



CloudPortal Business Manager 2.1

2015-05-17 04:45:00 UTC

© 2015 Citrix Systems, Inc. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)

Contents

- CloudPortal Business Manager 2.1**..... 12
 - Overview of CloudPortal Business Manager 13
 - Release Notes..... 14
 - Main Release..... 15
 - Defects Fixed 18
 - Known Issues and Limitations 20
 - Maintenance Release..... 22
 - 2.1.2..... 23
 - 2.1.1..... 27
 - Upgrading CloudPortal Business Manager..... 31
 - Installation Guide 33
 - System Requirements 34
 - Installing and Configuring CloudPortal Business Manager 35
 - Verifying - Starting and Stopping CloudPortal Business Manager..... 47
 - Administration Guide 48
 - Configuring Portal Settings..... 49
 - Configuring Billing Settings for the Portal..... 51
 - Configuring Marketing Settings for the Portal 53
 - Configuring Appearance Settings for the Portal..... 54
 - Configuring Server Settings for the Portal..... 55
 - Configuring Internal Email Addresses..... 56
 - Configuring CRM Email Addresses 57
 - Integrating the Directory Server 58
 - Enabling Anonymous Browsing..... 65
 - Configuring Trial Management Settings 66
 - Configuring Google Analytics settings 67
 - Configuring geo restriction 68
 - Understanding B/OSS subsystems 69
 - Configuring Fraud Control Settings..... 70

Configuring Salesforce.com Service Cloud Integration	72
Configuring Payment Gateway	73
Configuring TeleSign Phone Verification.....	74
Phone Type Codes	75
Adding a new service to CAS.....	79
Customizing the Service Provider Logo and Favicon	80
Creating Service Provider Users	81
Managing Content Templates	82
Email Templates and Notification Triggers	83
Sending test mails	86
Modifying the Header, Footer, and Signature in content templates	87
Modifying Invoice Templates	88
Objects and Properties in Invoice PDF template.....	89
Modifying Content Template Styles	92
Modifying the Approval Templates	93
Profile Management	94
What are roles	95
Understanding CloudPortal Business Manager Profiles	97
Managing Profiles.....	98
Cloud Service	99
Onboarding and Managing a service	100
Workflow Management.....	103
How workflows work.....	105
Viewing workflow status	109
Available activities and their properties	111
Managing your tasks.....	113
Account Management	114
Account Types	115
Account Lifecycle	116
Account Properties.....	117
Configuring accounts	120
Editing default account properties.....	121
Configuring Onboarding Settings	123
Configuring access based on geographic location	124
Configuring access based on email domains	125
Configuring account lockout settings.....	126
Configuring payment failure settings.....	127

Configuring account resource limits	128
Enabling initial deposit for an account type	129
Managing accounts	130
Adding accounts	131
Creating self-registered accounts	132
Editing accounts	133
Removing accounts	134
Changing account states	135
Payments and refunds	136
Issuing credit to customers	138
Managing users' information of a customer account	139
Changing the primary email address of a user	140
Managing services for a customer account	141
Monitoring accounts	142
Viewing customer account dashboard	143
Viewing customer account information	144
Viewing account users	145
Viewing billing history of an account	146
Operating your cloud	147
Understanding the dashboard	148
Launching the console	149
Service Health	150
Notifications	152
Configuring notification settings	153
Managing Notification Preferences	154
Tickets	155
Filing a Ticket	156
Closing a Ticket	157
Jobs and Job Management	158
Types of jobs	159
Viewing batch jobs	164
Reports	165
Configuring report settings	166
Viewing reports	167
Product Management	168
Products	169
Understanding products	170

