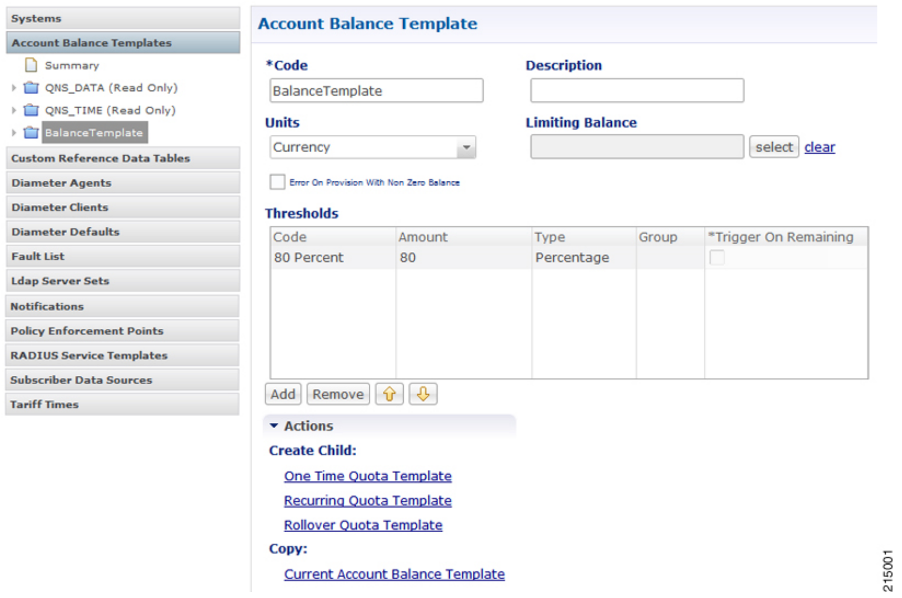
Balance Services

Account Balance Templates

Account Balance templates provide the overall structure to the data provisioned to a given subscriber.

Figure 1: Account Balance Template



Account Balance Template

*Code	BalanceTemplate
Description	
Units	Currency
Limiting Balance	
Error On Provision With Non Zero Balance	☐

Thresholds

Code	Amount	Type	Group	*Trigger On Remaining
80 Percent	80	Percentage		☐

The following parameters can be configured under Account Balance Template:

Table 1: Account Balance Template Parameters

Parameters	Description
Code	Required unique name for the template.
Description	Optional field to contain a brief description of the template's use case.
Units	The choice of units determines functionality options within the system. For example, Time units such as seconds or minutes will cause the system to behave differently than Data units like Bytes or Megabytes. Additionally, currency is an option and can be used to account for usage credit in a direct manner. **Note** Balance does not do any type of currency exchange rate calculation. The values are stored as is and represent whatever currency the SP and their subscribers commonly use. Default value is Bytes.
Limiting Balance	Limiting Balance refers to a Balance template that is used by a shared balance template. This establishes a link from the shared balance to a “limit” balance, so that Balance Manager knows which two balance codes it needs to reserve/charge against in the shared per user limit use case. **Note** The limiting MsBM account must be the MsBM account tied to the individual subscriber's credential. The limiting MsBM balance and quota must be provisioned in separate Balance/MsBM operation from the provisioning of the shared account, balance, and quota.
Error on Provision With Non Zero Balance	If a provisioning request is made (specifically any request that credits or provisions a subscriber balance) when there is remaining balance, i.e. non-zero amount, then the Balance module throws an error and does not provision the quota. Default value is False (unchecked).
Thresholds	
Code	Unique name for the threshold object.
Amount	An integer representing the amount of quota that will trigger the threshold notification.
Type	Unit of calculation like Percentage or Bytes.
Group	Thresholds can be associated with each other as a group. When thresholds are grouped by name, only messages for the first (top to bottom in the table in Policy Builder) threshold breached in the given threshold group will be returned.
Trigger on Remaining	This inverts the threshold function. Typically a threshold is calculated against the usage. For example, if a threshold is defined for 80%, by default that means 80% of quota used or 20% remaining. If the Trigger on Remaining check box is selected, then the function inverts and a threshold defined as 80% would trigger when 80% of the quota remains.

CRD Based Account Balance Templates

CRD balance table definitions are pre-shipped (read only) as part of the Policy Builder configuration. For this, the "CRD Balance" feature (`com.broadhop.balance.crdbalance.feature`) needs to be installed. Enable this functionality by selecting the **Enable Crd Balance Template Lookups** check box. For more information, see Balance Configuration.

The following required tables include result columns that directly map to the corresponding fields in Balance and Quota Templates. For more information on individual field description, see Account Balance Templates, on page 1.

- q_account_balance: Top Level Account Balance Template fields
- q_one_time_quota: One Time Quota Template definition
- q_recurring_quota: Recurring Quota Template definition
- q_rollover_quota: Rollover Quota Template definition
- q_threshold_definition: Threshold definitions
- q_threshold_balance_association: Balance to threshold association
- q_threshold_quota_association: Quota to threshold association

Threshold Priorities and Groups

Thresholds defined in Policy Builder (under Balance and Quota templates) have implicit top to down priority incase threshold group is defined. Same functionality is achieved using 'priority' column in `q_threshold_definition` table. Internally, thresholds defined in CRD are prioritized based on provided 'priority' column value. Higher the value, higher the priority (that is, higher in the Policy Builder list). Priority value has no effect if thresholds are not grouped.

Quota Templates

Quota templates define the specifics of how quota behaves. There are 3 basic types of quota: One Time, Recurring, and Rollover. Within that there are additional behavioral functions like BillCycle and Stackable, but those are just modifications to one of the three basic types.

Recurring

Recurring quota is refreshed periodically with a specific amount each refresh. It defaults to infinite duration meaning that it continues until the account is deleted from the system. However, it is possible to limit the duration of a recurring quota using the Recurrence Limit field. The most common time period is Monthly. Time periods defined in hours, days, weeks, months are possible. Initial credit and refreshed amounts by default expire at the end of the current time period.

The following parameters can be configured under Recurring Quota Templates:

Table 2: Recurring Quota Templates Parameters

Parameters	Description
Code	Unique name that identifies the quota template.
Description	Optional field to contain a brief description of the template's use case.
Amount	A default provisioning amount which can be overridden at the initial provision time via API or Policy configuration. **Note** • The upper limit on the amount is 1 Exabyte. • Future amount changes can be accomplished with the Credit API.
Priority	Priority ranks the template so when the Balance module is determining the next credit to use for reservations and debits, the template with the highest rank (Positive number Integer) wins. 1 is the highest rank. The default of no value is lowest priority. After priority, the most recent end date (Next to Expire) is used to determine the next credit. Default value is null.
Recurrence Frequency Amount	Integer used in conjunction with the Recurrence Frequency to determine the refresh period. Default value is 1. **Note** Recurrence Frequency Amount range must be between 1 – 12 when the Recurrence Frequency is configured as **Bill Cycle**. CPS takes the default value **1** when Recurrence Frequency is not configured.
Recurrence Frequency	Value used in conjunction with the Recurrence Frequency Amount to determine the refresh period. You can select the value from the drop-down list based on your requirements. The following values are supported: • Bill Cycle : Refresh interval is based on bill cycle day for each quota independently. • Day(s): Refresh interval is based on day(s). • Hour(s): Refresh interval is based on hour(s). • Minute(s): Refresh interval is based on minute(s). • Month(s): Refresh interval is based on month(s). • Week(s): Refresh interval is based on week(s). Default value is Month(s).

Parameters	Description
Rollover Quota	A Rollover Quota Template that this recurring quota rollovers unused quota to when the quota refreshes for the next recurrence period.
Calendar Type	MsBM supports both the Gregorian and Hijra calendar. The Hijri calendar is the Islamic calendar which is a moon-phase based calendar. Cisco has several customers in the Middle East who use the Hijri calendar instead of the Gregorian calendar to determine refresh dates. **Note** The data is still stored in the database as Gregorian dates, but the Balance module translates those to Hijri for any processing. SPR and the Unified API do not support Hijri dates. Default value is Gregorian.
Recurrence Limit	Integer that determines the duration for a recurring quota. When set to 0, the duration is infinite. When set to any positive number, the quota refreshes that number of times and then stop. For example, if the Recurrence Frequency is set to 1 Month, and the Recurrence Limit is set to 6, then the quota refreshes 6 times. If the quota is provisioned on January 1st, it expires on June 30th. Default value is 0.
Auto Rollover	When selected, automatically roll unexpired quota over into a Rollover quota when the refresh occurs. **Note** When not checked then rollovers can only be triggered by using the RolloverCredit API. **Note** If checked, the Recurrence Frequency for the quota must be >= 1 day. Default value is false (unchecked).
Align ROQ Validity Period With RQ BillCycle	When selected, CPS aligns the rollover quota validity period with recurring quota billing cycle. CPS ignores the Validity Period settings in the linked Rollover Quota Template and uses the next billing cycle date of RQ for Validity Period of ROQ. Default value is false (unchecked).
Use Rollover Expiration Time for Charge Priority	When selected, the Balance module uses the sum of recurring quota template's credit end date and the rollover credit's end date to determine priority for which credit to debit in the normal processing of charges. Default value is false (unchecked).
BillCycle Per Quota	When selected, BillCycle on quota level is used. When not selected, BillCycle on account level is used. Default value is false (unchecked).
Thresholds	

Parameters	Description
Code	Unique name for the threshold object.
Amount	An integer representing the amount of quota that triggers the threshold notification.
Type	Unit of calculation like Percentage or Bytes.
Group	Thresholds can be associated with each other as a group. When thresholds are grouped by name, only messages for the first (top to bottom in the table in Policy Builder) threshold breached in the given threshold group are returned.
Trigger on Remaining	This inverts the threshold function. Typically a threshold is calculated against the usage. For example, if a threshold is defined for 80%, by default that means 80% of quota used or 20% remaining. If the Trigger on Remaining check box is selected, then the function inverts and a threshold defined as 80% would trigger when 80% of the quota remains.

