales to the Account will get. Implemented as a Store Credit for each Sale.	
usagepctdiscountexpdate	Date	Date that the Usage discount expires.	
loginid	String	Portal Login Name for the Administrative User.	
password	String	Portal Password for the Administrative User.	
guarantor	Integer	Another Account that will pay the bills for this Account. If non-zero, a reference to the number field in the *account* table.	
timezone	Integer	Value from the *timezones* List.	

enter_saleorcredit

Used to create or modify a Pending Sale or Store Credit Transaction.

Usage

```
/usr/local/billmax/bin/enter_saleorcredit [options]
```

Options

Option	Notes
-s service	Service number
-c chargeType	Not Applicable for a Service or Item Sale. A value from the List *chargetypes*.
-m rate	Does not have to be a monetary amount. Can be fractional.
-d 'description'	
-a account	Account Number
-q quantity	
-M amount	The amount of the transaction. Generally should equal rate multiplied by quantity rounded to a
-T taxable	Not applicable for a Service Sale. **0** Not Taxable **1** Taxable

Option	Notes
	-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.

Name	Type	Description	Required
srchnumber	Integer	If non zero, the number of an existing Service.	Yes
number	Integer	The number to use for the Service. Recommended to use **0** for a new Service.	Yes
override	Integer	Used to change a Service number. Use of this variable is not recommended. **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*.	
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.	
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.

Name	Type	Description	Required
		0	
		Open	
		1	
		Closed	
		2	
		Suspended	
		3	
		Wait for Payment	
		5	
		Prospective	
startdate	Date		Yes if a new Service
enddate	Date		
servdef	Integer	Number of the Service or Package Definition of the underlying Product of the Service or Package.	Yes if a new Service
package	Integer	Number of the Package to which the Service is associated.	
service	Integer	Number of the Service to which this Service is associated. Only specify if this Service is a Service Fee	

free	Integer	Usage is not recommended. If a Service is marked Free, no billing activity is recorded. A better method is to give a 100% discount.
		
		0
		   Service is not free.
		1
		   Service is free.	

taxable	Integer	Used in conjunction with the User Tax Region to assess taxes.
		
		-1
		   Tax Not Applicable
		0
		   Tax Exempt for all taxes
		1
		   Taxable
			Yes if a new Service.

Name	Type	Description	Required

wsetup	Integer	**0**
		    Assess the Setup amount immediately
		1
		    Do not assess the Setup amount
		2
		    Assess the Setup amount when the Service is opened for the first time.	

descr	String	Description. May be displayed on Invoice.	
d01	String	Primary identifier for a Service.	
copypw	Integer	If non-zero, the number of a Service from which to copy the password and plain text password fields to the current Service.	
d02	String	A value that is encrypted when stored. Typically used for passwords.	
d03	String		
d04	String		
d05	String		
d06	String		
d07	String		
d08	String		
d09	String		
d10	String		
invdate	Date	Next Billing Date for a Service.	Yes if a new Service.
nextusagedate	Date	Next Usage Billing Date for a Service.	Yes if a new Service.
pop	Integer	Point of Presence 0 or a value from the *popregions* List.	
iprice	Money	First Price	
iduration	Integer	First Duration. A value from the *durations* List.	
icnt	Integer	Number of times to assess First Price. Must be greater	

Name	Type	Description	Required
		that 0. "1" is suitable for a single recurring price.	
price	Money	Second Price	
duration	Integer	Second Duration. A value from the *durations* List.	
cnt	Integer	Number of times to assess Second Price. If 0, Second Price and Third Price not used.	
lprice	Money	Third Price	
lduration	Integer	Third Duration. A value from the *durations* List.	
lcnt	Integer	Number of times to assess Third Price. If 0, Third Price not used.	
setup	Money	Setup amount	
setup_taxable	Integer	Used in conjunction with the User Tax Region to assess taxes. **-1** Tax Not Applicable **0** Tax Exempt for all taxes **1** Taxable	
ponumber	String	Purchase Order Number	
ou_notice	Float	Used for metered billing.	
ou_susp	Float	Used for metered billing.	
ouprime_notice	Float	Used for metered billing.	
ouprime_susp	Float	Used for metered billing.	
refaccount	Integer	The number of an existing referring Account.	
misc01	Integer		
misc02	Integer		
misc03	Integer		
misc04	Integer		
misc05	Integer		
misc06	String		
misc07	String		
misc08	String		

Name	Type	Description	Required
misc09	String		
misc10	String		
misc11	Date		
misc12	Date		
misc13	Date		
misc14	Date		
misc15	Date		
misc16	Time		
misc17	Time		
misc18	Time		
misc19	Time		
misc20	Time		
generation	Time stamp	**Blank** — Used for new Account **YYYY-MM-DD HH:MM:SS** — The time stamp of the existing Account record.	Yes
reason	Integer	The reason an Service is not open. 0 or a value from the *reasons* List.	
wdeposit	Integer	**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.	
quantity	Float	Quantity to bill.	
quantityfree	Float	Portion of Quantity that is not subject to billing.	
cyclept	Integer	Point in a billing cycle. Used for cyclical billing.	
taxrule	Integer	-1 or a number reference a Tax Rule from the *taxrule* table.	
salesrep	Integer	Deprecated.	

Name	Type	Description	Required
datetoreopen	Date	If a Service is not Open, the date in the future to reopen the Service automatically.	
flags	String	Comma separate list of values, each value a power of 2.	
		1	
		Issue Sales Returns for Services/Packages billed in the future when Closing the Service.	
		4	
		Issue Deposit Returns when Closing the Service/Package.	
		8	
		Reopen Services of Packages that where Suspended or Closed when the Package/ was Suspended or Closed.	
		16	
		Issue Sales Returns for period billed but Account was Suspended or Closed.	
		64	
		Bill a following full term if the first billing is prorated.	
		256	
		Create a Package, not a Service.	
		1024	
		Display zero priced usage sales on Invoices	
		2048	
		Ignore Discounts	
		4096	
		Automatically issue Store Credits against recurring Sales	
		8192	
		Automatically issue Store Credit against usage Sales	

Name	Type	Description	Required
		131072	
		Bill one full term. Used when billing in Prospective State.	
		524288	
		One term was billed.	
enddatereason	Integer	Reason that the End Date has been set.	
		4	
		The End Date was set because the Service is closing. Normal Suspend then Close will take place.	
		8	
		The End Date was set at the customer's request. The Service will be Suspended indefinitely until further action is taken.	
executehook	Integer	**0**	
		Do not execute the Service Hook program.	
		1	
		Execute the Service Hook program.	
importid	String	If imported from another billing system, the key in the old system for the Service	
pctdiscount	Decimal	Percent discount that all non-Usage Sales to the Account will get. Implemented as a Store Credit for each Sale. Supply "2.5" for a 2.5 percent discount.	
pctdiscountexpdate	Date	Date that the discount expires.	
usagepctdiscount	Decimal	Percent discount that all Usage Sales to the Account will get. Implemented as a Store Credit for each Sale.	
usagepctdiscountexpdate	Date	Date that the Usage discount expires.	

Name	Type	Description	Required
discountamount	Money	A fixed amount for automatically generating a Store Credit.	
cardid	String	Used for Prepaid Services.	
pin	String	Used for Prepaid Services	
vcarddef	Integer	Used for Prepaid Services	
contractenddate	Date	Date representing the date the Contract for the Service expires.	
locationuser	Integer	Number for user table representing alternative Location for the Service from User.	
salesagent	String	Comma separated numbers representing Authorized Users for whom should included in commission reports for this Service	
etfee	Money	Early termination fee.	
dobilling	Integer	**0** Do not billing the Service immediately. **1** Bill the Service immediately. This is the default.	

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:

Variable	Value
FORM_number	0
FORM_action	SAVE
FORM_paytype	2
FORM_account	Account Number
FORM_fname	First name
FORM_mname	Middle name
FORM_lname	Last name
FORM_company	Company
FORM_addr1	Address
FORM_city	City
FORM_statename	State
FORM_zip	Zip/Postal code
FORM_country	Country
FORM_phone	Phone number
FORM_email	Email address

FORM_batchmethod	**0**
	   Credit card added to Account but not used for automated payments
	 2
	   Credit card added to Account and used for automated payments.

FORM_ccnum	Credit Card number
FORM_cvnum	CVV number
FORM_ccexp_month	Expiration month
FORM_ccexp_year	Expiration year - 4 digit

FORM_recurnotok	**0**
	   Credit card
	 4
	   Prepaid/Gift Card

FORM_generation	blank

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

```
<SOAP-ENV:Body>
 <ns:edgeLogin>
  <protocol>3</protocol>
  <user>asdf</user>
  <password>asdf</password>
 </ns:edgeLogin>
</SOAP-ENV:Body>
```

Name	Type	Description	Required
protocol	Integer	Value must be 3	Yes
user	String	*Username* of the RAE	Yes
password	String	*Password* of the RAE	Yes

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`.

```
<SOAP-ENV:Body>
 <ns:edgeAppInfoStatus>
  <code>0</code>
  <message></message>
  <info>
   <number>0</number>
   <name></name>
   <sessionid></sessionid>
   <salesview></salesview>
   <defdomain></defdomain>
   <hotspoturl></hotspoturl>
   <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>
```

Name	Type	Description
code	Integer	**0** Success

Name	Type	Description
message	String	
info.number	Integer	Number of RAE
info.name	String	Name of RAE
info.sessionid	String	The current token used to access the edge application for the RAE. Use in subsequent calls to edge.
info.salesview	String	*Salesview* of RAE
info.defdomain	String	*Default domain* of the Virtual Company
info.hotspoturl	String	*Hot spot URL* of RAE
info.phone	String	*Phone* of RAE
info.email	String	*Email* of RAE
info.company	Integer	*Company* of RAE
info.profile	Integer	*Profile* of RAE
info.interface	Integer	*Type* of RAE
info.companyname	String	Not in use
info.wwwurl	String	Not in use
info.logourl	String	Not in use
info.flags	Integer	*Flags* from RAE
info.locsym	String	Local monetary symbol from Global Options

getLists
Retrieve allowable List data from BillMax

Input

```
<SOAP-ENV:Body>
  <ns:getLists>
    <appsessionid>myappsessionid</appsessionid>
  </ns:getLists>
</SOAP-ENV:Body>
```

Name	Type	Description	Required
appsessionid	String	info.sessionid returned from `edgeLogin`	Yes

Output

```
<SOAP-ENV:Body>
  <ns:listStatus>
    <code>0</code>
    <message>Success</message>
    <list>
```

```
          <elems>
            <list>listname</list>
            <descr>description</descr>
            <value>value</value>
            <multi>0</multi>
            <explanation>explanation</explanation>
            <flags>1</flags>
          <elems>
          .
          .
          .
        </list>
      </ns:listStatus>
    </SOAP-ENV:Body>
```

Name	Type	Description
code	String	**0** Success
message	String	
list	Container for one or more *elems*	
elems.list	String	Name of BillMax List to which this entry belongs.
elems.descr	String	Description of List Item
elems.value	String	Value of List Item - Integer or String depending on context.
elems.multi	Integer	**0** If List is not *invmethods* **1** If List is *invmethods*
elems.explanation	String	Explanation of List Item
elems.flags	Integer	Flags value of List entry

Typical Result for *titles* List

```
<elems>
   <list>titles</list>
   <descr>Mr</descr>
   <value>1</value>
   <multi>0</multi>
   <explanation></explanation>
   <flags>1</flags>
</elems>
<elems>
   <list>titles</list>
   <descr>Mrs</descr>
   <value>2</value>
   <multi>0</multi>
   <explanation></explanation>
   <flags>1</flags>
</elems>
<elems>
   <list>titles</list>
   <descr>Ms</descr>
```

```
      <value>3</value>
      <multi>0</multi>
      <explanation></explanation>
      <flags>1</flags>
    </elems>
    <elems>
      <list>titles</list>
      <descr>Dr</descr>
      <value>4</value>
      <multi>0</multi>
      <explanation></explanation>
      <flags>1</flags>
    </elems>
```

Staff Interface

The `edge` staff interface provides access to BillMax data by way of authenticating an Authorized User using the *auth* table.

⚠ **Important:** An Authorized User must have **Remote Applications/Staff** checked to use this interface.

⚠ **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 203.
2. Log the Authorized User into BillMax with *authLogin* on page 206.
3. To get data, use *staffGetRecord* on page 207.
4. To put data, use *staffPutRecord* on page 210.

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

```
<SOAP-ENV:Body>
 <ns:authLogin>
  <appsessionid></appsessionid>
  <loginid></loginid>
  <password></password>
 </ns:authLogin>
</SOAP-ENV:Body>
```

Name	Type	Description	Required
appsessionid	String	info.sessionid returned from `edgeLogin`	Yes
loginid	String	Authorized User Login Id	Yes
password	String	Authorizes User Password	Yes

Output

```
<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>
```

Name	Type	Description
code	String	**0** Success
message	String	
info.loginid	String	Authorized User Login Id
info.fname	String	Authorized User First Name
info.lname	String	Authorized User Last Name
info.auth	Integer	Authorized User Number
info.sessionid	String	Session Id for the Authorized User. Use in subsequent requests for the Authorized User.

staffGetRecord
Used to retrieve data from a table in BillMax.