Adding a product.....	173
Replacing an existing product with a new product	175
Editing a product	176
Retiring a product	177
Editing product pricing.....	178
Product Catalogs	179
Understanding current, next, and historical revisions of the catalog	180
Product Bundles	181
Adding a product bundle.....	184
Editing a product bundle	186
Publishing and unpublishing product bundles	187
Editing product bundle pricing	188
Entitlement Behavior and Calculation.....	189
Scheduling activation.....	190
Channels	191
Adding a channel.....	193
Editing a channel	194
Deleting a channel.....	195
Adding currencies to a channel.....	196
Synchronizing with reference catalog	197
Adding product bundles to a channel catalog.....	198
Editing product and product bundle prices in channel catalog	199
Scheduling activation	200
Previewing the channel catalog	201
Campaigns and Promotions	202
Adding a campaign	203
Understanding Billing in CloudPortal Business Manager	204
Billing Periods.....	205
Account Statement.....	206
Invoices	207
Accounting for Entitlements.....	209
Subscriptions	210
Instant Payment	211
BSS API Guide	212
Intended Audience.....	213
Overview of BSS API	214
REST Architecture	215

BSS API data model	216
Supported Operations	217
Nested Resources.....	218
Simulating PUT and DELETE.....	219
Custom Actions.....	220
Requesting data formats	221
Standard query parameters	222
Root API Path and Versioning	226
Sample Responses	227
Localization.....	233
Roles	234
Using the API	235
Generating the signature	236
Resources	237
Accounts	238
Users	249
Subscriptions.....	256
Accountstatements	264
Invoices	268
Payments	272
Cloud Services	275
Catalogs.....	277
Events.....	280
Service Health Status.....	286
Service Maintenance Activities.....	289
Channels	291
Currencies	293
Usage.....	294
Profiles.....	298
AccountTypes	300
Countries	301
Known issues	303
Customization Guide.....	304
Customization Overview	305
Installing and building custom components	306
Customizing UI	311
Customizing Channel-specific CSS.....	313

Customizing Validation Rules	314
Customizing Fields and Validations	321
Customizing Resource Properties	325
Customizing Workflows	326
Adding an existing activity to a workflow	334
Creating approval workflow for subscription creation with specific bundles	336
Customizing Task Templates	337
Customizing Topic Subscriber	338
Customizing Payment Gateway and Ticketing Service	340
Customizing Reports	344
Adding a new Custom Report Class to CloudPortal Business Manager	346
Scheduling custom reports	350
Generating custom reports from the UI	351
Customizing Jobs	352
Customizing Country List	354
Customizing Models, DAOs, and Services	355
Customizing Email Templates	367
Objects and Properties for Email Templates	369
Customizing Connector UI	384
Customizing Billing Exports	386
Customizing Pricing Filters	395
Customizing Profiles	397
Customizing Account Types	398
Customizing Discount Adjusters	400
Customizing Tax Adjusters	403
Migration Guide	406
Upgrade Mechanism	407
List of Pre-checks	408
Upgrade steps	409
Steps to Upgrade LDAP push from CloudPortal Business Manager 1.4.x to CloudPortal Business Manager 2.x	411
Impact on customization	412
Migration to Cloud Service, OSS and associated Service Instances	414
List of behavioral changes	416
Known issues	417
Connector Development Kit	419
Terminology	420
Target Audience	421

Overview	422
Multiple Instances	424
Onboarding	425
Metadata.....	426
SPI	427
Resource Bundles.....	428
Service Installation and Bootstrap	429
Installing a Cloud Service Connector	430
Service Bootstrap.....	431
Service Metadata	432
Configuration Metadata.....	433
CloudConnectorFactory Interface	434
ServiceInstanceConfiguration	435
CloudConnector Interface	436
Security Roles.....	438
Product Management.....	439
CPBM Product Management Primer	440
Products	441
Product Bundles.....	442
Product Metadata and Metadata Registry	444
Resource Type.....	445
UsageType	446
Unit of Measure (Uom)	447
Filter	448
MetadataRegistry Interface	449
Understanding How This Works	453
Lifecycle Management.....	461
Account Lifecycle	462
User Lifecycle	466
Subscription Lifecycle.....	470
Usage Collection	473
Event Collection	476
UI Contributions.....	479
Single Sign-on Setup.....	481
Using CAS (Central Authentication Service)	482
Implementing “SsoHandler” Interface	483
Development Using SDK.....	485