Note All the dates in Balance such as start, expiration, refresh, etc. have a time element. What is set for the time element affects expiration and refresh time on the given day.

Refresh Dates

There are two important dates - Last Recurring Refresh (LRR) and Next Refresh. The LRR is used to calculate the Next Refresh. The LRR is the value stored in the database while the Next Refresh is the value that is calculated during processing and is returned in API responses.

The LRR is set to the provision date by default. For a monthly recurrence frequency that means, if provisioned on the 12th, it will refresh again on the 12th of the next month. The LRR can be overridden in a provisioning request (CreateBalance API). When creating quota with the CreateBalance API, set the LRR date to the day when the refresh would have occurred had the quota existed. For example, if the CreateBalanceRequest is sent on 01/01/2012 at 08:00:00 (January 1st, 2012) and the intention is to have the quota refresh on the 28th of the month, then the LRR (lastRecurringRefresh) should be set to 28/12/2011T00:00:00 (December 28, 2011) in the request. The Balance engine uses the LRR to calculate the Next Refresh date, so by setting the LRR to December 28th (the previous month in relation to the provision) the new refresh date of January 28th, 2012 will be calculated correctly. Please note that months have a variable amount days and will refresh accordingly.

Note Valid date formats for API requests are explained in the Unified API documentation. Contact your Cisco Technical Representative for the API documentation.

Manual LRR Override

When overriding the LRR via API, make sure that the start date and end date align properly. That is, the end date must be the same date as what the Next Refresh date would be (LRR + recurrence frequency) when calculated by the Balance engine. This means that the provisioned credit will end when the new credit is created via the refresh which is how the system operates by default.

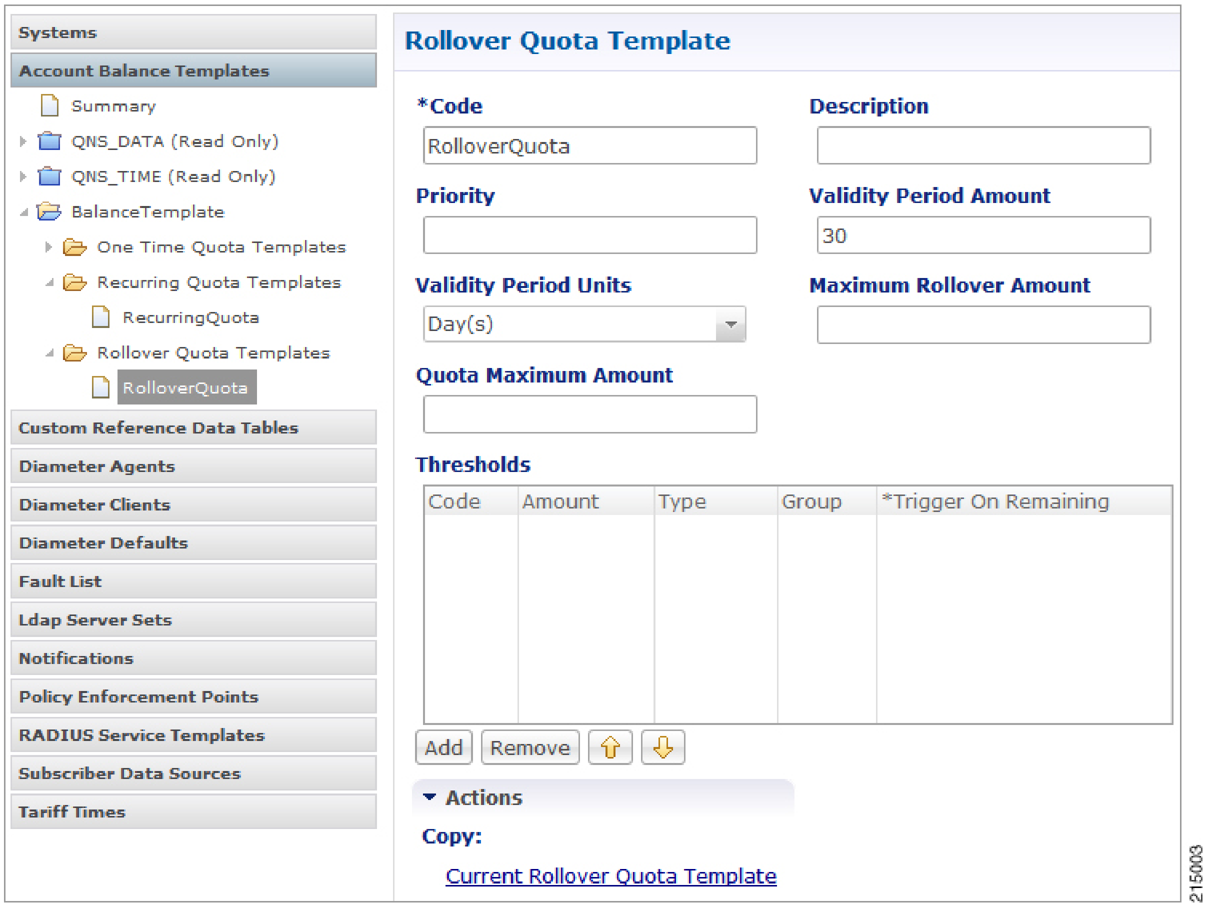
The refresh occurs on the next Balance action instead of on the actual Next Refresh date so that not all subscriber accounts refresh at the exact same moment, thus balancing load and resources. However, it should be noted that the date of the new credit created by the refresh will still have its dates based on the stored LRR and not on when it is actually refreshed by the Balance engine. The new credit will have a start date equal to the new LRR after the refresh has occurred. The new credit end date will be the start date + recurrence frequency. This value is also the new Next Refresh Date.

Rollover

Rollover quota templates are special quotas that store leftover amounts from a Recurring quota. Rollover occurs when the Recurring quota refreshes. Rollovers can also be triggered manually via API. The amount to rollover can be limited, and the total amount in the rollover quota can be limited.

Rollover quota templates behaves like One Time quota templates, but should not be provisioned directly. Unlike One Time quotas, Rollover quotas have no default/initial amount.

Figure 2: Rollover Quota Template



Rollover Quota Template

*Code	RolloverQuota
Description	
Priority	
Validity Period Amount	30
Validity Period Units	Day(s)
Maximum Rollover Amount	
Quota Maximum Amount	

Thresholds

Code	Amount	Type	Group	*Trigger On Remaining

The following parameters can be configured under Rollover Quota Template:

Table 3: Rollover Quota Template Parameters

Parameters	**Description**
Code	Unique name that identifies the quota template.
Description	Optional field to contain a brief description of the template's use case.
Amount	A default provisioning amount which can be overridden at the initial provision time via API or Policy configuration. **Note** Future amount changes can be accomplished with the Credit API.
Priority	Priority ranks the template so when the Balance module is determining the next credit to use for reservations and debits, the template with the highest rank (Positive number Integer) wins. 1 is the highest rank. The default of no value is lowest priority. After priority, the most recent end date (Next to Expire) is used to determine the next credit. Default value is null.
Validity Period Amount	Integer used in conjunction with the Validity Period to determine the length of time for which the quota is valid. Default value is 30.
Validity Period Units	Value used in conjunction with the Validity Period Amount to determine the length of time for which the quota is valid. Default value is Days.
Maximum Rollover Amount	The maximum amount of quota that can be rolled over at any one time.
Quota Maximum Amount	The total amount of rollover the quota can contain.
Thresholds	
Code	Unique name for the threshold object.
Amount	An integer representing the amount of quota that will trigger the threshold notification.
Type	Unit of calculation like Percentage or Bytes.
Group	Thresholds can be associated with each other as a group. When thresholds are grouped by name, only messages for the first (top to bottom in the table in Policy Builder) threshold breached in the given threshold group will be returned.
Trigger on Remaining	This inverts the threshold function. Typically a threshold is calculated against the usage. For example, if a threshold is defined for 80%, by default that means 80% of quota used or 20% remaining. If the Trigger on Remaining check box is selected, then the function inverts and a threshold defined as 80% would trigger when 80% of the quota remains.

Rollover Quota Example

Assumptions:

- Assume the parent Balance Template's units are Megabytes.
- Assume the Maximum Rollover Amount is 100 MB.
- Assume the Quota Maximum Amount is 2048 MB (or 2 GB).
- Assume the current balance of the rollover quota is 1.95 GB.
- Assume the unused usage at recurring quota refresh time is 200 MB.
- Assume the Auto Rollover checkbox is checked.