Input

```
<SOAP-ENV:Body>
 <ns:staffGetRecord>
  <appsessionid></appsessionid>
  <authsessionid></authsessionid>
  <table></table>
  <number>0</number>
  <form></form>
 </ns:staffGetRecord>
</SOAP-ENV:Body>
```

Table 7:

Name	Type	Description	Required
appsessionid	String	info.sessionid from `edgeLogin`	Yes
authsessionid	String	info.sessionid from `authLogin`	Yes

Name	Type	Description	Required
table	String	Name of a supported table in BillMax	The following tables are supported: • *account* • *user* • *service* • *servdef* • *config* • *profile* • *efpdata* • *ticket*
number	Integer	number field for table in BillMax	**0** Return a zeroed dataset for the following: • *account* • *user* • *service* • *efpdata* • *ticket* **> 0** Return the record that matches this value.
form	String	Associated Form value from RAE Portal data.	By default the following are supported but may be locally customized: ***account*** 　*account* table ***user*** 　*user* table ***service*** 　*service* table ***efpdata*** 　*efpdata* table

Output

```
<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>
```

Name	Type	Description
code	String	**0** Success
message	String	
data	Container	Contains one or more *elems*
data.elems.name	*TABLE.field*	Each name is table name in caps followed by a period followed by field name.
data.elems.list	String	Name of BillMax List if the field uses a List as source of options.
data.elems.length	Integer	Field length
data.elems.stype	Integer	**1** string **2** pointer **3** integer **4** float **5** double **6** special
data.elems.dtype	Integer	Internal Use Only
data.elems.permissions	Integer	Permissions as specified by the RAE Portal data for the form. A bit mask: **1** Read **2** Write

Name	Type	Description
data.elems.cdata	String	Present if elems.stype is 1,2 or 6
data.elems.idata	Integer	Present if elems.stype is 3
data.elems.fdata	Float	Present if elems.stype is 4
data.elems.ddata	Double	Present if elems.stype is 5
data.elems.message	String	Populated by `staffPutRecord` is create or update for specified field denied.
data.elems.flags	Integer	Internal Use Only

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.

```
<SOAP-ENV:Body>
 <ns:staffPutRecord>
  <appsessionid></appsessionid>
  <authsessionid></authsessionid>
  <create>0</create>
  <table></table>
  <form></form>
  <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>
```

Name	Type	Description	Required
appsessionid	String	info.sessionid from `edgeLogin`	Yes

Name	Type	Description	Required
authsessionid	String	info.sessionid from `authLogin`	Yes
create	Integer	Denotes whether a new record is being created	**0** Update existing record **1** Create new record
table	String	Name of a supported table in BillMax	The following tables are supported: • *account* • *user* • *service* • *efpdata* • *ticket*
form	String	Associated Form value from RAE Portal data.	By default the following are supported but may be locally customized: ***account*** *account* table ***user*** *user* table ***service*** *service* table ***efpdata*** *efpdata* table
data	Container	Contains one or more *elems*	
data.elems.name	*TABLE.field*	Each name is table name in caps followed by a period followed by field name.	
data.elems.stype	Integer	Specifies the type of data being passed	**1** string **2** pointer **3** integer **4** float **5** double **6** special
data.elems.cdata	String	Present if elems.stype is 1,2 or 6	

Name	Type	Description	Required
data.elems.idata	Integer	Present if elems.stype is 3	
data.elems.fdata	Float	Present if elems.stype is 4	
data.elems.ddata	Double	Present if elems.stype is 5	
data.elems.message	String	Text if update or create not allowed. Returned.	
data.elems.flags	Integer	Internal Use Only	

Output

See result of *staffGetRecord* on page 207.

Customer Interface

The edge customer interface provide access to BillMax by way of authenticating a Customer using the *user* table.

Database Checking (`dbck`) Error codes

Error codes generate by `dbck`. Will most likely be noticed in the BillMax nightly log.

Table 8: Error Code Explanations

Code	Information or Error	Explanation
10003	Informational	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.
10022	Informational	The account is referencing an Account Administrator that no longer exists. Update to the correct user or select None.
11500	Informational	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 *support@billmax.com* for recovery. Access to the system will be needed by BillMax Support.
13222	Error	The message indicates a bug in processing. Please contact *support@billmax.com*
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;`

Code	Information or Error	Explanation
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 9: Database Conventions

Name	Description
number	Unique Identifier
entdate	The date the record was created
enttime	The time the record was created.
generation	The date and time the record was last modified. Current value in database must be submitted on update.
empl	The user that created the record. A reference to *auth.number*.
lastmodempl	The user that last modified the record. A reference to *auth.number*
state	The status of a record: closed; open; suspended; active etc.

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.

Name	Description	Type
adcode	Advertising Code. Used in Advertising report. List *adcodes*.	*tinyint(4)*
addr1	Address Line 1	*char(50)*
addr2		*char(50)*
addr3		*char(50)*
addr4		*char(50)*

Name	Description	Type
appointmentroute	Default Appointment Route. Used with Scheduling. List *appointmentroutes*.	*int(11)*
autoeft		*int(11)*
bacct	Deprecated.	
balance	Account balance.	*bigint(20)*
bcity	Deprecated.	

bmui	Customer Portal access:
	
	1
	    Yes
	0
	    No
		tinyint(4)

bname	Deprecated.	
brout	Deprecated.	
bstate	Deprecated.	
cancelreasons	Reason an Account canceled. List *cancelreasons*	*mediumint(8)*
ccaddr	Deprecated.	
cccity	Deprecated.	
cccountry	Deprecated.	
ccdeclines	Deprecated.	
ccexp	Deprecated.	
ccfname	Deprecated.	
cclname	Deprecated.	
ccname	Deprecated.	
ccnexttry	Deprecated.	
ccnum	Deprecated.	
ccnumenc	Deprecated.	
ccpass	Deprecated.	
ccphone	Deprecated.	
ccpin	Deprecated.	
ccstatename	Deprecated.	
cctype	Deprecated.	
cczip	Deprecated.	
cfname	First Name	
city		*char(30)*

Name	Description	Type
clname	Last Name	*char(30)*
cmname	Middle Name	*char(30)*
company	Customer's company.	*char(30)*
country		*char(30)*
county		*char(30)*
cphone	Phone	*char(30)*
customerof	Virtual Company	*int(11)*
datetoreopen	Date in the future to automatically reopen an Account.	*date*
deposit	Total deposits held.	*bigint(20)*
email		*char(100)*
empl		*mediumint(8)*
enddate	Date Account Closed or Date to Close. The date after end of service.	*date*
enddatereason	Reason for closing. List *reasons*	*tinyint(4)*
entdate		*date*
enttime		*time*
finactivity	Internal Use.	
flags	Internal Use.	
fphone	Fax.	*char(40)*
generation		*timestamp*
gracedate	Date through which reminders for overdue balances will not be sent and automated suspension will not occur.	*date*
guarantor	Account number of an Account in BillMax that will pay outstanding Debits for the current Account.	*int(11)*

immune	Bit Field. The following are bit values:
	1
	    Do not automatically Suspend
	2
	    Do not assess Late Fees
	4
	    Do not send Overdue Emails
		tinyint(4)

importid		*char(100)*
invday	Billing/Prorate Day of Month	*tinyint(4)*
invmethod	Billing Statement/Invoicing Methods. A bit mask that references	*int(11)*

Name	Description	Type
	First, Second, … from the Account Profile Document Delivery Methods	
lastmodempl		*int(11)*
latedate		*date*
lockowner		*mediumint(8)*
locktime		*int(11)*
misc01	Integer Field through misc05	*int(11)*
misc02		*int(11)*
misc03		*int(11)*
misc04		*int(11)*
misc05		*int(11)*
misc06	Character (50) Field through misc10	*char(50)*
misc07		*char(50)*
misc08		*char(50)*
misc09		*char(50)*
misc10		*char(50)*
misc11	Date Field through misc15	*date*
misc12		*date*
misc13		*date*
misc14		*date*
misc15		*date*
misc16	Time Field through misc20	*time*
misc17		*time*
misc18		*time*
misc19		*time*
misc20		*time*
nlatenotes		*tinyint(4)*
number		*int(11)*
pastduebal	Past Due Balance	*bigint(20)*
payerid		*char(50)*

paytype	Default Payment Method:
	1
	    Cash/Check
	2
	    Credit Card
		tinyint(4)

Name	Description	Type
	3 Electronic Check **4** EFT **5** Credit Card Once, then Cash/Check	
pctdiscount	Percent Discount for all Sales to this Account.	*decimal(10,6)*
pctdiscountexpdate	Date the Discount expires.	*date*
popupnote	Reference to record in *notes* table that is displayed when an Account is accessed in the Staff Portal.	*int(11)*
profile	Account Profile	*mediumint(8)*
purchases	Sales amounts owed	*bigint(20)*
reason	Reason why Account is not Open. List *reasons*	*mediumint(8)*
referredby	Referral Code	*char(30)*
reseller	Reseller ID	*char(30)*
sla	Service Level Agreement	*int(11)*
startdate		*date*
state	Status of the Account: **0** Open **1** Closed **2** Suspended **4** Collections	*tinyint(4)*
statename	State or Province	*char(30)*
summarydate	Last date a Billing Statement was generated.	*date*
suspdate	Date the Account was Suspended.	*date*
taxable	**1** Account is subject to paying taxes. **0** Account is not subject to paying taxes. Examples are non-profit and government organizations.	*tinyint(4)*

Name	Description	Type

taxregion	**-1**
	    Not Applicable. Only use if never collecting any type of tax.
	otherwise
	    Foreign Key to *taxregion.number*.
		int(11)

textmessage	Number to use to send text messages. If not integrated with SMS service, use email address for text message device.	*char(100)*
timezone	List *timezones*.	*tinyint(4)*
title	Salutation. List *titles*.	*tinyint(4)*
ucash	Amount of unused Payments.	*bigint(20)*
ucredit	Amount of unused Store Credit.	*bigint(20)*
usagepctdiscount	Percentage discount for Usage Sales.	*decimal(10,6)*
usagepctdiscountexpdate	Date percentage discount for usage sales expires.	*date*
useradmin	User authorized to make Account adjustment through the Customer Portal.	*int(11)*
vataxid	Value Added Tax ID or customer.	*char(15)*

verifiedaddr	Verification results if using Address Verification:
	0
	    Not Verified
	1
	    Verified
	2
	    Address Verification in use, but address saved even though Address Verification failed.
		int(11)

zip	ZIP/Postal Code	*char(30)*

servdef Table (Product Definitions [Package/Service, Recurring Fee, One Time, Bundle])

Due to historical reasons, the meanings for *descr* and *name* are reversed. *descr* is the **Name** and *name* is the **Billing Description**. Pricing is in cents.

Name	Description
number	
entdate	
enttime	

Name	Description
empl	
lastmodempl	
sortorder	
state	Availability • 0 - Available • 1 - Discontinued - No non-Closed Services reference the Definition • 2 - No New Users - not available to be added.
name	Billing Display
descr	Name
startdate	The first date a Product will be available for Sale.
enddate	
price	Second Price
setup	Set up Price
taxable	• 1 - taxable • 0 - not taxable • -1 - not applicable
renewable	• 0 - Bill one set of applicable First, Second and Third prices through time. • 1 - Bill one set of applicable First Second and Third prices through time. When one set finished, continue to bill last applicable price. • 2 - Bill one set of applicable First, Second and Third prices, repeat cycle in definitely.
duration	Second Duration - positive integer for months, negative integer for days
resources	Resources (using in Provisioning)
d01	Customizable Provisioning Field Label
d02	Customizable Provisioning Field Label
d03	Customizable Provisioning Field Label
d04	Customizable Provisioning Field Label
d05	Customizable Provisioning Field Label
d06	Customizable Provisioning Field Label
d07	Customizable Provisioning Field Label
d08	Customizable Provisioning Field Label
d09	Customizable Provisioning Field Label
d10	Customizable Provisioning Field Label
cd01	Constraint or List of d01

Name	Description
cd02	not used
cd03	Constraint or List of d03
cd04	Constraint or List of d04
cd05	Constraint or List of d05
cd06	Constraint or List of d06
cd07	Constraint or List of d07
cd08	Constraint or List of d08
cd09	Constraint or List of d09
cd10	Constraint or List of d10
def02	not used
def03	Default for d03
def04	Default for d04
def05	Default for d05
def06	Default for d06
def07	Default for d07
def08	Default for d08
def09	Default for d09
def10	Default for d10
ou_notice	
ou_susp	
ouprime_notice	
ouprime_susp	
basehours	
hourlyprice	
charge_type	Generally used for designated Accounting bucket.
iprice	First Price
refamt	Referral Amount
misc01	
misc02	
misc03	
misc04	
serviceclass	Generally used for Provisioning.
company	deprecated
tierplan	Usage Pricing Plan

Name	Description
hourlyprice_peak	deprecated
basehours_peak	deprecated
setup_taxable	• 1 - taxable • 0 - not taxable • -1 - not applicable
deposit	Money held for Deposit
taxrule	Tax rule table
iduration	First Duration
nosync	Do not prorate dates/amounts
prepaid	Denotes that this is a prepaid service definition. Managed using Prepaid Cards
d01conflict	Manages name space of Package/Service *d01* fields where a name space is defined by Resources and Service Definitions that have a common Resource. • 0 - All *d01* values in the name space for non-closed Services must be unique. • 1 - *d01* values in the name space are not checked for uniqueness. • 2 - *d01* values within a User may be duplicated, but must be unique in the name space if belonging to different Users
pwautogen	• 0 - do not auto-generate a password • 1 - auto-generate a password
free	Denotes is a Service is free. Free is defined as no billing activity for either recurring or usage billing. From an accounting perspective, this is different than billing while giving a 100% discount. The latter has associated billing activity. • 0 - not free • 1 - free
icnt	Number of times to bill First Price (*iprice*). Must be non-zero.
billinarrears	deprecated
useplaintextpass	Store the submitted password (Service/Package *d02* value which is encrypted) in the Service/Package *plaintextpass* field. This might be done if needed for provisioning. • 0 - No • 1 - Yes
usageinvday	Day of month usage billing should occur. See *usageoptions* for use.
usageduration	Duration between usage billings. See *usageoptions* for use.
usageoptions	Specifies Usage timing • 1 - Bill on the Service *invday* (Day of Month to Bill - Same as Account Day of Month for synchronized Services) • 2 - Bill on the Service *invdate* (Next Bill Date)