Creating the CS-SDK Maven STS project	486
Connector Fragment	489
Contributing Custom UI for Configuring a Subscription for Provisioning.....	492
Citrix® CloudPlatform™ Connector Guide	498
Installing Citrix CloudPlatform Connector	499
Sending Citrix CloudPlatform APIs	500
Onboarding and Managing the Citrix CloudPlatform service	501
Preparing Citrix CloudPlatform to work with CloudPortal Business Manager	502
Integrating CloudPortal Business Manager with Citrix CloudPlatform for	504
Single Sign-on	
Enabling the Citrix CloudPlatform Connector.....	505
Adding an instance to the CCP service.....	506
Region support configuration.....	508
Service Controls	510
Editing a CCP instance.....	511
CCP Connector user roles.....	512
Mapping of CPBM and CCP entities.....	513
Product Onboarding	516
CCP Usage Types	517
CCP Resource Types	519
Sample CCP Products	520
Sample CCP Product Bundles	522
CCP Resource Provisioning	524
Async Job	525
Managing resources.....	526
Instances.....	527
Accessing the VM Console	529
Resource properties for provisioning VM	530
Volumes.....	532
Snapshots	536
IP Addresses.....	538
Networks.....	540
Templates	542
ISOs	544
Setting service instance specific user settings.....	546
FAQs	547
User Guide	548
User Management.....	549

Users and Profiles	550
Adding a User	552
Managing information of your account users.....	553
Unlocking accounts	554
Managing company settings	555
Managing user profiles and preferences	557
Changing your primary email address.....	559
Viewing API Credentials	560
SSH keys.....	561
Managing your spend	562
Setting your budget.....	563
Creating alerts.....	564
Viewing spend summary.....	565
Viewing spend notifications	566
Understand Your Bill	567
Browse Catalog and Subscriptions.....	571
Anonymous Browsing.....	573
Understanding the Subscription Cycle	574
Subscription states.....	575
Subscribing to a Product Bundle	576
Provisioning a resource.....	577
Reconfiguring a resource without changing the bundle or subscription	578
Reconfiguring with a new bundle or subscription	579
Filter Subscriptions	580
Terminating a subscription	581
Provisioning a resource from the pay-as-you-go mode	582
Filing support tickets.....	583
Viewing service health	584
Understanding the dashboard.....	585
My Services Tab	587
Enabling a service	589
CloudPortal Services Manager Connector Guide.....	590
System Requirements	591
Installing the Connector	592
Enabling the CloudPlatform Services Manager Connector	593
Connector Service Configuration	594
Connector User Roles	595

Connector Usage Types	596
Connector Resource Types	597
Connector Account Configuration	598
Single Sign On	599
Windows Desktop Connector Guide	600
System Requirements	601
Installing the Connector	602
Connector Service Configuration	603
Connector User Roles	604
Connector Usage Types	605
Connector Resource Types	606
Single Sign On	607
OpenStack Connector Guide.....	608
System Requirements	609
Installing a Single Node OpenStack Server	610
Single Sign On Configuration	611
Building Connector From Source	612
Installing Connector in CloudPortal Business Manager	613
CloudPortal Business Manager OpenStack Connector Service Configuration	614
Connector Offerings (Resource Types)	615
Connector Usage Types	616
Connector Service Controls	617
Connector User Roles	618
Utilities	619
Active Directory Import Tool.....	620
Setting up the Customization Package.....	621
Setting up Active Directory Import Tool	622
Triggering the Active Directory Import Tool On Demand	624
Configuring the Active Directory CloudPortal Business Manager Account Mapping	625
Active Directory Connection Configuration	626
Active Directory Import Run Reports.....	628
Frequently Asked Questions	629

CloudPortal Business Manager 2.1

Citrix® CloudPortal™ Business Manager 2.1 is a cloud services delivery and operations platform for delivering self service IT-as-a-service to consumers. It enables enterprises and service providers to unify the commerce, user management, service provisioning and operational aspects of running a cloud interface for delivering anything as a service (XaaS).