Function:

- The recurring quota has 200 MB, but only 100 MB is allowed to be rolled over because the Maximum Rollover Amount is set to that value.
- Rolling over 100 MB would cause the total amount of the rollover quota to exceed 2 GB (Quota Maximum Amount is set to 2048 MB).
- Therefore, 2 GB - 1.95 GB = 50 MB, which is the amount that is actually rolled over.

Limitations and Restrictions

Rollover Quotas may experience undesirable behavior when used in conjunction with Recurring Quotas that have a recurrence frequency of less than 1 day.

The recurring quota and rollover quota involved in the rollover operation must be defined under the same Balance template. Rolling over from one Balance template to another Balance template is not supported.

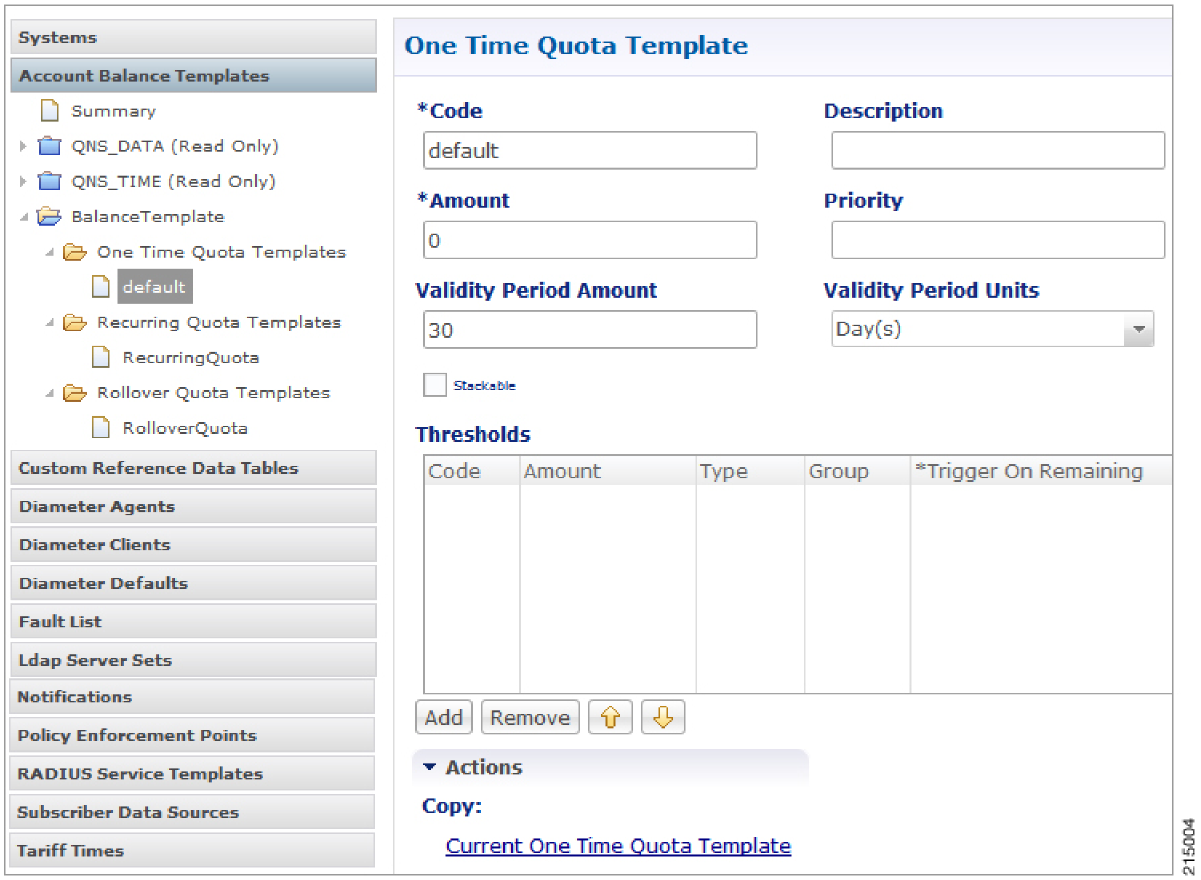
Note Do not provision rollover quotas using the Control Center. Even though Rollover quota is a special type of One Time quota, they are not designed for manual provisioning. They are designed to work with a Recurring quota and receive credits only based on the unused amounts rolled over from that Recurring quota to which they are linked.

Adjustments can be made to Rollover quota via the Credit or Debit APIs, but this is not a typical or common use case, and is not recommended by Cisco.

One Time

One Time quota templates are used for one time applications like TopUp or Bonus quota that has a finite duration (start and end date) and amount. One Time quota does not refresh automatically.

Figure 3: One Time Quota Template



*Code: default
Description:
*Amount: 0
Priority:
Validity Period Amount: 30
Validity Period Units: Day(s)
Stackable: [ ]

The following parameters can be configured under One Time Quota Template:

Table 4: One Time Quota Template Parameters

Parameters	Description
Code	Unique name that identifies the quota template.
Description	Optional field to contain a brief description of the template's use case.
Amount	A default provisioning amount which can be overridden at the initial provision time via API or Policy configuration. **Note** Future amount changes can be accomplished with the Credit API.
Priority	Priority ranks the template so when the Balance module is determining the next credit to use for reservations and debits, the template with the highest rank (Positive number Integer) wins. 1 is the highest rank. The default of no value is lowest priority. After priority, the most recent end date (Next to Expire) is used to determine the next credit. Default value is null.

Parameters	Description
Validity Period Amount	Integer used in conjunction with the Validity Period to determine the length of time for which the quota is valid. Default value is 30.
Validity Period Units	Value used in conjunction with the Validity Period Amount to determine the length of time for which the quota is valid. Default value is Days.
Stackable	When selected the One Time quota becomes "stackable" which is explained Stackable Quota or MsBM Multiple Prepaid Plans. The general idea is that it is possible to provision a Stackable Quota multiple times, but only one instance will be active at any given time. The other instances will "stack up" or queue behind the active one waiting to be used. Essentially, it's a different way to configure priority of credit usage. Default value is False (unchecked).
Thresholds	
Code	Unique name for the threshold object.
Amount	An integer representing the amount of quota that will trigger the threshold notification.
Type	Unit of calculation like Percentage or Bytes.
Group	Thresholds can be associated with each other as a group. When thresholds are grouped by name, only messages for the first (top to bottom in the table in Policy Builder) threshold breached in the given threshold group will be returned.
Trigger on Remaining	This inverts the threshold function. Typically a threshold is calculated against the usage. For example, if a threshold is defined for 80%, by default that means 80% of quota used or 20% remaining. If the Trigger on Remaining check box is selected, then the function inverts and a threshold defined as 80% would trigger when 80% of the quota remains.

Stackable Quota or MsBM Multiple Prepaid Plans

The unique feature of Stackable Quota is that although a quota instance is provisioned it does not get used until the subscriber activates it via their network usage. Stackable quota does not expire if it is not used. For example, if a subscriber has an active plan and purchases a Stackable quota package. That package will never expire as long as the subscriber's current active plan stays active and has valid quota. Once the first plan expires, only then will the Stackable quota be activated and used.

Note Once a credit on a stackable quota is active, any changes made to the template validity period will not have an effect.

Priority

A Stackable quota will not activate until it is needed. This is most important in cases where Stackable and non-stackable quotas are mixed under the same Account Balance. For example, if a non-stackable quota is selected first based on Priority and the Next to Expire rules, the Stackable quota will not be activated until the non-stackable quota exhausts.

Pre-Paid Data Example

A subscriber purchases 5 pre-paid blocks of data quota with a default amount of 100MB and a validity period of 10 days. When the subscriber connects, the first instance becomes active, meaning the start date is set to the current date and time and the end date is set to 10 days later. So if the subscriber connected on January 1st, the quota became valid until January 11th (10 days from January 1st). After the subscriber uses all 100 MB or the 10 days passes, the next instance of quota is activated with the start/end dates set in the same manner - the start date is the current time at activation and the end date is set to 10 days from that time.

Pre-Paid Time Example

A subscriber purchases a time limit package that limits both "wall clock time duration" (calendar time since the package was bought) and volume of fair use quota. The package does not renew automatically, however the subscriber is able to purchase additional pre-paid plans prior to the expiration of the current package they have. Each pre-paid package will automatically start upon expiration of the previous plan just as in the data example. Like the data example, if the time limit is reached, the next package becomes active. If a subscriber reaches the volume of fair use quota limit, the current plan expires and the next plan becomes active regardless of the time remaining on the previous package. If there are no additional pre-paid plans available upon expiration of the current active package, the subscriber is redirected to a self-care portal and offered more options to purchase packages.

Provisioning

Provisioning a Stackable quota sets the start time to the current system time by default, and if a start date value is passed in, it is set to the passed in value. If the start date passed in is in the past and another Stackable quota is currently active, the new quota will not be used until the currently active Stackable quota is exhausted.