Name	Description
	• 4 - Bill Monthly on the Package/Service Definition *usageinvday* • 8 - Bill using both the Package Service Definition *usageinvday* and *usageduration*
printzerocharge	Display zero dollar sales on a Bill • 0 - No • 1 - Yes
quantity	Default quantity to sell. Sale is price multiplied by (quantity minus *quantityfree*)
quantityfree	Default amount of quantity that should not be billed
taxincluded	Tax included in sales price • 0 - No • 1 - Yes
cnt	Number of times to bill Second Price. If 0, no billing takes place and First Price is the last available price.
initialstate	Initial Status for a newly created Service • 0 - Open • 3 - Waiting for Payment • 5 - Prospective
initialreason	Initial reason for why a Service is not Open • 0 - Not Applicable • 10 - Future Start Date • 11 - Service Definition Setting • 12 - Wait for Qualification • 13 - Wait for Activation
lprice	Third Price
lcnt	Number of times to bill Third Price. If 0, no billing takes place and the setting of *cnt* determines billing.
lduration	Third Duration
billnexterm	In addition to a prorated term, bill an unprorated term
refwait	Number of days to wait before giving referral credit
waitcredit	Give Store Credit for time when Service Status was Wait for Payment
flags	Bit Field

Bit Position	Bit Value	Description
0	1	Must have a POP (Point of Presence) selected.
1	2	Bill *setup* even if Service is in Prospective Status

Name	Description		
	Bit Position	**Bit Value**	**Description**
			upon Service creation
	2	4	Bill *deposit* even if Service is in Prospective Status upon Service creation
	3	8	Do not execute Service Hook for Provisioning
	4	16	Do not display Dates of Service on the customer's Bill
	5	32	If prorate should occur, prorate date only, not billed amount
	6	64	Denotes a Recurring Service Fee with Provisioning Variables
	7	128	Denotes a One Time Item
	8	256	Denotes a Package
	9	512	Ignore Account Profile Service Offset Days
	10	1024	Create a Transaction for a Zero Dollar billing
	11	2048	unused
	12	4096	Automatically generate Store Credit for Recurring Sales
	13	8192	Automatically generate Store Credit for Usage Based Sales
	14	16384	Available only to a Package
	15	32768	Uses MAC Authentication
	16	65536	Denotes a Bundle

Name	Description		
	Bit Position	**Bit Value**	**Description**
	17	131072	Bill *iprice* even if Service is in Prospective Status upon Service creation
	18	262144	Setup billing is Setup price multiplied by quantity
	19	524288	Used to display Package/Service Definitions available for online Customer Sale.
	20	1048576	Roll up/Combine all billings to a single line on bill for Packages
	21	2097152	Display the Package/Service Name (*service.d01*) on bill.
	22	4194304	Use to preview a subset of Package/Service Definitions available for online Customer Sale.
	23	8388608	Display the Package/Service Definition Name (*descr*) on bill.
	24	16777216	Allow billing of Services associated with a Virtual Company different than those to which this Service Definition is associated.
	25	33554432	Must have an AP (Access Point) selected.
	27	134217728	A Recurring Service Fee will use the *fccservice* value of the parent Package/Service

Name	Description		
	Bit Position	**Bit Value**	**Description**
	28	268435456	Ignore discounting settings.
	29	536870912	Do not use prorate setting of parent Package/Service
servicetemplate	deprecated		
generation			
meterdrounding			
warnexpirationdays			
invoicediplaygroup	Used to group like billing together on a bill		
useuser	Bit Field		
	Bit Position	**Bit Value**	**Description**
	0	1	Copy user *loginid* to Service *d01* field on creation
	1	2	Copy user *password* to Service *d02* field on creation
taxclass	Setting from List "*taxclasses*"		
billingclass	Setting from List "*billingclasses*"		
f01	Bit fields used to specify setting for d01-d10 fields		
	Bit Field	**Bit Value**	**Description**
	0	1	Value required
	1	2	Value is a password
	3	8	Value must be unique
	5	32	Append the Virtual Company domain
	10	1024	Display value on Bill
	11	2048	Uses List for values
	14	16384	Prorate value to subordinate Services with the same provisioning Label
f02	See *f01*		

Name	Description
f03	See *f01*
f04	See *f01*
f05	See *f01*
f06	See *f01*
f07	See *f01*
f08	See *f01*
f09	See *f01*
f10	See *f01*
e01	Explanation for d01
e02	Explanation for d01
e03	Explanation for d01
e04	Explanation for d01
e05	Explanation for d01
e06	Explanation for d01
e07	Explanation for d01
e08	Explanation for d01
e09	Explanation for d01
e10	Explanation for d01
saledescr	Item Name
saledetail	
saleimage	
overduefee	Fee assessed when a Service is Suspended for being Overdue
fcctechcode	Value from List fcctechcodes3
fccuprate	Advertised maximum upload rate - Mbps
fccdownraate	Advertised maximum download rate - Mbps
fcclloop	deprecated
fccvoip	deprecated
fccwhsale	deprecated
fccother	deprecated
fccservice	Value from List *fccservice*
a01	Asset Class for d01
a02	not used
a03	Asset Class for d03
a04	Asset Class for d04

Name	Description
a05	Asset Class for d05
a06	Asset Class for d06
a07	Asset Class for d07
a08	Asset Class for d08
a09	Asset Class for d09
a10	Asset Class for d10
etfee	Early Termination Fee
etfeetaxable	• 1 - taxable • 0 - not taxable • -1 - not applicable
feeselectgroup	Selectable Fee Group (Recurring Service Fee)
county	County(Recurring Service Fee)
changeclass	Value from List "*changeclasses*"
notebilldetail	Reference to Note table containing additional Billing Detail for Bill
technology	Value from List "*infrastructuretechs*"
accountclass	Value from List "*accountclasses*"
features	Value from List "*features*"
excludetaxtype	not used
feetype	Fee Type(Recurring Service Fee)

service Table (Products)

Pricing is in cents.

Name	Description	Type
number		*int(11)*
user	Foreign key to *user.number*	*int(11)*
state	Status • 0 - Open • 1 - Closed • 2 - Suspended • 3 - Waiting for Payment • 5 - Prospective	*tinyint(4)*
startdate		*date*
enddate		*date*
servdef	Foreign key to *servdef.number*	*int(11)*
free		*tinyint(4)*
wsetup	• 0- Yes	*tinyint(4)*

Name	Description	Type
	• 1 - No • 2 - On Open	
entdate		*date*
enttime		*time*
empl		*mediumint(8)*
lastmodempl		*mediumint(8)*
d01	Customizable Provisioning Field Label	*char(100)*
d02	Customizable Provisioning Field Label	*char(100)*
d03	Customizable Provisioning Field Label	*char(100)*
d04	Customizable Provisioning Field Label	*char(100)*
d05	Customizable Provisioning Field Label	*char(100)*
d06	Customizable Provisioning Field Label	*char(100)*
d07	Customizable Provisioning Field Label	*char(100)*
d08	Customizable Provisioning Field Label	*char(100)*
d09	Customizable Provisioning Field Label	*char(100)*
d10	Customizable Provisioning Field Label	*char(100)*
invdate	Next Bill Date	*date*
taxable	• 1 - taxable • 0 - not taxable • -1 - not applicable	*tinyint(4)*
taxrule		*int(11)*
iprice	First Price	*decimal(30,10)*
iduration	First Duration - positive integer for months, negative integer for days	*int(11)*
icnt	Number of Times to Bill First Price	*int(11)*
price	Second Price	*decimal(30,10)*
duration	Second Duration - positive integer for months, negative integer for days	*int(11)*
cnt	Number of Times to Bill Second Price	*int(11)*
lprice	Third Price	*decimal(30,10)*
lduration	Third Duration - positive integer for months, negative integer for days	*int(11)*
lcnt	Number of Times to Bill Third Price	*int(11)*
tot_seconds	used for metered billing	*int(11)*
prime_seconds	used for metered billing	*int(11)*
ponumber	PO Number	*char(30)*

Name	Description	Type
refaccount	Referral Account Number	*int(11)*
refdate	Referral Credit Date	*date*
misc01	Integer Field through misc05	*int(11)*
misc02		*int(11)*
misc03		*int(11)*
misc04		*int(11)*
misc05		*int(11)*
misc06	Character (50) Field through misc10	*char(50)*
misc07		*char(50)*
misc08		*char(50)*
misc09		*char(50)*
misc10		*char(50)*
misc11	Date Field through misc15	*date*
misc12		*date*
misc13		*date*
misc14		*date*
misc15		*date*
misc16	Time Field through misc20	*time*
misc17		*time*
misc18		*time*
misc19		*time*
misc20		*time*
generation		*timestamp*
reason	Reason for why a Service is not Open • 0 - Not Applicable • 1- Overdue • 2- AUP Violation • 3- Administrative • 4 - Expired • 5 - Over Usage • 8 - Customer Request • 9 - No Time Left • 10 - Future Start Date • 11 - Service Definition Setting • 12 - Wait for Qualification • 13 - Wait for Activation	*mediumint(8)*
reasonbyuser		*tinyint(4)*

Name	Description	Type
cleardate	Last date for metered billing the totals were zeroed out	*date*
bwreportdate	Last date for metered billing fields were updated	*date*
tiervalue	Last date for metered billing the totals were zeroed out	*decimal(20,8)*
suspdate	Date service was suspended	*date*
setup_taxable	1 - taxable 0 - not taxable -1 - not applicable	*tinyint(4)*
wdeposit	0 - it has not been collected 1 - do not collect 2 - collect on open 3 - it has been collected	*tinyint(4)*
resources	Provisioning resources that should match the Service Definition Resources	*int(11)*
quantity		*decimal(14,6)*
invcnt	Number of times an item has been billed not counting proration	*int(11)*
changetocnt	Number of times a Service has gone through a Service Change	*int(11)*
plaintextpass	Number of times a Service has gone through a Service Change	*char(50)*
invday	Billing/Prorate Day from an Account	*tinyint(4)*
nextusagedate	Next Usage Bill Date	*date*
quantityfree		*decimal(14,6)*
cyclept	Where you are in the cycle	*int(11)*
salesrep	No longer used - now in *serviceagenttable*	*int(11)*
tiercost		*bigint(20)*
prevstate		*tinyint(4)*
prevreason		*mediumint(8)*
package	The Package a Service belongs to if needed	*int(11)*
flags	<table><tr><th>Bit Position</th><th>Bit Value</th><th>Description</th></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>2</td><td>Sales Return of Future Billed</td></tr></table>	bit-mask

Name	Description			Type
	Bit Position	**Bit Value**	**Description**	
			Service when closing a Service	
	2	4	Deposit Return of $xx	
	3	8	Reopen Services	
	4	16	Sales Return to be given when opening	
	5	32		
	6	64	Bill an additional term in addition to the prorated term	
	7	128		
	8	256	Package	
	9	512		
	10	1024	Display zero priced Usage	
	11	2048	Ignore Discounts	
	12	4096	Auto Store Credit Recurring Sales	
	13	8192	Auto Store Credit Usage Sales	
datetoreopen	Date to Reopen/Date to Resume Billing			date
enddatereason				*tinyint(4)*
importid	Import id of the system that the service was imported from			char(100)
service	If a fee, this is the service it belongs to			int(11)
pctdiscout	Discount Percentage			decimal(10,6)

Name	Description	Type
pctdiscountexpdate	Date Discount Expires	date
usagepctdiscout	Usage Discount Percentage	decimal(10,6)
usagepctdiscountexpdate	Date Usage Discount Expires	date
discountamount	Discount Amount	*bigint(20)*
prepaidtime	Prepaid card usage	*int(11)*
prepaidexpire	Prepaid card expiration	*char(19)*
etfee	Early Termination Fee	*bigint(20)*
contractenddate	Contract End Date	date
locationuser	Location - Used for mapping and timezone purposes	int(11)
setup	Setup Fee	*bigint(20)*
recurbillnote	Additional Billing Text - Note that will appear on bill	*int(11)*
descr	Description	*char(100)*
promotion	Referral Amount	*int(11)*
ap		*int(11)*
creditreason	Credit Reason	int(11)
disctype	Discount Type	*tinyint(4)*
creditbillingdisplay	Credit Billing Display - will show with discount on bill	*char(100)*
cnttf1	count for tax feature	*mediumint(8)*
cnttf2	count for tax feature	*mediumint(8)*
cnttf3	count for tax feature	*mediumint(8)*
plsp		decimal(10,6)

ticket Table (Ticket)

Name	Description
number	
generation	
empl	
lastmodempl	
subject	
originator	Who authored the data for the ticket.
entdate	
enttime	
user	*user.number* - deprecated

Name	Description
reference	The *number* of the record from the *fromtable* table.
queue	*queue.number*

state	**0**
	    Open
	-1
	    Inactive
	    Unassigned
	1
	    Closed
	2
	    Reopened

priority	Calculated from the queue problems and automatically updated if Service Level Agreement in effect.
assignedto	*auth.number*
problem	Populated from the queue problems
resolution	Populated from the problem resolutions
lastactdate	Date of last activity for the ticket.
lastacttime	Time of last activity for the ticket.
misc01	
misc02	
misc03	
misc04	
misc05	
misc06	
misc07	
misc08	
misc09	
misc10	
externalid	Used to populate reference to third party system.
fromtable	table to which the ticket is attached

user Table (User)

The User table is used for service locations and alternate contacts. Multiple User records may be linked to one Account.