It provides a wide range of business operations features that include availability, system health, performance, account status, utilization, billing, helpdesk, and alerts. It also leverages the partner ecosystem and newly available SDKs and APIs to seamlessly deliver any service (IaaS, PaaS, storage, and so on) while broadening the portfolio of services to differentiate and increase the value of the cloud.

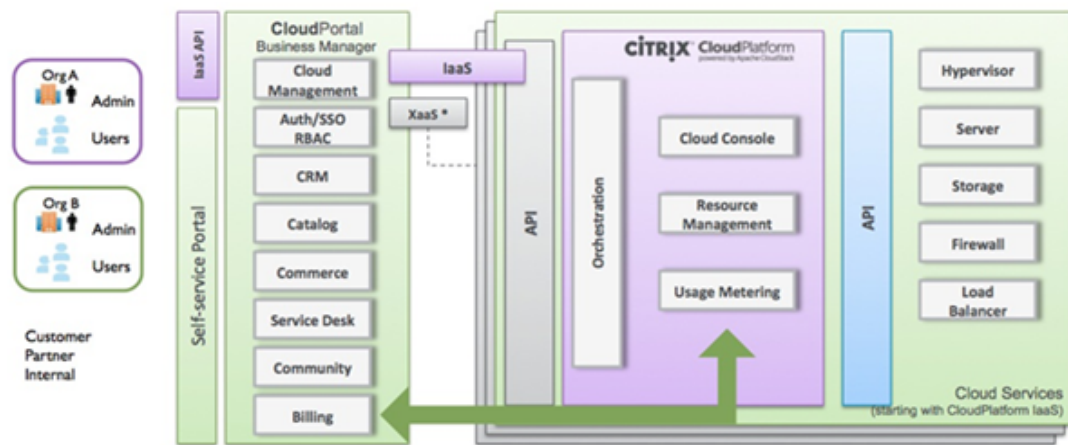
In addition, CloudPortal Business Manager has features uniquely designed for the needs of enterprises such as AD Integration and approval based workflows. CloudPortal Business Manager is also localized and can operate in 8 languages and 160+ currencies, making it readily deployable in global environments.

In This Section

Release Notes
Installation Guide
Administration Guide
BSS API Guide
Customization SDK Guide
Migration Guide
Connector Development Kit
CCP Connector Guide
User Guide

Overview of CloudPortal Business Manager

CloudPortal Business Manager is a cloud services delivery and operations platform for delivering self service IT-as-a-service to consumers. It is targeted towards both Service Providers as well as Enterprise IT organizations and provides a wide range of business operations features include built in billing and spend tracking engines, product and catalog management, self serviceability of resources, integrations with payment gateways and fraud control systems. In addition, CloudPortal Business Manager has features uniquely designed for the needs of enterprises such as AD Integration and approval based workflows. CloudPortal Business Manager is also localized and can operate in 8 languages and 160+ currencies, making it readily deployable in global environments.



Originally developed as a B/OSS suite for delivering Infrastructure as a service, CloudPortal Business Manager provides a powerful platform for packaging, delivering and metering a custom set of cloud services to users. As an open and extensible platform, cloud operators, and administrators can leverage the newly available SDKs to offer cloud services beyond IaaS, and seamlessly integrate with existing business and operations systems. CloudPortal Business Manager is going to help cloud builders in three main ways:

- Deliver Anything-as-a-Service from a customizable catalog of services capable of delivering IaaS, PaaS, SaaS, 3rd party cloud services and enterprise.
- Empower users with self-service access to the cloud and IT services, infrastructure and account management tools they need from an easy to use cloud services "storefront".
- Simplify cloud services management by unifying and automating services provisioning, B/OSS systems, billing and metering and user management under a single, extensible platform.

Release Notes

Release notes describe the enhancements, changes, bug fixes, and known issues for a particular release of CloudPortal Business Manager software. Release notes are categorized into:

- [Main Release](#)
- [Maintenance Release](#)

Main Release

What's New in CloudPortal Business Manager 2.1

New Features:

- **BSS API**

Business Support System (BSS) API allows developers to create new management solutions or integrate existing systems with CPBM. This API represents the set of business-related services exposed by the CPBM.