Debits and Reservations

As the accounting functions operate, reservations check for active credits. When a credit expires, the system automatically looks to find the next credit based on various criteria including the next most recent expiration date. If the found credit is part of a Stackable quota and is not currently active, the system will activate it by setting the start date to the current date and time and setting the end date to the start date plus the validity period.

If it is necessary to activate a second stackable quota to satisfy the requested reservation amount, even if you release the reservation (charge zero or less than what is remaining on the first quota), the system will maintain two active Stackable quotas.

If no quota is active for a subscriber, a Stackable quota will not get activated until a reservation is made.

QueryBalance API

The QueryBalance API displays all credits whether the Stackable quota is active or not. The API does not provide an indication of whether a quota is Stackable.

Template Definition Changes

If a quota template is changed from stackable to not stackable or from not stackable to stackable, any credits for quotas of that quota code provisioned/credited prior to this template change will behave in the following manner:

- From Stackable to Not Stackable: Any credits on quotas of that quota type that have already been provisioned/credited will have those existing credits behave as a normal one time quota's credits with no expiration date regardless of any set validity period. Future credits will have their end dates set by the validity period.
- From Not Stackable to Stackable: Any credits on quotas of that quota type that have already been provisioned/credited will have those existing credits behave as a normal one time quota's credits with the start date of the provision date or the start date that was passed in if it was specified and the end date that was specified or if not specified the start date plus validity period at provision time. Any future credits will be treated as stackable credits on a stackable quota.

BillCycle

BillCycle quotas were introduced in Balance 2.3.0. BillCycle is a special type of Recurring quota that handles end of month refresh dates better than the typical Recurring quota template. The Bill Cycle functionality aligns better with some customers' billing cycles and removes the recommended limitation of only using days 1 - 28 for Recurring quota starts/ends.

Note The “RFAmt ignored” hint that appears on BillCycle in Policy Builder is just a reminder that the Recurrence Frequency Amount field is ignored if you select a Recurrence Frequency of BillCycle. Refresh happens every 1 BillCycle regardless. The system cannot wait 2 or more BillCycles before refreshing.

Updating BillCycle

The ChangeBillCycle API is the only way to change the BillCycle value for a subscriber.

Repurposing Recurring Quota Templates

It is possible to use BillCycle by repurposing a currently existing Recurring quota that has a recurrence frequency other than BillCycle. When repurposing an existing Recurring quota template and changing it to BillCycle, existing subscribers will have the BillCycle value set automatically at the next refresh time to the day that the quota refreshes. For example, if a subscriber’s quota is scheduled to refresh on the 25th, he/she will continue to use quota until the refresh date as normal. When the quota refreshes on the 25th, the BillCycle value will be set to 25, and the subscriber’s quota will now follow the BillCycle frequency rules instead of the previous recurrence rules.

Note Repurposing works best with Recurring quota templates that have a recurrence period of 1 Month.

Monthly vs. BillCycle

Monthly and BillCycle really only differ when BillCycle is set to 29, 30, or 31. Current subscribers won’t be able to take advantage of 29, 30, 31 if you reuse a quota code. However, using the ChangeBillCycle API existing subscribers can update their BillCycle setting to 29, 30, or 31.

Any new subscribers provisioned with a repurposed quota template will start out with BillCycle functionality and a BillCycle value must be passed in with the CreateBalance API.

End Date and Last Recurring Refresh (LRR)

End Date will be set to 23:59:59.999 in the server's local time zone on the day before the BillCycle day. For example, if the BillCycle value is 15, with the server set to GMT (Zulu time), then the end date in March would be 2013-03-14T23:59:59.999Z.

The Last Recurring Refresh (LRR), which drives the Next Refresh date that appears in API responses and drives the actual quota refresh trigger, will be midnight on the BillCycle day in the previous month. For example, if the BillCycle value is 15, with the server set to GMT (Zulu time), then the LRR in the credit period before March 15th will be 2013-02-15T00:00:00.000Z, which would display a Next Refresh date in a QueryBalance response as 2013-03-15T00:00:00.000Z.

End Date Provisioning

The start date defaults to the date the provisioning call is made. The LRR defaults to the start date. The end date defaults to the start date plus one month with any necessary modifications of the day to respect the BillCycle value. The start, end, and LRR dates can be overridden if a start, end, or LRR date is passed in on the CreateBalance API request. Overriding those dates can cause Balance malfunctions if incorrectly set, so use caution!

Month End Dates Example

Recurring Quota templates are only able to use 1-28 for refresh dates. BillCycle was an enhancement for Recurring quota that allows Balance to accommodate the number of days variance of months. If the subscriber's BillCycle is set to 30, the refresh in February will be on the 28th or 29th if a leap year, and QueryBalance API responses would show the Next Refresh Date as YYYY-02-28 or YYYY-02-29. And once the refresh has occurred and it's now March the system is able to reset the Next Refresh date back to the 30th based on the BillCycle and would show as YYYY-03-30. Compare this to regular Recurring quota which would change the refresh date to the 28th or 29th permanently for the rest of the year. Even if the refresh occurred on January 30th, when February arrived, the refresh would be set to the 28th or 29th if a leap year. Unlike BillCycle, once the refresh has occurred and it's now March the system does not know how to reset the refresh back to the 30th as is occurred in January for regular Recurring quota. The Next Refresh date would show as YYYY-03-28 or YYYY-03-29.

Note All the dates in Balance such as start, expiration, refresh, etc. have a time element. What is set for the time element will affect expiration and refresh time on the given day.

Thresholds

Thresholds allow policy actions to be taken when a certain amount of quota has been used. Actions can be taken on threshold breach, unbreach, and continued breach status. Thresholds can be grouped to suppress past threshold breaches.

The threshold table in the Policy Builder sets thresholds that will be reported on when breached/unbreached and what their current amount is while breached. These messages are sent back to the policy engine from MsBM on Credit, Debit, Charge, and Provision functions so that policies can make decisions and take actions based on the threshold breach.

The basic conditions to use in policy configuration are:

- An OCSThresholdBreach exists

- An OCSThresholdUnbreach exists
- An OCSThresholdStatus exists

A typical action upon threshold breach is “Send a SMS notification”. To send an SMS notification, the Notifications feature must be installed and configured in the system.

Threshold Event Types

- OCSThresholdBreach: It occurs when a threshold is violated for the first time
- OCSThresholdUnbreach: It occurs when a credit, provision, refresh, or other action causes usage to drop back below a given threshold
- OCSThresholdStatus: It is the message that is sent every time an action is conducted against an account where a balance threshold or quota threshold is currently exceeded. This message reports the fact that the threshold is still breached and what the current level of the breach is.

Note A threshold is breached when the value is greater than or equal to the threshold value.

Balance Functions That Evaluate Thresholds

- Charge: Checks thresholds of the account balance specified in the charge request and any quotas under that account balance whose total changed due to the charge.
- Credit: Checks the thresholds of the account balance and quota specified in the credit request.
- Debit: Checks the thresholds of the account balance specified in the debit request and all quota codes under that account balance unless a quota code is specified on the debit request, in which case, it only checks the thresholds of that quota.
- Reserve: Checks all thresholds on the account.
- QuerySubscriber: Checks all thresholds on the account.

Reference Data vs. Subscriber Specific Thresholds

Reference Data Thresholds (RDT)

- Reference Data thresholds (RDT) are defined on the Balance or Quota Template in Policy Builder.
- RDTs are evaluated for all subscribers provisioned with the related balance or quota code whose template has the threshold defined.
- RDTs are stored in the reference data that the Policy Engine reads for operational configuration.

Subscriber Specific Thresholds (SST)

- Subscriber Specific Thresholds (SST) are defined via API or Policy Action.
- SSTs are only applicable for the subscriber for which the SST was defined via API or Policy Action. You must defined the SST individually for each subscriber for which you want the threshold applicable.
- SSTs contain the same types of information as RDTs, but the information is stored on the subscriber account in the database.

Unique Names

Thresholds must have unique names. SSTs and RDTs must have unique names as well. The same SST name can be used for multiple subscribers, but that value must be unique compared to the name values for the RDTs.

Important Clarifications

- Even though both kinds of thresholds share the same types of information, there is no crossover between the two sets of information. RDT definition via the Policy Builder is for RDTs only. SST definition via API is for SSTs only.
- It is important to understand that the codes and information defined in Policy Builder for RDTs have no relationship to SSTs.
- If you use an RDT code when creating an SST, the information needs to be defined for the SST and will not read the RDT information just because it's the same code.

Threshold Groups

Thresholds with the same value in the Group column will be "grouped" together. When thresholds are grouped in this manner, only messages for the first (top to bottom in the table in Policy Builder) threshold breached in the given threshold group will be returned.

For example, if you define a 80 percent, a 60 percent, and 50 percent threshold and they are in descending order, top to bottom in the table, and put them in a threshold group named CiscoPercents, the system will only send threshold messages about the highest threshold breached. This helps reduce duplicate messages. For example, a subscriber's usage is at 62%, the subscriber will only get messages about the 60 percent threshold's status. When the usage crosses 80% and goes to 81%, the subscriber will no longer get the 60 percent threshold's status message, but instead will get an 80 percent breach message and moving forward will only get 80 percent threshold status messages.