Name	Description	Type
account	Foreign key to *account.number*	*int(11)*
addr1	Address Line 1	*char(50)*

Name	Description	Type
addr2		*char(50)*
addr3		*char(50)*
addr4		*char(50)*
aphone	Additional Phone.	*char(40)*
appointmentroute	Appointment Route for service location. Used with Scheduling. List *appointmentroutes*.	*int(11)*
billinglabel	Additional information added to Location Based Billing Statements/ Invoices	*char(100)*
censusblock		*char(8)*
censustract		*char(8)*
city		*char(30)*
cleartextpassword	Clear text password for Customer Portal logins. Storing of the clear text is controlled by List *systemstate* entry *usercleartextpassword*.	*char(100)*
company	Company name at service location.	*char(50)*
country		*char(30)*
county		*char(30)*
countyfips		*char(8)*
custsecret	Original use deprecated. May be used for miscellaneous data.	*char(50)*
dataphone	Data Phone.	*char(30)*
datetoreopen	Date in the future to automatically reopen a User.	*date*
dlnumber	Original use deprecated. May be used for miscellaneous data.	*char(20)*
dlstate	Original use deprecated. May be used for miscellaneous data.	*char(30)*
dob	Original use deprecated. May be used for miscellaneous data.	*date*
dphone	Day phone.	*char(40)*
email		*char(100)*
empl		*mediumint(8)*
enddate	Date User Closed or Date to Close. The date after end of service.	*date*
enddatereason	Reason for closing. List *reasons*	*tinyint(4)*
entdate		*date*

Name	Description	Type
enttime		*time*
flags	Internal Use	
fname	First Name	*char(30)*
generation		*timestamp*
importid		*char(100)*
lastaccess	Last time Customer Portal used by this User.	*int(11)*
lastip	Last IP from which Customer Portal accessed.	*char(16)*
lastmodempl		*mediumint(8)*
lname	Last Name	*char(30)*
location	Point object representing Latitude/ Longitude	*point*
locationnote	Note about Service Location	*text*
loginid	Customer Portal user name.	*char(100)*
maiden	Original use deprecated. May be used for miscellaneous data.	*char(30)*
misc01	Integer Field through misc05	*int(11)*
misc02		*int(11)*
misc03		*int(11)*
misc04		*int(11)*
misc05		*int(11)*
misc06	Character (50) Field through misc10	*char(50)*
misc07		*char(50)*
misc08		*char(50)*
misc09		*char(50)*
misc10		*char(50)*
misc11	Date Field through misc15	*date*
misc12		*date*
misc13		*date*
misc14		*date*
misc15		*date*
misc16	Time Field through misc20	*time*
misc17		*time*
misc18		*time*
misc19		*time*

Name	Description	Type
misc20		*time*
mname	Middle Name	*char(30)*
modem	List *modems*	*mediumint(8)*
mspeed	List *mspeeds*.	*tinyint(4)*
nphone	Night Phone	*char(40)*
ntries	Number of failed login attempts using the Customer Portal	*tinyint(4)*
number		*int(11)*
os	List *oss*	*tinyint(4)*
password	Password for the Customer Portal	*char(11)*
pctdiscount	Percent Discount for all Sales to this User. Overrides Account Percentage Discount.	*decimal(10,6)*
pctdiscountexpdate	Date the Discount expires.	*date*
prevreason	The previous reason a User was not Open.	*mediumint(8)*
prevstate	The previous Status of a non-Open User/	*tinyint(4)*
reason	The reason an User is not open.	*mediumint(8)*
reasonbyaccount	Internal Use Only.	
resetdate	Date password reset request started.	*date*
resettime	Time password reset request started.	*time*
resettoken	Token for password reset request.	*char(32)*
schedulearea	Deprecated.	
securityanswer1	Answer for *securityquestion1*	*char(255)*
securityanswer2		*char(255)*
securityquestion1	List *securityquestions*	*tinyint(4)*
securityquestion2		*tinyint(4)*
sessionid	Session id for Customer Portal.	*char(32)*
ssn	Original use deprecated. May be used for miscellaneous data.	*char(12)*
startdate		*date*
state	Status of the User: **0** Open **1** Closed	*tinyint(4)*

Name	Description	Type
	2 Suspended	
statefips		*char(2)*
statename	State or Province	*char(30)*
suspdate	Date the User was Suspended.	*date*

taxregion	Tax region for Services billed at this location.
	-1
	    Not Applicable. Only use if never collecting any type of tax.
	otherwise
	    Foreign Key to *taxregion.number*.
		int(11)

timezone	List *timezones*	*tinyint(4)*
title	Salutation. List *titles*	*tinyint(4)*
usagepctdiscount	Percentage discount for Usage Sales at this Service Location.	*decimal(10,6)*
usagepctdiscountexpdate	Date percentage discount for usage sales expires.	*date*
verifiedaddr	Address verified using Address Verification API - currently only USPS supported.	*int(11)*
zip	ZIP/Postal Code	*char(30)*

config Table (Virtual Company)

Name	Description
number	
generation	
empl	
lastmodempl	
sortorder	
entdate	
enttime	

state	**0**
	    Available
	1
	    No New Accounts
	2
	    Discontinued

Name	Description
company	
addr1	
addr2	
addr3	
addr4	
city	
statename	
zip	
country	
phone	
fax	
taxid	
vataxid	
bcfname	Billing Contact First Name
bclname	Billing Contact Last Name
bcemail	Billing Contact Email
acfname	Administrative Contact First Name
aclname	Administrative Contact Last Name
acemail	Administrative Contact Email
primestart	
primeend	
ccexpwarn	
cctries	
ccspan	
defdomain	
ccdecipher	Deprecated
autocredit	
syncinvdate	
latetaxable	
lateonlate	
rounding	
defaultprofile	
btries	
bspan	

Name	Description
graphic	

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

Open 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

Open Accounts, Users and Services that are in *Suspended* **Status** where **Date to Reopen/Date to Resume Billing** is today or in the past.

SuspendExpiredAccounts

CloseSuspendedAccounts

SuspendExpiredUsers

CloseSuspendedUsers

SuspendExpiredServices

CloseSuspendedServices

SuspendExpiredServdefs

Change the **Status** of *Open* Accounts/Users/Packages/Services and underlying entities to *Suspended* where the **End Date** is today or in the past.

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 Exipre** 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

Open Services that were Suspended for over usage if a new term is in effect.

BillServics

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

UpdateStaticPastDueBalances

Update the financial fields stored in the Account table that summarize the Account's financial status.

ResetGraceDates

Clears Accounts' **Grace Date**s 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 base 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

Open 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

Close Service Agent Commission entries that have an **End Date** today or in the past.

NoticeLongSuspendAccounts

NoticeLongSuspendUsers

NoticeLongSuspendServices

NoticeAccountLocks

NoticeRefundsDue

NoticeLicenseExpiration

NoticeUnFundedRefunds

Generate various system notices that will be output to the log file.

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 Fee**s 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

III

Customer Portal

Topics:

- *Concepts*
- *How To*
- *Reference*

Chapter

1

Concepts

Topics:

- *Customer Portal Overview*
- *Architecture*
- *Setting up the Customer Portal on Debian*

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 147) 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.

Setting up the Customer Portal on Debian

The Customer Portal requires PHP 8.1 or greater. It also requires the following php modules: *php-pear*, *php-soap*, *php-xml*, *php-gd*, *php-pdo*, *php-mbstring*, *php-curl*. These prerequisites are handled by the BillMax installation process. However, should you choose to run the Portal on another host (your web site for instance), you will need to ensure these packages are installed.

Prior versions of BillMax running CentOS 7® required replacing the stock PHP available on the OS. This step is no longer necessary. The portal uses SSL for communication between the portal (client) and BillMax (server). To use the portal, a SSL certificate must be created. In addition BillMax must be configured for the portal using the setup of a Remote Application in the BillMax Staff Interface.

Chapter

2

How To

Topics:

Configure Customer Portal for Debian

1. 📝 **Note:** During this process you will be prompted to enter a password or pass phrase. Be sure to supply the same password everywhere prompted.

   Generate a SSL certificate to be used for encryption of data between the Customer Portal and the Customer Portal daemon (Edge) and create the bx_edge Debian Service used to manage the edge daemon.

   a) As root user, execute `/usr/local/billmax/bin/setup_edge`
   b) Enter a value for all prompts except: Organizational Unit Name, Challenge password, and Company name.

2. Generate a new application encryption key. As the billmax user, execute:

   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) Choose **Customer Portal** for **Type**.
   f) 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 as well as the statement and invoice emails.
   g) Choose **Company** and **Customer Profile**. These values are used only for new customer registrations.
   h) 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.
   i) Modify the **Customer Portal Login Parameters** as required. Note that the security questions and **CAPTCHA** options are presently not used in the new portal.
   j) Click **Save**.

4. Edit `/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. Start the edge service. As the root user:

   a) `systemctl start bx_edge`

6. Test functionality by pointing a browser at `https://<billmaxserver>/testportal` where *<billmaxserver>* is the address of the BillMax server. A view similar to what is shown below indicates success. If you get a permission error, check `zbillmax.conf` in `/etc/httpd/conf.d`. In the `/usr/local/`

billmax/html section, change *AllowOverride None* to *AllowOverride All* and restart the Apache server.
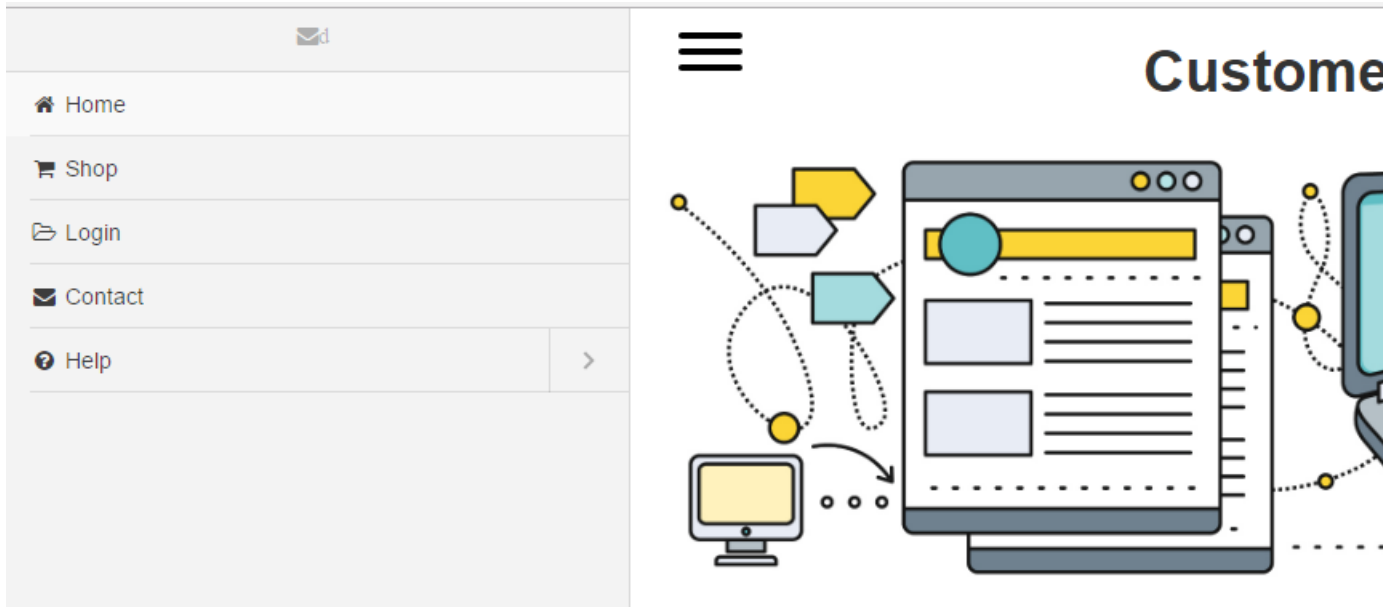
7. Click **Login** to test access to the `edge` service. Enter a user name and password. The credentials exists in BillMax on user (location) records.

Setup the Customer Portal on a Remote Host

Ideally the BillMax server should not be accessible to the Internet for security reasons. If the BillMax host is generally unavailable to the public, the Customer Portal must be installed on a server that is publicly available. Typically, this server is the public Web server for an ISP. It may also be installed on a server dedicated to use by the Portal.

1. 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 include the period at the end of this command.**

   c) `scp /tmp/portal.v2.tgz <remote-user>@<remote-host>:`

2. 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`.

3. 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.

4. When installing the portal at a remote site, ensure that the Apache server allows the use of `.htaccess` files by using the "`AllowOverride All`" directive in the appropriate <Directory> section for the portal in the `httpd conf` file.

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 or choose **System Administration** > **Customer Portal Configuration** to update the file in the Staff Portal.