RESTful endpoints to access the following business objects in CloudPortal Business Manager:

- Users
- Profiles
- Account
- Account Types
- Account Statements
- Subscriptions
- Invoices
- Payments
- Usage
- Products
- Catalog
- Channel
- Cloud Services
- Currencies
- Countries
- Events
- **Customization SDK**

The Customization SDK provides a set of supported interfaces and tooling to enable Do-It-Yourself (DIY) implementation teams to extend CPBM product to meet business objectives.

The Customization SDK provides interfaces to allow the customization of:

- User Interface
- Workflows
- Reports
- Email Templates
- Pricing filters hooks
- Billing data exports
- Payment Gateway and Ticketing Service
- Fields
- Validation Rules
- Resource Properties
- Model, DAO, and Services
- Topic Subscriber
- Country List
- **Support in self-service registration for Channel-specific Account creation.**
- **Channel Features**
 - Service providers can now set up Channel-specific self-registration forms.
 - Service providers can now define channel specific UI.
- **Usability Enhancements**
 - Redesigned Catalog User Experience:
 - Removal of the “Default” Utility Bundle
 - Ability to view the bundles without being forced to choose filters
 - Re-designed “Utility Rate Card”
 - Ability for Product Managers to publish bundles immediately
 - Ability for Product Managers to define a global sort order and therefore, specify display order of products and product bundles in channel catalogs
 - Support for Categories and Filters to browse Products and Bundles for Admin users
 - Support for users to browse subscriptions based on subscription state
 - Support for anonymous users to browse channel specific catalog
 - Enhancement to Approval workflows

- Redesign of “My Services” Tab
- One-time product bundle is renamed to monthly bundle with no recurring changes
- Enhancements to the on-boarding experience of a new service target instance through a CloudPortal Business Manager connector:
 - Intelligent defaults for connector properties
 - Reducing the number of required properties
- Improvements to Product Setup:
 - Pre-defined UOM and Conversion factors
 - One-click setup of Products during the on-boarding of a service target instance
- **Other Features**
 - E-mail notifications upon the creation of new subscriptions, both to the Customer user as well as Service Provider e-mail recipients
- **Connector Updates**
 - Support for newer versions of connectors. Connectors can now add additional properties in Bootstrap and Resource properties, and also add new Groups in Resources.
 - Experimental Connector to CloudPortal Services Manager 11.0
 - Experimental Reference Connectors for:
 - Apache CloudStack 4.2
 - Ubuntu OpenStack Grizzly
 - **CCP Connector**
Product Managers can define bundles that are specific to a zone.
- **OS Support**
 - Support for RHEL 6.3, in addition to CentOS 6.3
- **Upgrades**
 - Upgrade path available from CloudPortal Business Manager versions 1.4.3 and later, to CloudPortal Business Manager 2.1.

Defects Fixed

This section describes the defects fixed in the release of CloudPortal Business Manager 2.1.

- DE5790: Truncated descriptions are displayed in the Configuration > Portal > Billing > Edit page.
- DE5862: Trial tenant can be created using expired promotion code.
- DE5814: Auto-created subscriptions are not terminated if the resource is not active.
- DE5372: While defining a mediation rule, both operators (Include and Exclude) are allowed for a discriminator.
- DE5793: CloudPortal Business Manager UI does not display the Unlimited checkbox while defining entitlements.
- DE5413: Incorrect bundle price is displayed when the reference catalog is revised multiple times and the bundle price is over-ridden.
- DE5395: Channel catalog does not display both, the timestamp of the current plan and the timestamp of expiry of the previous plan.
- DE5818: Power user cannot delete the subscriptions of a Master user.
- DE5771: The Utility Rate Card displays retired products.
- DE5681: While subscribing to a bundle, a Tenant can provision a resource even if the option to provision is not checked.
- DE5662: In the Pay-as-you-go model, instead of the dollar symbol, USD is used to specify the list price.
- DE5494: While defining products, you can incorrectly specify a usage type using only Exclude as the aggregate operator.
- DE5611: Changes made to accounts and users are not communicated to the cloud service.
- DE5115: On retiring a product, the page refreshes and loses the reference to the retired product.
- DE5594: Customers do not have the Recover VM option for a Destroyed VM.
- DE5980: Products are incorrectly listed on the Products page.
- DE6088: The UI does not display the Pay-as-you-go option while subscribing to a product bundle.
- DE5950: The Root user password cannot be modified.