Note Order is very important! This functionality is not based on the highest value. If there are two thresholds in a group say at 60 percent and 80 percent and they are ordered in the table top to bottom in ascending order, that is, 60 nearest the top, the subscriber will never get 80 percent threshold notifications unless you select the amount remaining option instead of amount used (default).

Thresholds and Reduction of Reservation Granted Amounts

A Threshold defined on an Account Balance Template reduces the reservation amount as it nears the threshold. For example if the subscriber is 50 MB away from the threshold and the default reservation amount is 100 MB, the reservation will be reduced to 50 MB so as to not exceed the threshold.

Note For all Balance versions/revisions built prior to 7 Jan 2014, reservation amount reduction as a threshold is approached only works for thresholds NOT defined as Trigger On Remaining, that is, it only works for thresholds that measure the amount used.

A Threshold defined on a Quota Template does NOT reduce the reservation amount as it nears the threshold.

When the reservation granted amount is reduced from the requested amount due to a threshold, the quota granted is reduced to the amount between the current usage level and the value where the threshold would be breached. This reduction continues on each successive reservation until the Default Minimum Dosage defined on the Balance Plugin Configuration is reached. After that value is reached for the granted amount, the next reservation will go back to normal behavior and trigger the OCSThresholdBreachOccurred condition.

Soft vs. Hard Thresholds

Currently, all thresholds in CPS are "soft" thresholds.

The difference between a soft and hard cap is that the system would still grant the minimum dosage with a soft cap; however with a hard cap the system will deny the quota request if the minimum dosage would breach the threshold. The plan is that when a hard threshold is implemented, an API call or Policy Action would have to be made to "unlock" the threshold to allow reservations to breach the threshold and for normal operation to resume.

Other Threshold Information

- Thresholds are based on CHARGED amounts. Reserved amounts are not included.
- Thresholds can be defined on the Account Balance Template (monitors all child quotas as an aggregate) and the Quota Template (only monitors the credits of that quota).
- Thresholds are based on the total of all currently valid credits under the specified balance/quota. A "currently valid credit" is a credit for which its start date is before the current date and its end date is after the current date. For example:
  1. There is a credit of 1 GB that ends on Oct 15th.
  2. There is a percentage threshold at 90%.
  3. The subscriber uses 900 MB of data, which triggers the threshold.
  4. Another 1 GB credit is applied that ends on Oct 31st.
  5. The calculated percentage against the threshold is now 55%. However, if the subscriber waits to use the network until after Oct 15th, then the calculated value will be 0%.
- Percentage based balance thresholds are based on the ((amount charged divided by the original amount) * 100) across all currently valid credits of all quotas defined under the given balance. For quota thresholds only the currently valid credits of that specific quota are considered
- Using the QuerySubscriber API:
  - The original amount that a threshold is compared against can be determined using the calculation of balanceTotal + debitedTotal + reservedTotal.
  - The amount charged is the debitedTotal.
  - Therefore a percentage threshold is calculated as (debitedAmount/(balanceTotal + debitedTotal + reservedTotal)) * 100.

Depletion and Exhaustion

The Depleted flag is set when isExhausted is set to true and the granted quota is zero on the OCSCreateReservationResponse from the Balance Manager.

IsExhausted is set whenever the full requested reservation amount cannot be fulfilled.

Note Keep in mind that in both cases these conditions may not mean that the balance is completely exhausted permanently. If there is more than one reservation against the balance, one of those reservations may only be partially charged or released altogether (either through expiration or zero charge) which will release an amount of quota which will again become available for use.

Depletion and Exhaustion vs. Thresholds

Depletion and Exhaustion, as discussed above, are based on BOTH Charged and Reserved amounts.

Thresholds are ONLY based on Charged amounts.

This differentiation is particularly important when using 100% or total amount used thresholds. Just because the Depleted flag is set to True DOES NOT mean a 100% Threshold will have been breached yet. The outstanding reservations that may exist when Depleted is triggered need to be FULLY charged before the Threshold Breach will occur.

Charging Expired Reservations

The Balance Plug-in Configuration contains a field called Expired Reservations Purge Time, which when set allows the retention and charging of expired reservations. In some systems with significant lags in usage reporting, this option provides a mechanism to maintain more accurate accounting.

The Expired Reservations Purge Time is how long reservations can be charged after they expire as long as quota is not exhausted.

- A 0 value for Expired Reservations Purge Time means it doesn't keep any expired reservations after they expire.
- A non-zero amount is the amount of time in minutes it will keep a reservation and allow charges against said expired reservation.
- There is not a recommended value other than zero which is the default for legacy reasons. The value depends on how the system is being used, what network device is being used, and how often and how late it reports usage.

Credit Selection Logic for Reservations and Debits

Determining the next active credit to reserve against is done by the following logic:

- Credits belonging to the highest priority (lowest numerical priority value; priority 1 is highest) quotas are examined first.
- Credits that are next to expire are examined next. That is, credits with the soonest end date. If there are multiple credits with the same soonest end date, then the credit will be selected from that subset as the one with the oldest start date.
- If no credits with end dates are found, credits with no end date are examined, and the credit with the oldest start date is used first.

Note This logic is restricted to a given quota code when a Debit is performed with a quota code specified as opposed to a Debit without a quota code which will check across all the quotas defined for the subscriber to find an applicable credit.

Rates and Tariff Times

Rates provide a mechanism to alter how quota is billed to a customer. Typically, it's a 1 to 1 relationship. A customer uses 1 MB and is charged an amount, say $1, for that 1 MB. By changing the rate, the SP changes the cost that the subscriber pays. For example, a rate of 0.25 will charge quota at a cost of 1 MB for every 4 MB used. A rate of 2 will charge quota at a cost of 2 MB for every 1 MB used.

The rate is specified when the system makes a reservation. The default rate is 1.

Tariff Times

Tariff Times is the CPS nomenclature for defining rates and when to apply them. To determine the current TariffSwitchTime, Balance takes the current time (using the time zone specified on the tariff time reference data configuration) and checks each switch time in order top to bottom to see if the current time matches a TariffSwitchTime. The first tariff switch time that matches (including the associated valid dates OR a holiday date) will be used. The time of the tariff switch will be the end of the current tariff period. Then the next tariff switch time is calculated, by taking the end of the current tariff switch time, adding one second and searching each tariff switch time (top to bottom) to find the first one that matches.

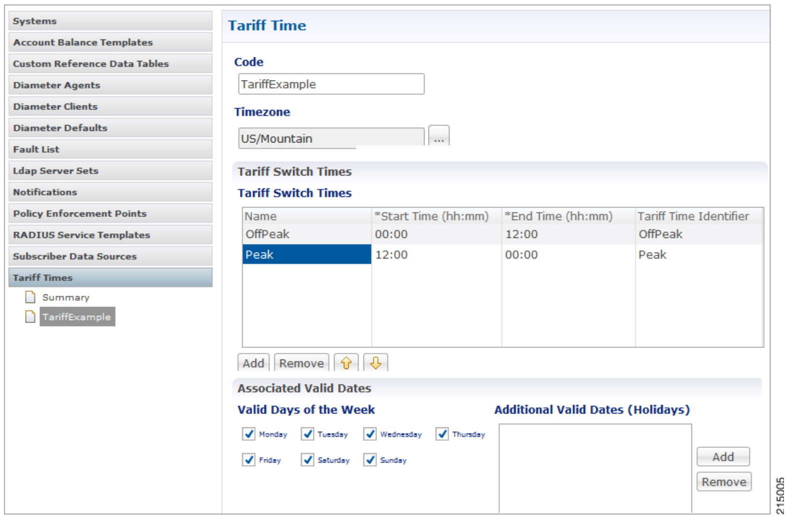
Tariff Times Configuration

This covers setting up rates for any component which uses Autowire Balance. Autowire Balance is the default blueprint for Balance that must be configured in the Policy Builder for use in the base system setup. Autowire Balance is an extension of the main system blueprint which Cisco engineering refers to as Autowire.

1. In Policy Builder, select **Reference Data** > **Tariff Times**.
2. Create a new child.
3. Set the timezone if needed.
4. Make sure that you make rows which cover all 24 hours in a day.

   Note A Start Time of 00:00 (midnight) is inclusive. An End Time of 00:00 (midnight) is exclusive because 00:00 technically is the start of any given day. By using 00:00 for the end time instead of 11:59, it allows the system to account for all 86,400 seconds (24 hours) in a day.