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 and *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             ▼]
│
```

4. To add Terms and Conditions, for packages, services, or one time items through the portal, add the relevant file. The terms and conditions go in this directory and can have up to one per service definition if needed: `<portal root dir>/storage/app/public/termsXX.html` where XX is the *servdef* number of a package, service or one time sale item chosen in the cart.

Add Authorization in Customer Portal

Note: Payment Authorizations can be captured when inputting Bank Accounts or Credit Cards.

1. Choose **System Administration** > **Customer Portal Configuration** and set *authorization_required* to true.

2. Create a Ticket Queue *Payment Method Authorization* with an Issue *Payment Method Authorization* and a Resolution of *Payment Method Authorization* and note the number of each item. Make sure to set **Messages Used** to Yes for the Ticket Queue.

3. 📝 **Note:**

```
/*
    * Ticketing support queues available to user; queues must exist on
server!
    */
   'queues' => '{"Billing": 1, "Service": 2, "Payment Method
Authorization": 3}',

   /*
    * Hard coded; problem set on create. resolution set on close
    */
   'problems' => '{"Billing": 1, "Service": 2, "Payment Method
Authorization": 2}',
   'resolutions' => '{"Billing": 1, "Service": 2, "Payment Method
Authorization": 5}',
```

Update `edge.php` by choosing **System Administration** > **Customer Portal Configuration** and add these items by number and save. Restart the customer portal.

4. Verify that an authorization is saved. When a customer clicks the Authorize button when adding a payment method, a closed Ticket will be created on the User with a Message containing identifying information and User

Messages

[New] [Collapse]

29   10/26/2021 16:31:59 Payment Method Authorization

I hereby authorize Test Co to electronically charge my credit card (shown below) for payment of Test Co services and includes electronic credit (for refunds) if applicable.

I understand that this authorization will remain in affect as long as this specific payment method remains on file and

Credit Card Details

Account Holder:	Michele Lee
Credit Card:	4***********1111
Expiration Date:	2025-09-30
AutoPay?	Yes

Authorized By

Customer:	Michele Lee
Time:	10/26/21 9:10 pm
Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/

information.

Add Shop by Virtual Company in Customer Portal

1. Choose **Systems Administration** > **Customer Portal Configuration**.
2. Set *shop_by_company* to true.
3. Set other shopping cart variables such as whether you want a ticket to be created for the new order, an email confirmation sent to the customer, or a payment collected for the order.
4. Limit the products available to the customer by Virtual Company by setting **Company** on the Package or Service Definition.

Allow Customers to Sign into Customer Portal with Social Accounts

Configure the Customer Portal to accept logins using AWS Cognito® for one or more social accounts.

1. Choose **Systems Administration** > **Customer Portal Configuration**.
2. Set *cognito* to true.
3. Update the other AWS Cognito® parameters. Click Save. Contact *support@billmax.com* for further support.
4. For each social account type, an app account will have to be setup. Check the documentation for each system about how to create.

Limit Payment Methods in the Customer Portal

1. Select **Billing Administration** > **Lists** and find the list **paytypes**. Click on the List number.
2. Click the checkbox under **Select List Item** that you want to add or delete from your clients' choices.
3. Click **Toggle Portal Usage**. It is a button at the top of **List Items**
4. Select **System Administration** > **Manage Edge Process** under **Operations**.
5. Click **Stop edge** and wait until it is completed. Click **Start edge** to restart the Customer Portal with your updated list.

Chapter

3

Reference

Topics:

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
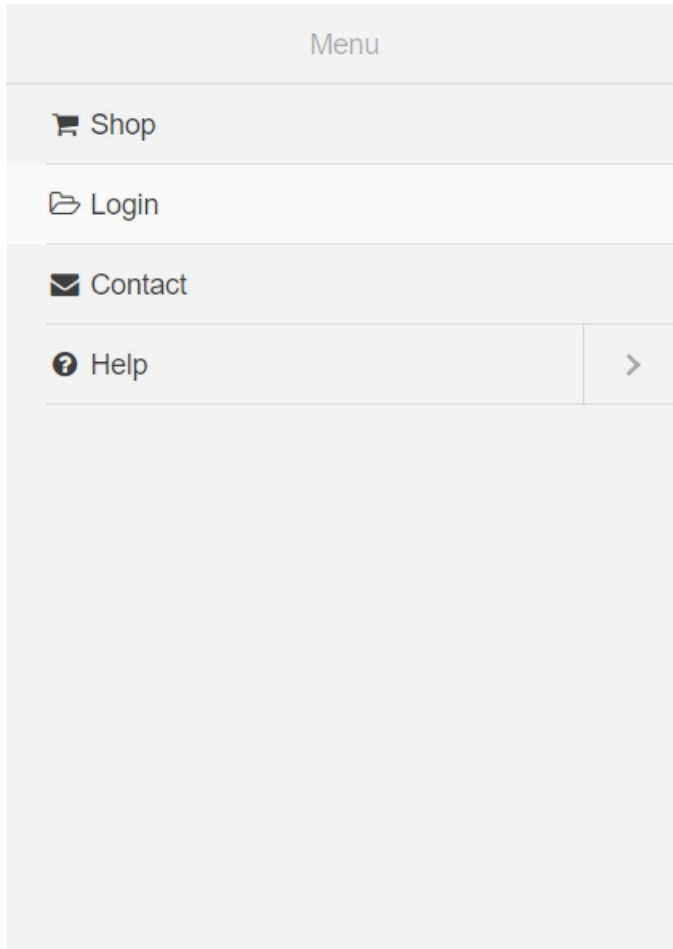
/nm/getAccount

On the BillMax server after the Customer Portal is configured, the URL */testportal/portaliframe.html* may be used to examine the various endpoints.

Part

IV

Integrations

Topics:

- *Calix Integration*
- *Market Broadband Integration*
- *MagicMail Integration*
- *Salesforce Integration*
- *Zoho CRM Integration*

Chapter

1

Calix Integration

Topics:

- *Concepts*
- *How To*

Concepts

Calix Subscriber Provisioning

BillMax's architecture allows for easy integration and provisioning of subscriber information in any OSS. Please see Provisioning *Concepts* on page 132 for general details on this methodology. This chapter explains our Calix Subscriber provisioning. Contact Sales (*sales@billmax.com*) for availability information on other OSS systems.

These updates identify and authorize subscribers and their equipment. In addition to access, our provisioning currently enables ExperienceIQ, ProtectIQ, and Bark. Contact Support (*support@billmax.com*) if you need changes to this service list.

How To

Configuring BillMax for Calix

Obtain the necessary software, licenses, and account from Calix. In particular, you will need access to their Subscriber Services API in the Calix Operations Cloud. For the API usage, you will need the following settings: Client Id, Client Secret, Username, and Password.

1. In BillMax, choose **System Administration** > **Resources.** Create a Resource name "Calix" and record the associated resource number. You will need this number when editing the service hook and production definitions.

2. Choose a location field on the user record not already being used for holding a unique location identifier required by Calix. Edit the *useridlabels* list (**Billing Administration** > **Lists).** . Change the *Item* field to Calix ID. Note that the labels vary in the maximum length of data accepted. Choose a label (2-6) with enough space to hold your unique ID for the location. This ID has no required format. Choose a schema that provides for uniqueness when the ID is entered. Using the table below record the field used. You will need it when editing the service hook.

   - useridlabels

   Table 10:

Label	Field	Size
Label2	ssn	11
Label3	dlnumber	20
Label4	dlstate	20
Label5	maiden	30
Label6	custsecret	50

3. Create one or more product (package or service) definitions for Calix services to be sold and provisioned. In the Provisioning Information tab, click the Resources checkbox labeled Calix and set up provisioning fields: Mac Address (Router Mac address), ONT FSAN (Optical access network number) and USOC(Calix Universal Service Ordering Code). Record the dXX fields used; you will need them when editing the service hook.

4. Edit the service hook (/usr/local/billmax/local/service_hook). If the file does not exist, rename or copy the *service_hook-dist* file in the local directory. Make sure the file is owned by 'billmax' and is executable. In the service_hook file, edit the SERVICES hash. Add the following where ? is the resource number used for Calix:

```
my $cmd = "$BILLMAXDIR/local/calix.pl";
$SERVICES{?}{name}              = "calix";
$SERVICES{?}{provision}         = "yes";
```

```
$SERVICES{?}{data}              = ["-s", "{NEW_state}", "-n", "{NEW_d01}",
 "-fsan",   I    I     "{NEW_d04}", "-usoc", "{NEW_d05}" ];
$SERVICES{?}{cmd}{create}       = "$cmd -action create";
$SERVICES{?}{cmd}{delete}       = "$cmd -a delete " . escL($ENV{NEW_d01});
$SERVICES{?}{cmd}{update}       = "$cmd -a update";
$SERVICES{?}{remotecmd}         = "";
```

5. Edit the calix.pl script (`/usr/local/billmax/local/calix.pl`). Set the credential values for Client ID, Client Secret, Username, Password. Set the value that identifies the field used for the Location ID (location_field), MAC address (mac_field), ONT FSAN (ont_fsan_field) and USOC (usoc field).

6. Install the user hook (`/usr/local/billmax/local/user_hook`). Assuming that no user hook already exists, create an alias to the calix.pl script.

   a) Execute `ln -s calix.pl user_hook`

   b) Make sure the calix.pl script is owned by 'billmax' and is executable.

   c) If you already use a user hook for other purposes, you will need to edit your script to include the calix.pl script functionality.

Chapter

2

Market Broadband Integration

Topics:

- *Concepts*
- *How To*

Concepts

Integration Overview

BillMax is now integrated with Market Broadband's Proximity Plus Marketing Service[SM]. This service uses data received from BillMax to market your internet services to potential customers whose premises are nearby any new subscribers. The integration sets up a recurring query for new subscribers and a FTP file upload.

How To

Activate Market Broadband® Integration

Please see *https://marketbroadband.com* for details.

1. Create an account at Market Broadband®.

2. 📝 **Note:**

   > *Account* is the account name at Market Broadband. *FccService* defines the service or package types used for new subscribers queries. 1 = Internet, 3 = VoIP. *FtpDirectory* is the upload directory on the FTP server. *Username* and *Password* are the FTP credentials for logging in. *Radius* specifies the area (in miles) around your new customers that are potential marketing targets.

   Configure the list **MarketBroadband** for all fields except *LastProcessedTime* from the values obtained from Market Broadband.

3. Choose **System Administration** > **Batch Processing**, click the number of the **marketbroadband Process Group**.

4. Choose the batch process number for **Market Broadband Proximity Plus Export** change the status to **Active**.

5. Click **Run Process** to manually invoke the process for a test.

6. If successful, change the **marketbroadband Process Group** to **Active**.

Chapter

3

MagicMail Integration

Topics:

- *Concepts*

Concepts

Integration Overview

MagicMail® is a carrier grade email platform built specifically for ISP's, Telco's, and Cable companies. It offers Spam protection, stability and scalability.

Configure MagicMail according to what you are selling. The integration supports domain where an entire domain is hosted or provided (example.com), individual email accounts (bill@example.com) and aliases (forwards). Typical setup is a general domain for your users, like example.com that is not billed, but has mailboxes for your customers that are billed or given away as well as domains and emails for business customers. Email passwords can be changed by your customers in the BillMax customer portal or in the app provided by MagicMail. Changes to suspended Users are not allowed by MagicMail.

To use, the MagicMail list in the BillMax UI must be configured. You will need a URL, username and password from MagicMail.

The scripts include:

• BxProvisionMagicMailMailbox.pl - Handles mailbox and email addresses.
• BxProvisionMagicMailList.pl - Handles forward lists.
• BxProvisionMagicMailDomain.pl - Handles domains.
• MagicMail.pm - Magic Mail module (used by all of the above).

To incorporate into your service hook use the following SERVICES hashes (change resource # {N} to a designated resource. See System Administration > Resources in UI. Note that these directives require the latest `service_hook` (from 2017 QTR 1 on) to work.

More can be learned about the integration in the readme file. Contact support with questions.

Chapter

4

Salesforce Integration

Topics:

- *Concepts*
- *How To*
- *Reference*

Concepts

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 147 entry.
- *Profile* - Number of Account Profile to use when creating Accounts in BillMax. Default is the value from Remote Application entry.

How To

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

Type	Label	Length	Field Name
Text	Company	50	*BillingCompany*
Text	Email Address	100	*BillingEmail*
Text	First Name	30	*BillingFirstName*
Text	Last Name	30	*BillingLastName*

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

```
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).