Defects Fixed

- DE5955: By default, the Ops Admin user does not have the necessary permissions to schedule a maintenance operation.
- DE5956: The Master user cannot view the service tickets filed by a normal user of the account.
- DE5958: Changing currency in the Pay-as-you-go option does not work.
- DE5959: For retail accounts, if pre-authentication is enabled, subscriptions do not get activated.
- DE5962: While adding entitlements in a product bundle, you can specify a negative value for the product units.
- DE5991: During subscription creation, instead of the selected zone name, its UUID is displayed.
- DE5997: Subscription creation fails if you deploy a VM from a user-defined ISO.
- DE6090: While browsing through the catalog, an anonymous user cannot view the bundle price in different currencies.

Known Issues and Limitations

1. The CloudPortal Business Manager database installation fails if you use the same server and username as CloudPlatform.
2. Server startup fails on a setup migrated from CloudPortal Business Manager 2.0 to 2.1, when it has cloud-portal encryption enabled.
3. If you delete a user and create another user with the same name as the deleted user, then a warning message pop-up will be shown when you enable cloud services for the user.
4. Billing invoices are generated for terminated accounts.
5. "Email invoice as PDF" option is not available in the UI for Root user and Finance Admin.
6. If the billing job is not executed for more than a day, it may result in incorrect billing invoices.
7. When you reconfigure a subscription (change a subscription), the "Your Selection" pane displays template UUID instead of template name.
8. The job failure email contains "{0} Job failed" in the subject line.
9. In Internet Explorer 9, the download template link does not work. As a workaround, use any other browser or http download tools to download the template.
10. In pay-as-you-go mode, resource type's SINGLETON constraint is ignored.
11. The invoice details section may not display name and unit of retired products.
12. Numerical Values in Fusion Charts on the dashboard will not be displayed in the region format of the user.
13. When Account and User Lifecycle operations (such as update, terminate) are performed, CloudPortal Business Manager will not invoke update, unregister handlers for OSS services.
14. When CloudPortal Business Manager is started first time, CloudPortal Business Manager server logs will contain liquibase changeset messages at ERROR log level. You can ignore these messages.
15. Sometimes the lock on table "ACTIVEMQ_LOCK" under schema "cloud_portalmq" is not released when the machine on which CPBM server runs shut down, without stopping CPBM. In such cases, when machine is restarted, it takes very long time for CPBM to start because the ActiveMQ broker cannot acquire the required lock.

Workaround: Kill the db process which holds the lock or restart the CPBM database.

CCP Connector Issues

1. Ensure only numeric value is provided for custom disk size when you provision a new volume.

Localization Issues

1. While enabling a cloud service, some UI pages do not display localized text.
2. Email templates include wrong strings in Simplified Chinese and Japanese.
3. VPC names with localized characters are not supported.
4. “**Payment Received**” email template in Simplified Chinese and “**Reset Password**” email template in Japanese contain garbled characters.
5. Invoice PDF content template is available only in English.

Limitations

1. OSS settings cannot be changed after a customer account is created in CloudPortal Business Manager.
2. Minimum length of user names in CloudPortal Business Manager is 5.
3. To issue credit to an account, the account has to be in Active state.
4. Localization is not supported for the experimental connectors: OpenStack, CloudPortal Services Manager, and Apache CloudStack.

Maintenance Release

This section describes the enhancements, changes, fixed issues, and known issues provided in the maintenance releases of the CloudPortal Business Manager software.