Figure 4: Tariff Time



Tariff Time

Code: TariffExample

Timezone: US/Mountain

Tariff Switch Times

Name	*Start Time (hh:mm)	*End Time (hh:mm)	Tariff Time Identifier
OffPeak	00:00	12:00	OffPeak
Peak	12:00	00:00	Peak

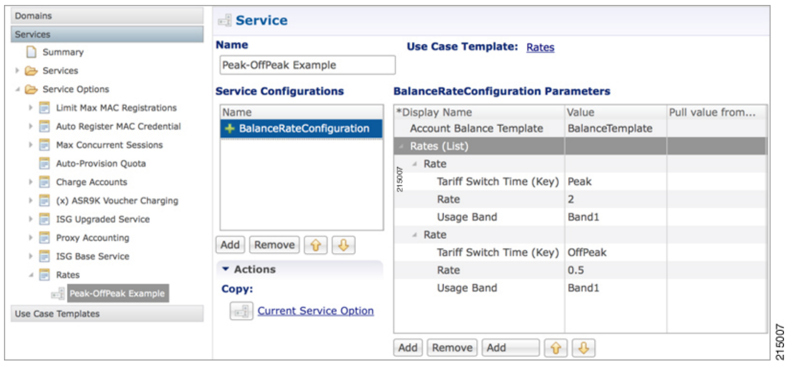
Associated Valid Dates

Valid Days of the Week: ✓ Monday ✓ Tuesday ✓ Wednesday ✓ Thursday ✓ Friday ✓ Saturday ✓ Sunday

Additional Valid Dates (Holidays)

5. In Policy Builder, select **Service** > **Use Case Templates**.
6. Click **Actions** tab.
7. Select **Add** in **Service Configuration**.
8. Select **BalanceRateConfiguration**.
9. Choose an Account Balance Template.
10. Under the Rates List, choose a Tariff Switch Time (Key).
11. Under the Rates List, change the Rate as needed.
12. Click **Add Child** to add more Rate options.
13. In Policy Builder, select **Services** > **Service Options** > **Rates**.
14. Click **Create Child Service Option**.
15. Click **Add** in **Service Configuration**. Select **BalanceRateConfiguration**.

Figure 5: Service



Service

Name: Peak-OffPeak Example

Use Case Template: Rates

Service Configurations

Name
BalanceRateConfiguration

BalanceRateConfiguration Parameters

*Display Name	Value	Pull value from...
Account Balance Template	BalanceTemplate	
Rates (List)		
Rate		
Tariff Switch Time (Key)	Peak	
Rate	2	
Usage Band	Band1	
Rate		
Tariff Switch Time (Key)	OffPeak	
Rate	0.5	
Usage Band	Band1	

16. Click **File** > **Publish to Runtime**.

Edge Cases

It is strongly recommended that Tariff Switch Times cover all times during a 24 hour period and do not have gaps. Some customers use a default Tariff Switch Time entry that covers all the other times that have not been specifically defined.

Tariff Times are not allowed to cross over the midnight boundary for a given day. In practice, this means that often 2 or more tariff switch times must be created to cover a single logical period. For Example:

Table 5: Edge Case Examples

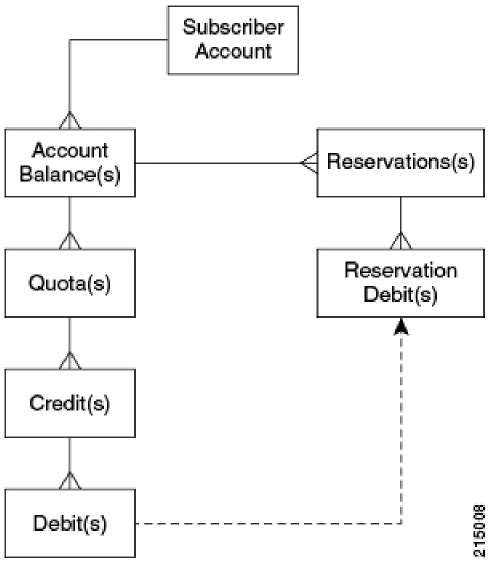
Name	Start Time	End Time	Tariff Time Identifier
Nights Before Midnight	17:00	00:00	NIGHT
Nights After Midnight	00:00	07:00	NIGHT
Days	07:00	17:00	DAY

If a Tariff Switch occurs during a daylight savings time forward switch (i.e. between 2:00am and 3:00am during 'Spring Forward' in March), an error will occur in processing during that time since that hour is lost. Therefore, it is recommended that switch times NOT occur during these times on those days.

Subscriber Record

For any given subscriber, the following illustrates the database relationship of the objects described in this chapter.

Figure 6: Subscriber Records



An important point to keep in mind when looking at the data structure for MsBM is that this is a document based database and not a relational database. The below description attempts to describe the equivalent RDMS concepts as far as primary and foreign keys are concerned; however, these concepts are not the same in a document based database, and are included merely to assist in providing understanding since most of audiences are familiar with RDMS.

For example, since Mongo stores data in a structure that is essentially a map of maps, there is no need for the concepts of a primary key (PK) or a foreign key (FK) since relationships between the data is defined by the physical structure. Conceptual PKs and FKs are only needed when interaction with the QNS system or when the querying is necessary and the full document is not retrieved.

Note This listing is not a complete listing of all fields (conceptually columns) in each map (conceptually table), but only the important fields used for data structuring or querying.

Subscriber Account

- subscriberId (PK)
  - Key field to reference a given subscriber account. This account may be shared among multiple subscribers.
  - Value is also stored as _id which is a key field for the Mongo database document. These values were different in early versions of MsBM, but have been aligned to ensure greater system stability. The _id was originally a Mongo assigned UUID.
- The Subscriber Account contains all data structures related to a specific subscriber's account as it relates to MsBM. When you retrieve a Subscriber Account, you obtain all MsBM information for a given subscriber.

Account Balance

- accountBalanceCode (PK within the scope of a given subscriber account)
  - relates to the Account Balance Template reference data information defined in Policy Builder.
  - key for Policy Engine to access this balance's information for a selected subscriber account.
- An account balance, or simply a balance as it is often referred to, is essentially a group of quotas. The idea is that a customer could create a balance (quota grouping) called Data and within that have several quotas defined, like Monthly, Topup, and Bonus. Then when the subscriber uses account, usage is charged against their Data balance and MsBM determines which underlying quota should be debited based on rules set up in Policy Server (QNS). The intent here is to simplify charging so that specific “buckets” do not need to be specified and necessary business rules can be defined once in the Policy Engine.

Quota

- quotaCode (PK within the scope of a given subscriber account)
  - relates to the Quota Template reference data information defined in Policy Builder.
  - key for Policy Engine to access this quota's information for a selected subscriber account.

- A quota defines the actual amount or "bucket" that can be drawn from for tracking of usage. This value internally is unit-less and thus can be defined to be whatever is needed in the implementation. Some units are provided in reference data in Policy Builder to assist users. Common usages for quota amounts are bytes of volume, seconds of time, but other counts can also be tracked.

Credit

- A UUID (PK) is used as a key to the map containing all the credits for a given quota.
- A UUID was selected since the data structure was defined for transactional speed and stability and was not intended to be queried directly. MsBM records are intended to be accessed using the Policy Engine, the provided APIs, and the Reporting Engine. The intent in the design of this structure was transactional usage and performance, not queryability. A sequence concept or other more predictable key was not used since a sequence would most likely be rapidly exhausted and more difficult to manage since credits are regularly created and destroyed as part of the normal operation of MsBM. Uniqueness and assurance of avoidance of deadlocking is the primary concern.
- Credits can be viewed as positive entries on an accounting balance sheet.

Debit

- A UUID (PK) is used as a key to the map containing all the debits for a given credit.
- The reasoning for selection of the UUID is the same as discussed under Credit, on page 23.
- Debits can be viewed as negative entries on an accounting balance sheet for a specific credit. Debits under a given parent credit cannot exceed the amount of the parent credit.

Reservations

- reservationId (PK)
    - Also a UUID and was selected for similar reasons as discussed under Credit, on page 23.
- Reservations describe an amount of conditional debiting of quota that is reserved until a network device reports that the quota has actually been utilized.

Reservation Debits

- reservationDebitId(PK)
    - Also a UUID and was selected for similar reasons as discussed under Credit, on page 23.
- reservationId (FK)
- debitId (FK)
- Reservation debits serve the purpose of providing a link between a reservation and an actual debit entry. When a reservation is charged, the reservation and associated reservation debits are removed and the

debit linked to those reservation debits are made permanent. That is, as long as a debit has a linked reservation debit, that debit is not permanent, but only reserved in a non-committed state.

Thresholds

The Thresholds collection contains any reference data thresholds (defined in Policy Builder) that are currently breached or any user-defined thresholds that are specific to a given subscriber account.

The thresholds map is not utilized and may not appear in all implementations. It will only appear as needed if the related functionality in Policy Server (QNS) is active.

Expired Reservations

An identical structure as Reservations is used for Expired Reservations, except that Reservation Debits do not have a link to Debits. Functionality for charging expired reservations, that is, creating debits, is handled in a different fashion.

The expired reservations map is not utilized and may not appear in all implementations. It will only appear as needed if the related functionality in Policy Server (QNS) is active.

Shared Quota

In CPS there are several ways to set up shared quota. SPR supports parent and child profiles, for example one parent in a household is the primary SPR record and all the other members of the household are set as child records of that SPR profile. This would allow for shared quota and is mostly configured through SPR data management. Because Balance and SPR can be used separately, Balance also supports shared quota use cases that are configured solely within the Balance module.

Shared Per User Limit Use Case

There will be two Balance accounts associated with a subscriber that is participating in a shared quota but also needs a per user limit on said shared quota. The first Balance account is the subscriber's personal account. This account will contain any balances/quotas that are only available to the subscriber. This account will also contain one balance that will be used for tracking the per user limit. The second Balance account is not owned by the subscriber and contains the shared balance/quotas.