7. Select **File>New>VisualForce Page**.

8. Enter **Name** as *BillMaxAccount*.

9. Enter for the page

```
<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**.

13. Find **New** and choose **Edit** action.

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 271 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 271 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 **saleforce-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.
7. In Salesforce select **Setup>Apps>App Manager** and click **New Connected App**.
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 AUTHENTICATIONS SETTINGS section and set the values of the parameters that have been retained from *Create Salesforce CRM User* on page 270 .
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/locatl/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.

Reference

Salesforce Field Mappings

Table 11: Accounts

BillMax Field	Salesforce Field
importid (Primary Key)	*Id*
cfname	*BillingFirstName_c*
clname	*BillingLastName_c*
company	*BillingCompany_c*
addr1	*BillingStreet*
city	*BillingCity*
statename	*BillingState*
email	*BillingEmail_c*
zip	*BillingPostalCode*
country	*BillingCountry*
cphone	*Phone*
fax	*Fax*

Table 12: Contacts

BillMax Field	Salesforce Field
importid (Primary Key)	*Id*
fname	*FirstName*
lname	*LastName*
addr1	*OtherStreet*
city	*OtherCity*
statename	*OtherState*
email	*Email*
zip	*OtherPostalCode*
country	*OtherCountry*
dataphone	*OtherPhone*
nphone	*HomePhone*
aphone	*MobilePhone*
account (Primary Key)	*AccountId*

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.

Chapter

5

Zoho CRM Integration

Topics:

- *Concepts*
- *How To*
- *Reference*

Concepts

Integration Overview

Zoho CRM® is a Customer-Relationship Management (CRM) tool. The BillMax Zoho CRM Integration provides bi-directional data flow between BillMax Accounts and Users and Zoho Contacts. On the BillMax side, the integration is comprised of an Edge service endpoint to receive changes from Zoho and account and user hooks to "push" changes from BillMax to Zoho. The integration requires a BillMax license key that includes the Customer Portal module for the Edge service to be implemented.

Zoho Contacts when created or edited result in the corresponding data being replicated in BillMax and vice versa.

The mapping is a single Zoho Contact to a BillMax Account and User. The "Other*" address fields in the Contact are mapped to the BillMax User.

This integration requires a Zoho account that includes the necessary licenses to use Zoho's Webhooks and CRM REST APIs and is available only in BillMax 4.0.8500 or versions released since.

How To

Configure Zoho

Refer to Zoho documentation for the specific details. Items in *italics* denote BillMax site specific values that must be supplied by the BillMax customer.

1. Add new fields to the Contact record.

    Warning! The field names must be exactly as shown and are case sensitive! **Copy to Billmax** is optional and needed only if not all Contacts are synced with BillMax.

    Table 13: Zoho Fields

Field Name	Type
Billmax Account Number	Number
Company	Single Line
Copy to Billmax	Checkbox

2. Configure the `BillMax Webhook` with the following settings:

    Name: `Billmax`

    URL: `https://`*Billmax portal host*`/zohoContact`

    Method: `POST`

    Module: `Contacts`

    Warning! The parameter names must be exactly as shown and are case sensitive!

    Warning! If the checkbox **Copy to Billmax** is used, the `webhook` rule must be changed so the hook is run only if **Copy to Billmax** is true.

    Table 14: Parameters

Parameter Name	Parameter Value
`Contact_Id`	Contact Id
`Last_Name`	Last Name
`First_Name`	First Name

Parameter Name	Parameter Value
`Lead_Source`	Lead Source
`Email`	Email
`Phone`	Phone
`Mailing_Street`	Mailing Street
`Mailing_City`	Mailing City
`Mailing_State`	Mailing State
`Mailing_Zip`	Mailing Zip
`Mailing_Country`	Mailing Country
`Other_Street`	Other Street
`Other_City`	Other City
`Other_State`	Other State
`Other_Zip`	Other Zip
`Other_Country`	Other Country
`Other_Email`	Other Email
`Other_Phone`	Other Phone
`Billmax_Account_Number`	Billmax Account Number
`Company`	Company

3. Create Work flow Rule.

   Name: `Billmax webhook`

   When: `On Record Create. Create or Edit.`

   Condition: `All Contacts` or `Contacts matching certain criteria`

   If only certain Contacts are synced between Zoho and BillMax, the condition should be **Copy to Billmax** is selected.

   Instant Action: `Billmax` (webhook created in Step 2)

Create REST API Credentials

1. Using the Zoho Developer Console, create a new Self Client application. The Zoho Developer Console is not part of the regular interface. Once logged in do a google search to find it and it "knows" about your login - credentials generated are for the logged in account.

2. Using Postman or a similar application, create a request as follows:

   URL: `https://accounts.zoho.com/oauth/v2/token`

   Type: `POST`

   Parameters:

   grant_type = `authorization code`

   client_id = `client id` from Step 1 above

   client_secret = `client secret` from Step 1 above

   code = `value` from 'Generate Code' step

3. Use the 'Generate Code' action to create a new grant token or code.

4. Execute the prepared request.

   a) Be sure to enter code and run before grant token expires.

b) Capture *client_id*, *client_secret*, and *refresh_token* values from the request. These values will be needed when configuring BillMax.

Configure BillMax for Zoho CRM

1. Edit the **zoho.pl** provisioning script by setting the values for *client_id*, *client_secret* and *refresh_token*.
2. Edit the **account_hook** (assuming Perl is used). Add `system("$BILLMAXDIR/local/zoho.pl -v -t account");`
3. Edit the **user_hook** (assuming Perl is used). Add `system("$BILLMAXDIR/local/zoho.pl -v -t user");`
4. Set the Zoho list values through the BillMax web interface. If the list is not updated with these values, BillMax will default to values from remote application.

   Virtual Company: `Number` of the Virtual Company used when adding Accounts.

   Profile: `Number` of Profile used when adding Accounts.
5. Edit the **zohosync.pl** script in the same manner as step 1. Run this script from the command line. This step is necessary only if, at the time Zoho is integrated with BillMax, there are already Contacts in Zoho and Accounts in BillMax.

   If using **Copy to BillMax** (see *Configure Zoho* on page 274), the command should be run as `zohosync.pl --setsync` otherwise `zohosync.pl`

Reference

Zoho CRM Log Files

The following is a list of places to check for errors and activity.

`/usr/local/billmax/logs/zoho.endpoint.log` – This is the log for the Edge service that creates or edits Accounts and Users.

`/usr/local/billmax/logs/zoho.provisioning.log` – This is the log for the `zoho.pl` scripts used to "push" changes from BillMax to Zoho.

`/`**"path to portal"**`/storage/logs/*.log` – These log files contain logs for all portal activities. Of particular interest here is the `/zohoContact` route which serves as a "front end" to the Edge Zoho service.

Part

V

Reports

Topics:

Chapter
1

CDR Billing Reports

Topics:

* *Concepts*

Concepts

Chapter

2

FCC 477

Topics:

- *Concepts*
- *How To*

Concepts

Financial Reports 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 Access Points in BillMax 2020 and beyond. For wireless, POP Regions represents towers. Technology and coverage are now indicated on Access Points. For each POP Region there is a section for location information and each Access Point has deployment and speed information. 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 Access Points 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.

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/Package 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. Choose the **Service Type** and **Technology Code**.
4. Enter the **Dowload Rate** and **Upload Rate**. Click Save.

Create POP Regions

Each tower must be entered for the location data for FCC 477 report.

1. Select **System Administration>Pops**, and click **New**.

2. Enter the name or identifier for the tower.

3. Enter the Postal Address for the POP. If the Latitude and Longitude are known, then these values can be entered and then click **Get Census From Location**.

4. For each tower, indicate the type of service: **Cusumer Service** or **Business Service** or both.

5. Choose **Customer Fields** if applicable and enter any **Custom Data** and **Save**.

Create Access Points

Each Access Point must be entered for the coverage data for FCC 477 report.

1. Select **System Administration>Access Points**, and click **New**.

2. Enter the name or identifier for the tower.

3. For each Access Point, enter the **Coverage** and **Technology** data.

4. Choose **Customer Fields** if applicable and enter any **Custom Data** and **Save**.

Validate Data Pass 1

Each **Service/Package** that is to be reported must be connected to an **Access Point** 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:

Error	Issue	Resolution
Invalid census data	**User Address** is not geocoded.	Verify address is valid and country is USA or enter Latitude and Longitude and click **Get Census From Location**.**Save** the **User**.

Generate Coverage

The FCC report requires coverage information from each **POP Region**.

1. Select **Reports** and select **FCC 477**.

2. Check the **Access Points** or click **Select All** on Instruction 4.

3. Select **Generate Coverage** from the menu.

4. Use the **Service Map By Access Point** to make sure that coverage extends beyond all **Service** locations.

5. If coverage needs to be extended, click **Modify** and move the circumference to extend the outer border.

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 Access Point**.

   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 file of CSV files is created to be uploaded to the FCC web site.

Chapter

3

FCC BDC Report

Topics:

- *Concepts*
- *How To*

Concepts

FCC BDC Overview

Beginning September 1, 2022 the FCC requires broadband availability to be reported either by serviceable addresses (the so-called Data Fabric) or by polygons (shapefile) in a BDC filing. Broadband service providers must submit **BOTH** the FCC's Form 477 and the Broadband Data Collection filing. Both reports are due semi-annually on September 1 for the period of January 1-June 30 and March 1 for the period of July 1-December 31.

Coverage data must be generated externally using whatever means deemed appropriate and then provided to BillMax as polygons on a per Access Point (PON etc) basis or at least as frequent as the unique technology and speed specification. These polygons will be associated with the FCC required meta-data (speeds, technology code, etc.) from BillMax or this meta-data may be provided with the polygons. In BillMax the serviceable addresses (Fabric Data) will be loaded and BillMax merge this data spatially to produce the needed CSV report files.

To be able to generate the CSV report files, Service and Package definitions must have valid FCC settings. Users must have a valid Latitude, Longitude and Census values that fall in the coverage area. Towers must have a valid Latitude and Longitude. Access Points must have FCC settings. Coverages have valid FCC meta-data if provided and there must be a coverage for each technology. The Provider ID and Brand Name(if not provided in shape files) must be provided to BillMax. After all data is validated and no errors remain, the FCC BDC CSV files will be generated so that they can be uploaded to the FCC web site.

BillMax Coverages

The term "Coverages", as used by Internet Services Providers, describes the geographical areas that internet service is offered by a provider. Coverages are useful for marketing purposes and also are a critical component necessary for the generation of governmental reports such as the FCC 477 and Broadband Data Collection (BDC) reports.

For the FCC 477 report, BillMax generates coverages using radius, azimuth and beam width settings per Access Point. For the new BDC report, coverages will no longer be created from within BillMax. Instead,coverage must be generated elsewhere using methods suitable to the FCC. Please see *https://www.fcc.gov/BroadbandData/resources* for details.

Beginning with the 2022 Q3 release, BillMax supports the generation of the BDC report. To do so, it uses the Data Fabric method of submittal where the Coverages are specified by addresses (not as census blocks as the FCC 477 requires).

To generate the BDC report in BillMax, the fabric (addresses) data and Coverages (as polygons) must be loaded into BillMax. The fabric data is available under license, free of charge, from Costquest Inc. See*https://www.costquest.com/reporting-obligations/*. Coverages, typically using Radio Frequency (RF) and terrain analysis are available from a number of third-party vendors. Some of which include, Cambium's cnHeat offering, Tower Coverage, Google's Network Planner (beta). BillMax makes no assertions as to the suitability of these products.

Coverages for the BDC, and eventually for other purposes in BillMax, are typically loaded by Access Point. The input for this loading is ESRI shapefiles. Shapefile actually refers to a set of files (up to 6). In BillMax these files are loaded as a ZIP (compressed archive) file.

Coverages outline areas but also are accompanied with meta-data that describe certain attributes for that area. For the BDC report the FCC requires the following attributes: Technology code (wireless,cable, fiber, etc.), Speed, Latency, Customer Type (Residential, Business).

Coverages must be provided for each unique combination of technology, speed, latency, and customer type. Where coverages overlap for a technology, the highest speed rates should be reported.

Ideally, for best functionality, Coverages should be maintained by Access Point. Doing so offers the possibility of spatial applications that involve customer service addresses (for example, the automatic assignment of a service address to an Access Point).

However, coverage data provided by third-party vendors often does not include this data. For this reason, it is possible to load Coverages such that the Access Point references are "virtual" in nature. For example, a provider

offers services via fiber and wireless. Each technology, has three tiers of service(speed). For such a scenario, at a minimum 6 separate shapefiles would be required. All of which could be loaded and associated with a single virtual Access Point (if the shapefiles have all the needed attributes, or via 6 access points with the appropriates settings if the attributes are not available in the files).

FCC Classifications

The BDC report has added two new technology codes to those used in the FCC 477 Report. If these new codes apply to you, you must use these codes on Package/Service definitions and Access Points. These new codes are not applicable to the FCC 477 report. However, the FCC 477 report has been modified so that when a new code is used, it will be translated to the appropriate 477 technology code.