- [2.1.2](#)
- [2.1.1](#)

2.1.2

This section contains the release information for the maintenance release of the CloudPortal Business Manager version 2.1.2. It contains important information and defects fixed and known issues in the maintenance release of the CloudPortal Business Manager version 2.1.2.

Important Notes

- From CPBM 2.1.1 onwards, a new directory `connectors_ext` has been created under `/usr/share/vts3/` directory. This directory contains the `com.citrix.cpbm.connector.ext.plan` and the `connector-ext.properties` file. Customers need to deploy their connectors into this directory and respective plan needs to be edited. Contents of `com.citrix.cpbm.plan` and `com.citrix.cpbm.connector.plan` should not be modified.
- In CPBM 2.1.0, if you had deployed your own connectors (other than CPBM's own connectors namely, ACS, CCP) under the `connectors/` directory, then move those connector bundles to the `connectors_ext` directory and edit the plan appropriately.

Example:

1. Move your connector artifact (for example):

```
<artifact type="bundle" name="<bundle.artifact.id>" version="[<x.x.x>, <x.x.x>]"/>
```

from `com.citrix.cpbm.connector.plan` located at `/usr/share/vts3/connectors` and add it to `com.citrix.cpbm.connector.ext.plan` located at `/usr/share/vts3/connectors_ext` after this line:

```
<artifact type="configuration" name="connector-ext"/>
```

2. Copy the connector OSGi service bundle in the Virgo's `/usr/share/vts3/connectors_ext` directory.
- As part of the upgrade, the contents of the `/usr/share/vts3/patches/` directory will be removed as the newer version will include all the fixes. After the upgrade, it will contain the `patches.plan` along with the `patches.properties` file. It is recommended that you back up the `patches` folder before the upgrade.

Note: Do not delete the `patches.properties` and `connector-ext.properties` files.

Defects Fixed

- **DE6854:** Security fix made on OWASP CSRFGUARD to overcome cross domain JS Hijacking using Java script object overrides.
- **DE7656:** Product bundles are not displayed in the end user catalog when "Zone Id" is not present in template/ISO attributes.
- **DE7500:** Editing a product bundle to add a provisioning constraint does not retain the previous value of the provisioning constraint.
- **DE7450:** Wrong parameter name in the service enablement BSS API.
- **DE7436:** User is not able to change his profile language.
- **DE6446:** For OSS connectors, update and terminate accounts/users are not called.
- **DE6796:** Upgrade to CPBM 2.1.x fails if the root username is changed.
- **DE7437:** Currency precision is lost under entitlements of product bundle on anonymous catalog page.
- **DE7275:** Trial account does not get suspended automatically on reaching the spend limit.
- **DE7427:** Billing job fails if the validation for a subscription's resource returns an exception.
- **DE7331:** Error in CCP connector while parsing negative timezones in async job API calls to CCP.
- **DE7296:** Descriptions for channel.enable.caching and date.format under Administration > Configuration > Portal > Server do not describe the configuration.
- **DE7410:** Any usage attributed to the service provider user impacts the billing job.
- **DE7657:** Log of workflow job errors are persisted to the database instead of the server log file.
- **DE7044:** If there are usages found that do not match with any product on CPBM, the email sent to the support admin contains IDs of usages that shows zero for all usage IDs.
- **DE7658:** Change in user level resource limits for shared account mode in CCP are updated.
- **DE7605:** In the CCP Connector change subscription on a utility subscription for VM fails.