Note The two Balance accounts need to be provisioned separately.

The shared balance template contains a field called Limit Balance that links the shared balance to a limit balance (the personal account), so that the Balance module knows which two balance codes it needs to reserve/charge against in the shared per user limit case. Since the per use limit is tied to a quota template, only discrete per user limits are supported. The number of balance/quota templates defined is not limited, but the templates must be defined for each per use limit level. The limit quotas must be defined in a balance/quota template.

The subscriber must be provisioned with the limit balance and an associated quota with a valid amount for the per user limit to be enforced. An AVP must added to the subscriber profile in SPR indicating the Balance account record and the Balance template name of the shared quota. This is done currently for some deployments. The subscriber must have the AVP set up prior to per user limits being available.

If the subscriber does not have the limit balance provisioned, then the subscriber will draw from the shared balance with no per user limit enforced.

Hard thresholds (meaning the subscriber cannot use more than a certain amount of the shared quota) will be enforced by the amount provisioned in the limit balance.

Soft thresholds (meaning the subscriber can continue to use more up to the hard threshold, but something should happen when the soft threshold is crossed) will be supported using the threshold mechanism defined on the limit balance.

The charge and reserve function of the Balance module were enhanced with conditional logic to make new calls to handle the shared per user limit reservations. The new calls allow charges or reservations against two Balance accounts at the same time. The two Balance accounts are the shared account, as indicated by the AVP, and the subscriber's personal balance account (the limit balance).

Policy Engine

The Balance module exposes various policy objects that can be used to monitor the status of an account. Aptly named the MsBM Account Status object, it contains information about a specific balance of a given subscriber. Each of the subscriber's balances will have its own MsBM Account Status object in the Policy Engine during policy execution.

The **Amount Remaining** value on the MsBM Account Status object DOES contain the values of any current reservations.

Proration

Balance provides some limited proration capabilities but in general, proration must be handled manually via API calls (Credit, Debit, ExtendCredit, CreateBalance).

Proration Example

A subscriber has 5 GB on the first plan and has used 3 GB of it. The subscriber then switches to a different plan with 2 GB. The subscriber will start with 2 GB of available quota UNLESS the CreateBalance API overrides the initial amount. Setting the override amount is a manual step that is handled by the calling system, i.e. customer portal or OSS/BSS application.

Quota Refresh Throttling

Balance has the ability to cause a batch of quota refreshes to be staggered over a time period, which causes session wakes up to be staggered, which not only keeps masses of subscriber accounts from being refreshed at the same exact time, but also causes any other events related to a session wake up to be staggered, i.e. RARs. This concept is called the Callback Validity Time (CBVT). The CBVT is usually set to the time where something changes in a subscriber’s balances/quotas. Typically this is the expiration date of their quota. The CBVT is that time at which a session will “wake up” and create a new reservation of quota. This "wake up" activity triggers a quota refresh if one is valid for a recurring quota.

For example, let's say that 50,000 subscribers on a monthly quota have their quota set to expire at 1 AM and all have sessions established. Normally their quota wouldn’t refresh until they next accessed their account, i.e. had an active session that made a reservation or other Balance request (the refresh is retroactive however). However, some deployments have subscribers who always have a session, but it may not be actively using quota, i.e. idle cable modem. In this scenario, at 1 AM, 50,000 subscriber sessions will “wake up” and refresh their accounts which could easily cause a serious load spike for system resources.

To combat this problem, the Recurring Refresh Max Delay parameter defined in the Balance Plug-in Configuration, is used to pad the CBVT value by a random number of minutes between 0 and the parameter value. If the Recurring Refresh Max Delay param is set to 120, then the CBVT value on the session will be set to 1 AM plus a random number of minutes chosen from between 0 and 120. Now, the 50,000 sessions will not all wake up at 1 AM. Because the CBVT values are set to the range from 1 AM to 3 AM, at any given minute only a small percentage of the total 50,000 sessions will wake up and refresh.

Active Session vs. Inactive Session

Any subscriber actively using their quota will refresh immediately at 1 AM when they qualify for the refresh and will not have their quota refresh delayed. Only subscribers with sessions that are not actively using quota will have their refreshes delayed.

Sy Server Implementation in OCS

This section describes how to configure an OCS Sy server to manage policy counters that map to a subscriber’s account balance template in an OCS node deployment.

Sy Client and Diameter Stack Configuration

This section describes how to configure the Diameter Sy client and the associated Diameter stack.

Step 1 In Policy Builder’s **Reference Data** tab, select **Diameter Clients** in the left pane.

Step 2 Expand **Sy Clients**, and click **Sy Client** under **Create Child** in the **Sy Clients Summary** pane.

Step 3 Configure the client as needed.

Step 4 Set the **Counter Lookahead Interval Minutes** option to the number of minutes to look ahead to determine when the lookahead balance states configured for the SyServerSLRInformation service configuration object will expire, refresh, or start. It is set to 180 minutes by default.

Step 5 In the left pane, select **Systems** > **system_name** > **Plugin Configuration** > **Diameter Configuration**.

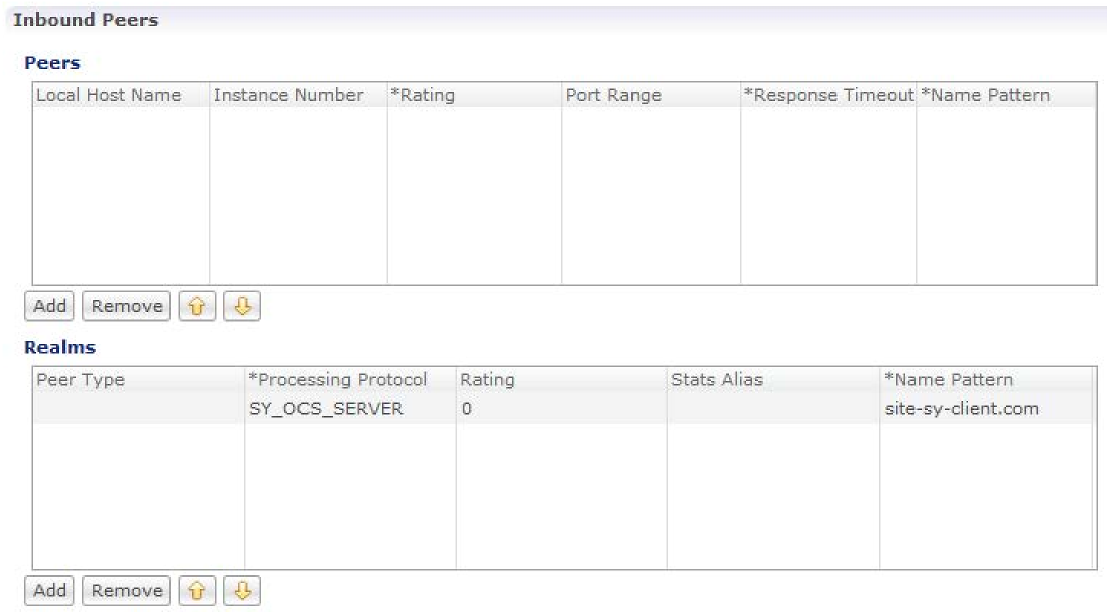
Step 6 In the **Diameter Configuration** pane, click **Create Child**, click **Diameter Stack**.

Step 7 In the **Diameter Stack** pane, type a **Name** and a **Realm**.

Step 8 Under **Inbound Peers** in the **Realms** table, select **SY_OCS_SERVER** in the **Processing Protocol** pull-down menu.

Step 9 Provide a **Rating** and a **Name Pattern** as well.

Figure 7: SY_OCS_SERVER Processing Protocol



Inbound Peers

Peers

Local Host Name	Instance Number	*Rating	Port Range	*Response Timeout	*Name Pattern

Realms

Peer Type	*Processing Protocol	Rating	Stats Alias	*Name Pattern
	SY_OCS_SERVER	0		site-sy-client.com

Account Balance Configuration

The Sy server uses the CPS balance manager for volume/quota definition, and association to subscriber and threshold configuration. For configuration information, refer to Quota Templates, on page 3.

Use Case Template and Service Configuration

This section describes how to configure a use case template that uses the SyServerSLRInformation service configuration object. You should create a use case template using this object based on the account balance template and configured thresholds.

Step 1 Select the Policy Builder Services tab.

Step 2 In the **Summary** pane, click **Use Case Template** under **Create Child**.

Step 3 In the **Use Case Template** pane, select the **Actions** tab.

Step 4 Type a **Name**; for example, xyz_green.