New Technology Codes:

1. Licensed Terrestrial Fixed Wireless (71). Maps to Terrestrial Fixed Wireless (70) for the 477 report.

2. Non-geostationary Satellite (61). Maps to Satellite (60) for the 477 report.

Please consult with the FCC to see if your technology should be reported using these new codes.

How To

Load Fabric Data

For the FCC BDC report, each service provider must request the production fabric for their locations.

1. Contact **nbfsupport@costquest.com** with the name and email of the provider's contact person, the provider's name and the FCC Registrations Number (FRN).
2. Instructions will be provided by CostQuest on how to access the fabric after completing an online form and executing a license agreement with CostQuest.
3. Go to **Reports** > **FCC BDC Report** and click *Load Fabric Data*.
4. Choose the file and click *Load*.

Load Coverage Polygons

Coverage data must be generated externally using a third party. These polygons will be associated with the FCC required meta-data (speeds, technology code, etc.) from BillMax or this meta-data may be provided with the polygons. .

1. Go to **System Administration** > **Access Points**.
2. Choose the number of the Access Point and click *Load Shapefile*.
3. Specify the zip file for the shapefiles. This file can contain (some of which are optional):
   .SHP, .SHX, .PRJ, .DBF, .CPG, .QMD files.
4. Specify *Upgrade Mode*.
   a) Append - This adds to the existing layer with the same Access Point (if other shapefiles have been loaded).
   b) Replace - This deletes the existing layer with the same Access Point and then adds the shapefile to the Coverage.
5. Specify *Attribute Mode*.
   a) *Use POP/AP values where values are missing* - Use the setting (technology, speed, etc.) from the POP and Access Point only if the values are not contained in the shapefile.
   b) *Ignore values in file* - Regardless of the presence of attributes in the shapefile, always use the values from the POP and Access Point.
6. Specify *Attributes*. Identify the names of the attributes in the shapefile. Set the attribute to blank if the attribute is not in the file or the Attribute Mode is 'Ignore'.
7. Repeat for each Access Point or unique technology and speed specification.

Validate Data and Generate BDC Files

1. Go to **Reports** > **FCC Broadband Data Collection**.
2. Choose *Validate Data*.
3. Correct errors listed and repeat until all are resolved. For additional help, please contact *support@billmax.com*.
4. Enter the Provider Id and Brand Name.
5. Choose *Generate BDC Report Data* to generate the CSV files to submit to the FCC.

Chapter

4

Financial Reports

Topics:

* *Concepts*

Concepts

Financial Reports Overview

Account Register

All transactions for an individual account over a date range with Invoices and Billing Statements noted. Settings include the Account Number and a date range.

A/R Aging

The Aging report is an 'As Of Date' report provided to view who owes money at a given time. This report can be run by Virtual Company, Account Class, or Account. It can show past due only, a summary, and have deferred revenue subtracted in support of (Generally Accepted Accounting Principles) - GAAP standards. Name, Email and Phone are included for convenience of follow up.

Billing Exceptions

The Billing Exceptions report can be run by Virtual Company. Packages and Services are listed with a count of Services that are at standard pricing and a count of Services at special pricing. If the count is clicked, a list of accounts is provided.

Bookkeeping

This report by Virtual Company lists Sales, Payments and Bank Deposits, everything needed to update an accounting system. It can be run as Cash or Accrual. From the Context Menu, a *Quickbooks IIF* file or *Netsuite GL* file can be created as well.

> **Tip:** An Option Set is recommended so that the same settings are used every time to run the report with the exception of the date range.

Current Deposits Billed/Held

Deposits are collected money that will eventually be returned to the customer. Typically a Deposit is collected as a surety against a piece of equipment. The report lists all Accounts that have been billed Deposits and the amount of money collected.

Deferred Revenue

This report lists all the prepayment for services yet to be delivered. The revenue can be viewed as straight deferred revenue (prepayment distributed over the time period) or by monthly price (front loaded revenue EXAMPLE: the revenue for a monthly service would appear in the first month even if the service crosses into a second month). This report can be run by Virtual Company, Account, and Classification at the summary or detail level.

Deposits

The report lists all Deposits and Deposit Reversals and whether they have been reversed.

Future Billings

Based on open Services, this report lists all billing that will occur until the specified End Date. This report can be run by Virtual Company, Account, and Classification at the summary or detail level. It can be run for a specific Classification item or All. It can be sorted and grouped by multiple settings.

Gross Receipts

This reports shows what has been invoiced to customers by Account Number and Invoice over a date range for a single Virtual Company or ALL. The report can be run for Accrual or Cash accounting methods and the transactions dates by Effective Date or Post Date.

Payments

This reports shows all Payments collected over a date range including Payment Reversals. This report can be run by Virtual Company, Account, and Branch at the summary or detail level. It can be run for a specific amount, tender type or tender number.

Refunds

This reports shows all Refunds created over a date range including Refund Reversals. This report can be run by Virtual Company or Account at the summary or detail level. It can be run for a specific amount, tender type or tender number.

Roll Forward A/R

The Roll Forward A/R report shows all the amounts that get the A/R from the start of the time frame to end of the time frame. This report can be run by Virtual Company, Account Class, or Account. The report can exclude Deferred Revenue or not. It can be run at the summary level or list all amounts for each Account.

Sales

This reports shows all Sales and Sales Reversals over a date range. This report can be run by Virtual Company, Account, and Classification at the summary or detail level. It can be run for a specific Classification item or All. It can be run as Cash or Accrual (with or without Deferred Revenue). It can be sorted and grouped by multiple settings. It can separate out one time sales, wash sales and taxes.

Sales By Agent

All Sales made over a time period sorted by Sales Agent. One Time Items can be separated out. The report can be run by Virtual Company and Account. It can be run as Cash or Accrual (with or without Deferred Revenue).

Scheduled Electronic Payments

This report is an 'Through Date' report provided to view upcoming expected payments. If your company bills and collects on the same day for Services, this report contains nothing because nothing is scheduled in the future. This report can be run by Virtual Company and Payment Type.

Store Credit Activity

This reports shows all Store Credits and Store Credit reversals over a date range. This report can be run by Virtual Company, Account, and Classification at the summary or detail level.

Write Offs

This reports shows all Write Offs and Write Off reversals over a date range. This report can be run by Virtual Company or Account, and grouped by Classification.

Part

VI

Scheduling and Appointments

Topics:

Chapter
1

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, time slots 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

2

How To

Topics:

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.
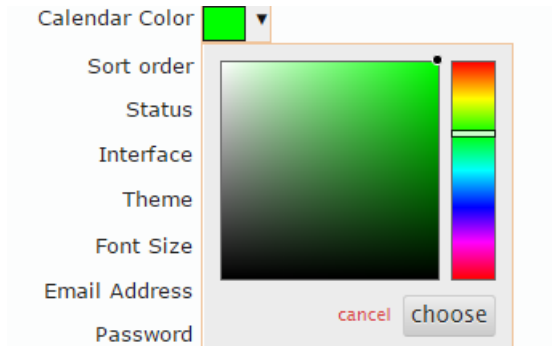
   e) If to be assigned to an appointment, specify one or more **Job Classes**.
   f) If scheduling relevant, make sure **May Schedule on Calendar** is **Yes**.

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 Item**s 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 **Unavailablity** 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 time slot entered for an Assignee.

6. Review the Templates for automated emails via **Correspondence** > **System Email Templates**

   Relevant Templates are:

   - *AppointmentScheduled*

- *AppointmentChange*
- *AppointmentReminder*

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

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.
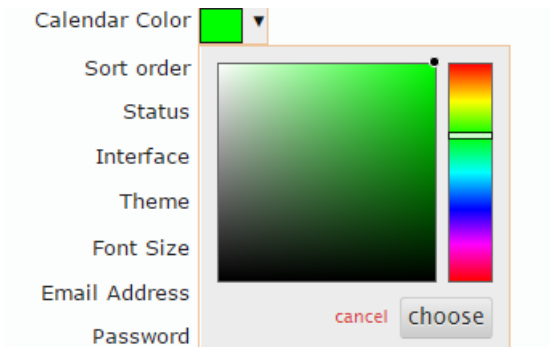
5. Specify one or more **Job Classes**.
6. If scheduling relevant, make sure **May Schedule on Calendar** is **Yes**.

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**.

Part

VII

Ticketing

Topics:

- *Concepts*
- *How To*

Chapter

1

Concepts

Topics:

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 293
- 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 are 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 echeck 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 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

Queue are used to group Tickets based on issue similarity. Examples are *billing*, *sales* and *technicalsupport*. 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

2

How To

Topics:

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 168 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 168 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

VIII

Service Availability Determination

Topics:

- *Fundamentals*

Chapter

1

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.

Out of the box BillMax supports TowerCoverage® . See *http://www.towercoverage.com*.

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 **prequailify** 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 a 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 **Ad Code** 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 321
*   The **Source of Lead** is automatically updated to towercoverage.
*   **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 307), *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 (CentOS 7)* on page 354.
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 303.
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

   yes

   Names and prices of Service Definitions with a both **Sell Online** and **Preview Online** will be shown if the customer can get service.

   SA_additionaldata

   yes

   Additional form fields to capture data will be displayed if the customer can get service. Data that has the attribute *data-note* will be captured as a Ticket Note, User Location Note and Account Note. To modify

what data is captured, edit the blade `/usr/local/billmax/portal.v2/resources/views/BillMax/Portal/TC.blade.php#coverage`.

SA_landing

The Customer Portal REST endpoint after a successful registration. Typically this will be *shop*.

TowerCoverage How To

Configure EUS

1. Login to TowerCoverage.
2. Click Account.
3. Click API.
4. Fill in the following:

a) **API Key**, **Username** and **Password** will be used
b) **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 324.

apiusername

> The *Username* from *Configure EUS* on page 324.

apipassword

> The *Password* from *Configure EUS* on page 324.

testmode

> See *Testing TowerCoverage Setup* on page 325.

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*

3. Enter data. The following in Phone field triggers the following behavior:

Blank

> Pre qualification returns no coverage.

F

Pre qualification returns fiber available only.

W

Pre qualification returns wireless coverage available only.

FW

Pre qualification returns both fiber available and wire coverage.

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
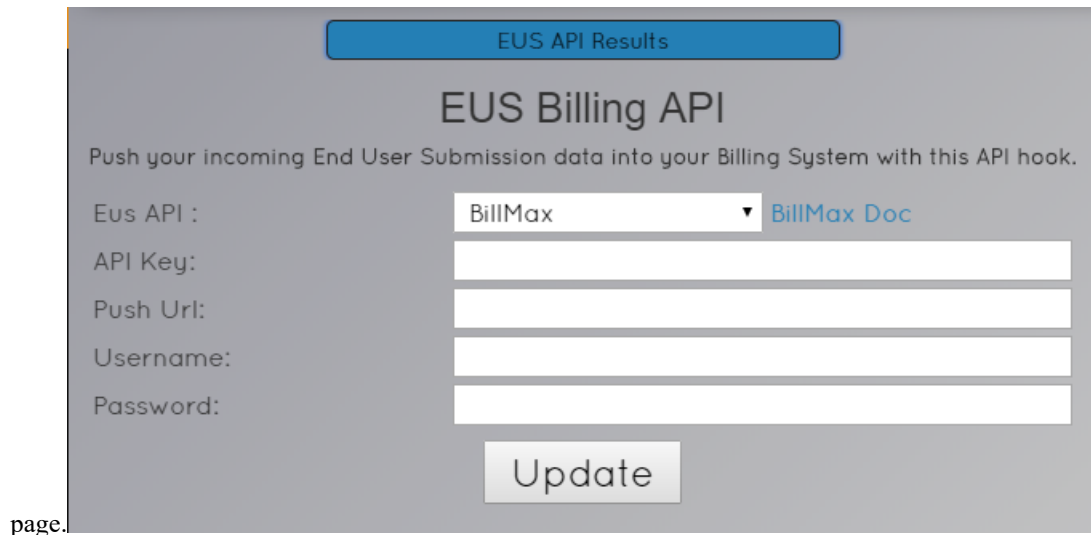
    page.

    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**.

    TCTest

    Set to **false**.

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

IX

VOIP/Telecommunication

Topics:

- *Concepts*
- *How To*
- *Reference*

Chapter
1

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/* 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/* 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/* 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/* 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/* will be done using *8174467776*.

The Rate Deck specification for BillMax may be found here *BillMax CDR Rate Deck* on page 346.

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 prepended 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 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*

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*. 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.

Plan FUSF example

1. Assume an FUSF assessment of 20%.
2. A plan has 1000 Plan minutes free with 10 cents a minute overage.
3. A customer uses 2000 Plan minutes resulting in an overage billing of 1000 minutes, or $100.00.
4. Based on CDR data, of the Plan minutes 65% were not interstate/international and 35% were interstate/international.
5. The customer would see 3 charges on their bill:

   a. The $65 for overage Plan minutes, not interstate/international.
   b. The $35 for overage Plan minutes that were interstate/international.

> **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 175.

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

2

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 344 and *Create a State Universal Service Fee* on page 344.

3. If using a rate deck, upload the rate deck into BillMax. See *BillMax CDR Rate Deck* on page 346 and *Upload a CDR Rate Deck* on page 340.

4. Create a CDR Billing plan. See *CDR Billing Plans* on page 336 and *Create a CDR Billing Plan* on page 341. Set **Billing Suspended** to **Yes**.

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 341.

6. Add a new CDR Source. See *Create a CDR Source* on page 341. 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 332, *CDR Preprocessing* on page 334, *Create Package/Service Association for a CDR* on page 341 and *BillMax CDR Input Format* on page 346. 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 342.
   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 343.
    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. Choose how the CDRs are to be classified for USF attributes (interstate, international, intrastate) with **Classify**.

6. 📝 **Note:** In order for CDRs to be displayed on the customer's bill, **Rate CDRs** must be not be No.

    Choose whether the CDRs are to be rated or if they come pre-rated with **Rate CDRs**.

7. 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.

1. Copy `/usr/local/billmax/local/processCDRFile-dist` to `/usr/local/billmax/local/processCDRFile`.

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