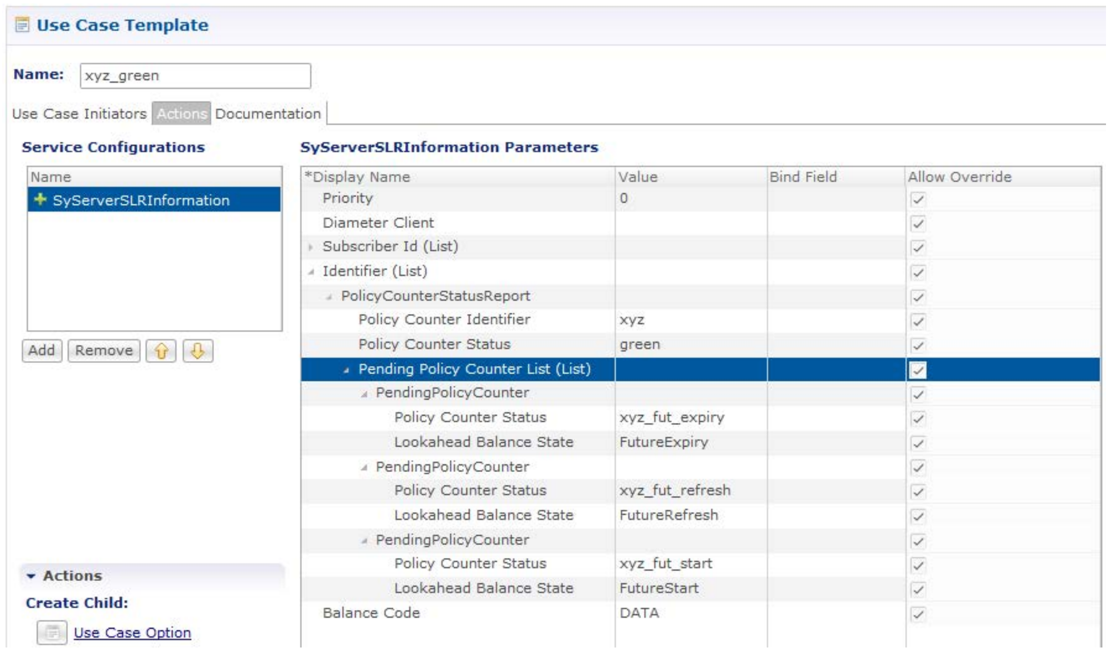
Step 5 Under **Service Configurations**, click **Add**.

Step 6 In the **Select Service Configuration** dialog box, scroll down to the **sy Server** section, select **SyServerSLRInformation**, and click **OK**.

Step 7 Configure the SyServerSLRInformation parameters.

The following figure shows an example configuration.

Figure 8: SySServerSLRInformation Service Configuration Object



Use Case Template

Name: xyz_green

Use Case Initiators | Actions | Documentation

Service Configurations

Name
SyServerSLRInformation

Add | Remove

Actions
Create Child:
Use Case Option

SyServerSLRInformation Parameters

*Display Name	Value	Bind Field	Allow Override
Priority	0		✓
Diameter Client			✓
Subscriber Id (List)			✓
Identifier (List)			✓
PolicyCounterStatusReport			✓
Policy Counter Identifier	xyz		✓
Policy Counter Status	green		✓
Pending Policy Counter List (List)			✓
PendingPolicyCounter			✓
Policy Counter Status	xyz_fut_expiry		✓
Lookahead Balance State	FutureExpiry		✓
PendingPolicyCounter			✓
Policy Counter Status	xyz_fut_refresh		✓
Lookahead Balance State	FutureRefresh		✓
PendingPolicyCounter			✓
Policy Counter Status	xyz_fut_start		✓
Lookahead Balance State	FutureStart		✓
Balance Code	DATA		✓

The SyServerSLRInformation parameters are described in the following table:

Table 6: SyServerSLRInformation Parameters

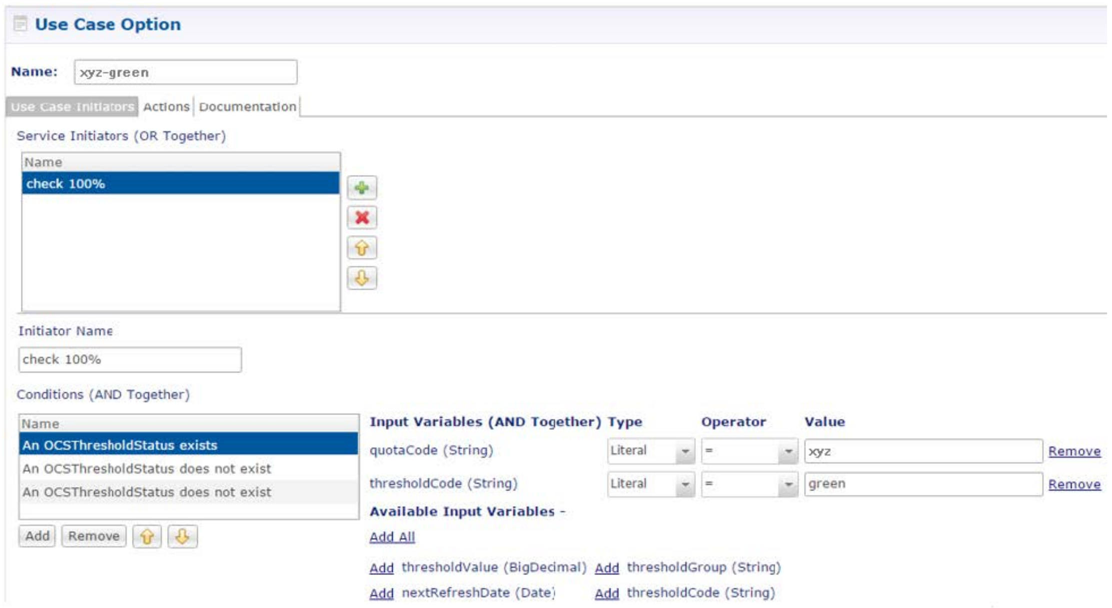
Parameter	Description
Priority	The priority of the message for processing. The higher the number, the higher the priority. Default for most settings: 0
Diameter Client	The client configuration is used to apply different policies based on PCRF type. To filter a service based on the Diameter client, specify which Diameter client you want the service to be applied to. This parameter is optional.
Subscriber ID (List)	Identifier – The user identity, which is mapped to the Subscription-Id AVP. Based on your requirements, you can configure one or more identifiers. Possible values for identifiers include: Session MSISDN – The MSISDN value of the subscriber. Session IMSI – The IMSI value of the subscriber.
Identifier (List)	Identifier Name – The subscribed Sy Policy Counter Identifier list, which maps to the Policy-Counter-Identifier AVP. OCS uses this list to send the Policy-Counter-Identifiers to the PCRF in the SLA/SSNR message.
PolicyCounterStatusReport	Contains the Policy Counter Identifier AVP, the status of this AVP, and the Pending Policy Counter List.

Parameter	Description
Pending Policy Counter List (List)	Contains the list of Pending Policy Counters. You can configure a maximum of three counters. • Policy Counter Status – You can name this anything. • Lookahead Balance State –You can select one of the following: **FutureExpiry**, **FutureRefresh** and **FutureStart**. **Note** This state is the future balance event—quota expiration, quota refresh, or quota starting in the future. Based on the look-ahead time interval, if a balance state event is going to occur within the look-ahead interval period, then the pending policy counter status is populated in the SLA/SSNR messages.

Step 8 Select the **User Case Initiator** tab, and configure a use case initiator for this use case template.

An example is shown below:

Figure 9: Use Case Initiator



Use Case Option

Name: xyz-green

Use Case Initiators | Actions | Documentation

Service Initiators (OR Together)

Name
check 100%

Initiator Name: check 100%

Conditions (AND Together)

Name
An OCSThresholdStatus exists
An OCSThresholdStatus does not exist
An OCSThresholdStatus does not exist

Add | Remove

Input Variables (AND Together)	Type	Operator	Value	
quotaCode (String)	Literal	=	xyz	Remove
thresholdCode (String)	Literal	=	green	Remove

Available Input Variables -

Add All

Add thresholdValue (BigDecimal) Add thresholdGroup (String)

Add nextRefreshDate (Date) Add thresholdCode (String)

Step 9 Configure a Service Option using the use case template.

Step 10 Configure a Service using the Service Option.

Loading Sy Session When Gy Session is From Different Realm

Sy and Gy calls originate from the same realm where SY OCS Server considers Gy as a secondary session. In such a call model, Gy CCR-I is ignored if the Gy CCR-I is received before Sy SLR the Gy CCR-I.

Sy and Gy calls can originate from different realms and Gy CCR-I needs to be handled even when Sy session is not present. In this scenario, both Gy and Sy sessions are independent of each other. Gy CCR-I can be received before Sy SLR and this needs to be handled.

You need to apply policy configuration to load an Sy session to a Gy session. The MSISDNkey and USuMSubscriberIdkey are used to correlate the Gy and Sy sessions.

Perform the following steps to load an Sy session to a Gy session:

Step 1 Log in to Policy Builder.

Step 2 Navigate to **Policies**.

Step 3 Load keys to Gy session and add the following two conditions to the policy configuration:

- A Diameter Gy v8 Session exists
- There exists an USuMSubscriber

Step 4 Under **Actions**, add the following keys:

- Add a MSISDNkey
- Add an USuMSubscriberIdKey

Step 5 To load Sy session, add the following conditions:

- A Diameter Gy v8 Session exists
- There exists a MsisdnKey
- There exists an USuMSubscriberIdKey
- An OCSThresholdBreach exists

Step 6 Under **Actions**, add the following key:

Retrieve a session from the cache