```
    <xsl:if test="noteSet">
     <fo:block>
      <xsl:text>Notes: </xsl:text>
     </fo:block>
     <xsl:for-each select="noteSet/note">
      <fo:block linefeed-treatment="preserve"   white-space-collapse="false"  white-space-
treatment="preserve" font="{$notefont}">
       <xsl:value-of select="text"/>
      </fo:block>
     </xsl:for-each>
    </xsl:if>
    <xsl:if test="not(//phoneUsageTables)">
     <fo:block id="endofdoc"/>
    </xsl:if>
   </fo:flow>
  </fo:page-sequence>
  <xsl:if test="//phoneUsageTables">
   <fo:page-sequence master-reference="phoneUsage">
    <fo:static-content flow-name="phoneusage-region-before" font="{$defaultfont}">
     <xsl:call-template name="standardUsageHeader"/>
    </fo:static-content>
    <fo:flow flow-name="xsl-region-body" font="{$defaultfont}">
     <xsl:for-each select="//phoneUsageTables">
      <xsl:call-template name="phoneUsageTables"/>
     </xsl:for-each>
     <fo:block id="endofdoc"/>
    </fo:flow>
   </fo:page-sequence>
  </xsl:if>
 </fo:root>
```

After

```
    <xsl:if test="noteSet">
     <fo:block>
      <xsl:text>Notes: </xsl:text>
     </fo:block>
     <xsl:for-each select="noteSet/note">
      <fo:block linefeed-treatment="preserve"   white-space-collapse="false"  white-space-
treatment="preserve" font="{$notefont}">
       <xsl:value-of select="text"/>
      </fo:block>
     </xsl:for-each>
    </xsl:if>
    <fo:block id="endofdoc"/>
   </fo:flow>
  </fo:page-sequence>
 </fo:root>
```

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

3

Reference

Topics:

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.

Column	Description
Description	Descriptive value of destination string.
Destination	A string representing the destination of the call. For numbers part of the North American Number Plan Administration (NANPA), all strings should begin with "1". Examples are: • 1817 - Fort Worth, Texas • 1905 - Ontario, Canada • 1817446 - further refinement of Fort Worth, Texas The longest string matched will be used.
Rate	Rate per minute. Specified in fractional minimum monetary units. Cents for USA. Support for up to 10 decimal places.
Minimum Duration	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.
Increment	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.
Connect Fee	The amount, specified in dollars and cents, that will be assessed for every completed call to this Destination.

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.

Field	Description
Source	The original source/inbound of the call.
Destination	The original destination/outbound of the call.
Source Channel	The source/inbound channel or path of the call.
Destination Channel	The destination/outbound channel or path of the call.
Call Start	The time stamp of the start of the call - UTC
Call End	The time stamp of the end of the call - UTC
Billable Seconds	must be non-zero.

Field	Description
Source Search Key 1	The first source/inbound value to try to match to the Package/Service d01 value.
Source Search Key 2	The second source/inbound value to try to match to the Package/Service d01 value.
Destination Search Key 1	The first destination/outbound value to try to match to the Package/Service d01 value.
Destination Search Key 2	The second destination/outbound value to try to match to the Package/Service d01 value.
Destination Rate Deck Key	The destination/outbound value to match to a value in a Rate Deck.

Flags	A bit mask denoting what type of call the CDR represents:
	2
	    Operator assisted
	4
	    International
	8
	    N11 call
	16
	    Toll Free
	32
	    Local Call
	1024
	    Premium Call

Unique ID	A unique id for the record. Combined with Start of Call in the BillMax database to create a unique key across sources.

Status	Status of call.
	0
	    Call is in progress.
	1
	    Call is complete.

Time Call Answered	The time stamp of when the call was answered - UTC
Record Number	Record number of CDR in current file/dataset.

Direction of Call	**0**
	    Not specified in CDR. BillMax will derive it.
	1
	    Call is outbound from Customer.
	2
	    Call is inbound to Customer.

The following may be used for pre-rated and pre-classified CDRs.	
Rated Amount	Rated amount based on duration of call.

Field	Description
Call Type	**0**
	Not specified in CDR. BillMax will derive it. Typically used for location less numbers such as Toll Free and Premium numbers.
	1
	Process as a Plan call.
	2
	Process as a Long Distance call.
	5
	A local call processed as an N11 call.
Connect Fee	Connect Fee to bill.
Billed Seconds	Billed Seconds for the supplied Rated Amount.
Source Description	Description of the Source of the call.
Destination Description	Description of the Destination of the call.
USF Region	The region/state to which the call/revenue/minutes should be attributed for Federal USF and State USF purposes.
USF Class	**0**
	Not relevant for USF purposes. In general outbound toll free, premium numbers and non billed inbound calls should have this value.
	1
	Intrastate.
	2
	Interstate.
	3
	International.

VoIP Glossary

Call Detail Record
See *https://en.wikipedia.org/wiki/Call_detail_record*

Call Detail Record (CDR)

CDR

Part

X

CentOS Documentation

Topics:

Chapter

1

CentOS 7 Documentation Items

Topics:

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

This documentation is for customers still on CentOS. The latest releases of BillMax are on Debian and other sections of documentation should be referenced.

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 353. 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 the second *'/usr/local/billmax* below 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.

9. Execute `mysql_upgrade` -u root -p on the new server.

10. Redo any local customizations that were noted in `upgradeddifferences.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.

15. Copy `.fetchmailrec` from BillMax system user to the new machine BillMax home directory.

16. To reset the `crontab` entries on the new server, go to **System Administration** > **Batch Processing**. Change the `nightly` batch process from ACTIVE to DISABLE and SAVE. Set the `nightly` batch process back to ACTIVE.

Install CentOS 7.x® for BillMax

Describes a basic installation of Cent OS for BillMax. Specifics such as network interface name, host name, IP address, etc. will differ. 64 bit architecture is required. **This documentation is for customers still on CentOS. The latest releases of BillMax are on Debian and other sections of documentation should be referred to.**
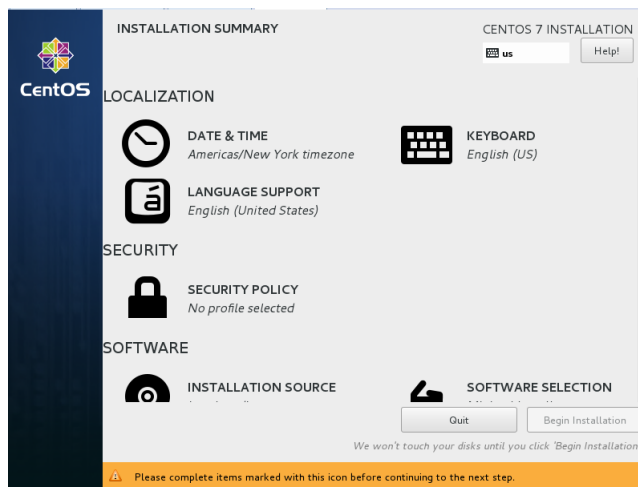
1. Retrieve from *http://www.centos.org* the latest version of the Minimal ISO 64 bit for Cent OS 7.

2. Start the installation

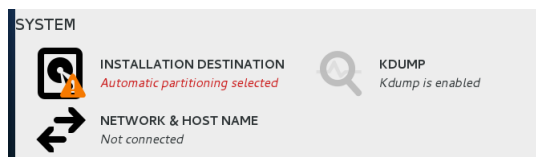3. Choose **Test this media & install CentOS 7**.
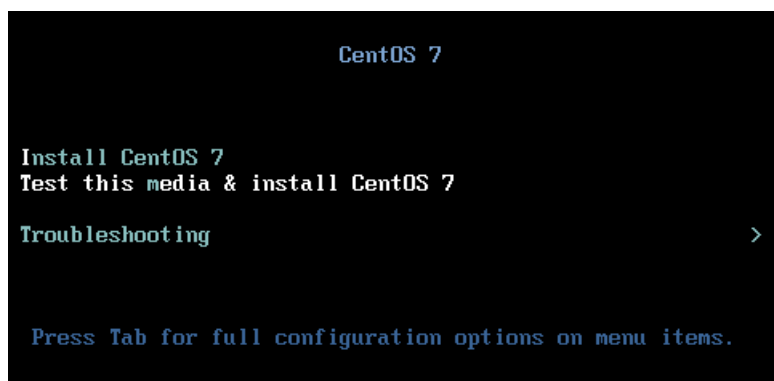

4. Click **Continue**.
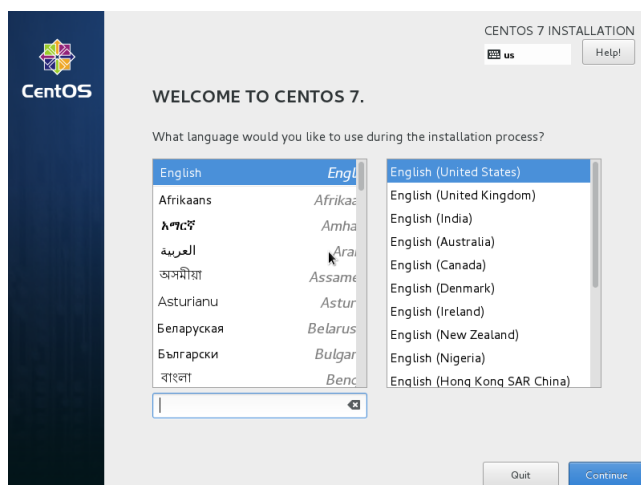
5. Click **Next**.

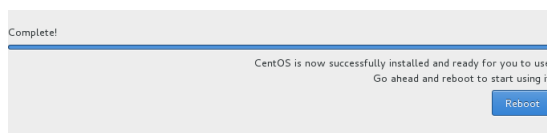6. Click **Date & Time** to set the Time Zone.

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.
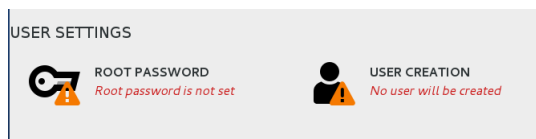
11. Click **Reboot**.

⚠   **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.

Setup the Customer Portal (CentOS 7)

This documentation is for customers still on CentOS. The latest releases of BillMax are on Debian and other sections of documentation should be referenced.

1. Generate a SSL certificate to be used for encryption of data between the Customer Portal and the Customer Portal daemon (Edge). As root perform `/usr/local/billmax/bin/setup_edge`
2. 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 CentOS 7, `wget -q` *http://rpms.remirepo.net/enterprise/remi-release-7.rpm*.

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 php-soap mod_php`

f) Execute `systemctl restart httpd`

3. Generate a new application encryption key:

   a) `cd /usr/local/billmax/portal.v2`

   b) `php artisan key:generate`

4. 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.

5. `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.

6. Start the edge service. As the root user:

   a) `systemctl start bx_edge`

7. Test functionality by pointing a browser at `https://<billmaxserver>/testportal` where *<billmaxserver>* is the address of the BillMax server. A view similar to what is shown indicate success. If you get a permission error, check `zbillmax.conf` in `/etc/httpd/conf.d`. In the `/usr/local/`

`billmax/html` section, change *AllowOverride None* to *AllowOverride All* and restart the Apache server.
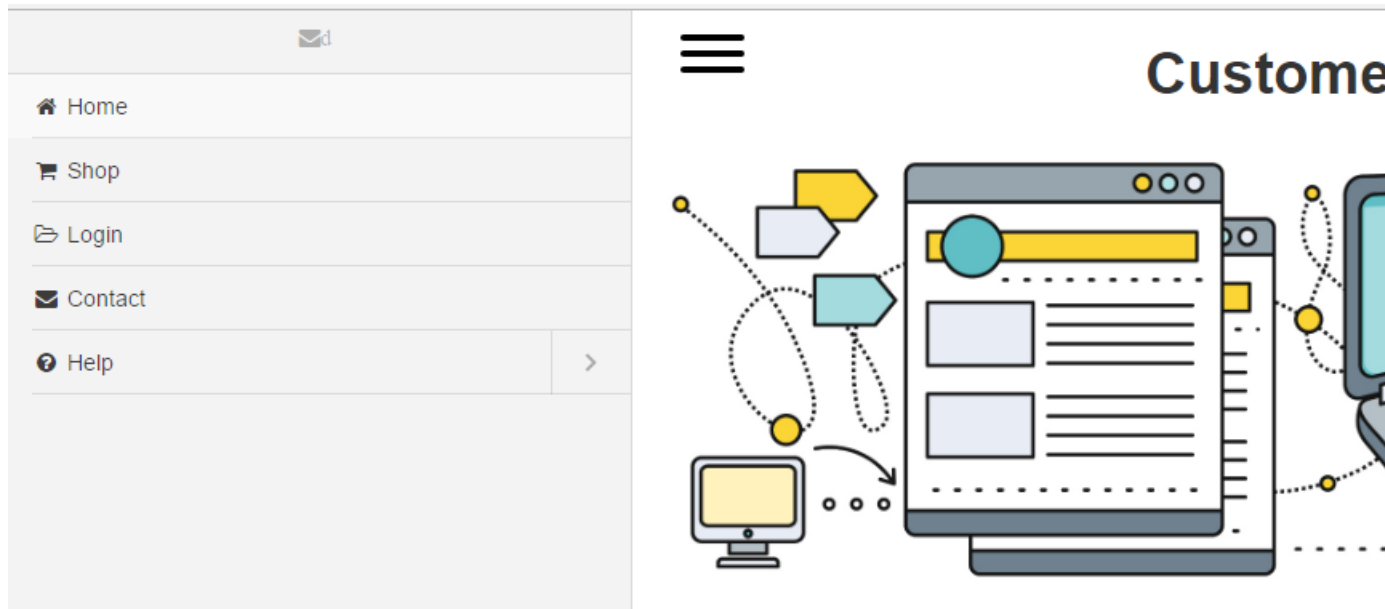
8. Click **Login** to test access to the `edge` service.

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 include the period at the end of this command.**

   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**.

12. When installing the portal at a remote site, ensure that the Apache server allows the use of `.htaccess` files by using the "AllowOverride All" directive in the appropriate <Directory> section for the portal in the `httpd conf` file.

Upgrade BillMax on CentOS

This documentation is for customers still on CentOS. The latest releases of BillMax are on Debian and other sections of documentation should be referenced.

1. Download the BillMax RPM.

   a) Using a web browser, go to *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 RPM.

   d) Copy the RPM to the CentOS machine.

2. Login as root.

3. 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`.

4. Make backups. See *Backups, Redundancy and Archiving* on page 163

5. 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:`"

6. 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.

7. `cd` /opt/billmax/install. Execute `./BxUroot-4.0.pl`

8. `su` to the *billmax* system user - typically "*billmax*"

9. `cd` /opt/billmax/install. Execute `./BxUpost-4.0.pl`

10. Test the interface.

11. Start any BillMax processes that were stopped in step 2.