Known issues and Limitations

- The CloudPortal Business Manager database installation fails if you use the same server and username as CloudPlatform.
- Server startup fails on a setup migrated from CloudPortal Business Manager 2.0 to 2.1, when it has cloud-portal encryption enabled.
- "Email invoice as PDF" option is not available in the UI for Root user and Finance Admin.
- If the billing job is not executed for more than a day, it may result in incorrect billing invoices.
- When you reconfigure a subscription (change a subscription), the "Your Selection" pane displays template UUID instead of template name.
- In Internet Explorer 9, the download template link does not work. As a workaround, use any other browser or http download tools to download the template.
- In pay-as-you-go mode, resource type's SINGLETON constraint is ignored.
- Numerical Values in Fusion Charts on the dashboard will not be displayed in the region format of the user.
- When Account and User Lifecycle operations (such as terminate) are performed, CloudPortal Business Manager will not invoke unregister handlers for OSS services.
- For a Power User and Normal User, 'Edit My Profile' does not work if the phone number is not specified.
- When CloudPortal Business Manager is started first time, CloudPortal Business Manager server logs will contain liquibase changeset messages at ERROR log level. You can ignore these messages.
- Reconfigure subscription is not working for the utility subscription when only compute offering is changed.

Warning: If you upgrade from CPBM 1.4.x version to CPBM 2.1.x version and if you have used "Default" system channel as the default channel, then a Trial Account or a Retail Account with promotional code cannot be created. **Workaround:** After the upgrade, set the channel code to a non-null and unique string (for example, DEFAULT).

CCP Connector Issues:

- Ensure that disk size is entered in the Disk Size in GB (Only required for Custom Disk Offerings) text field when you provision a new volume using custom disk offering.

Localization Issues:

- While enabling a cloud service, some UI pages do not display localized text.
- Email templates include wrong strings in Simplified Chinese and Japanese.
- VPC names with localized characters are not supported.

- “**Payment Received**” email template in Simplified Chinese and “**Reset Password**” email template in Japanese contain garbled characters.
- Invoice PDF content template is available only in English.

Limitations:

- OSS settings cannot be changed after a customer account is created in CloudPortal Business Manager.
- Minimum length of user names in CloudPortal Business Manager is 5.
- To issue credit to an account, the account has to be in Active state.
- Localization is not supported for the experimental connectors: OpenStack, CloudPortal Services Manager, and Apache CloudStack.

2.1.1

This section contains the release information for the maintenance release of the CloudPortal Business Manager version 2.1.1. It contains important information and defects fixed and known issues in the maintenance release of the CloudPortal Business Manager version 2.1.1.

Important Notes

- In CPBM 2.1.1, a new directory `connectors_ext` has been created under `/usr/share/vts3/` directory. This directory contains the `com.citrix.cpbm.connector.ext.plan` and the `connector-ext.properties` file. Customers need to deploy their connectors into this directory and respective plan needs to be edited. Contents of `com.citrix.cpbm.plan` and `com.citrix.cpbm.connector.plan` should not be modified.
- In CPBM 2.1.0, if you had deployed your own connectors (other than CPBM's own connectors namely, ACS, CCP) under the `connectors/` directory, then move those connector bundles to the `connectors_ext` directory and edit the plan appropriately.

Example:

1. Move your connector artifact (for example):

```
<artifact type="bundle" name="<bundle.artifact.id>" version="[<x.x.x>, <x.x.x>]"/>
```

from `com.citrix.cpbm.connector.plan` located at `/usr/share/vts3/connectors` and add it to `com.citrix.cpbm.connector.ext.plan` located at `/usr/share/vts3/connectors_ext` after this line:

```
<artifact type="configuration" name="connector-ext"/>
```

2. Copy the connector OSGi service bundle in the Virgo's `/usr/share/vts3/connectors_ext` directory.
- As part of the upgrade, the contents of the `/usr/share/vts3/patches/` directory will be removed as the newer version will include all the fixes. After the upgrade, it will contain the `patches.plan` along with the `patches.properties` file. It is recommended that you back up the `patches` folder before the upgrade.

Note: Do not delete the `patches.properties` and `connector-ext.properties` files.

Defects Fixed

- Fixes for seamless upgrade to CPBM 2.1.1 and for significant reduction in upgrade time.
- Fixes to improve Catalog page performance.
- **CPBMRAL-1434**: After upgrading from CPBM 1.4.x to CPBM 2.1.0, the email verification link does not work if an account creation happens before the upgrade while its email verification happens after the upgrade.
- **CPBMRAL-1432**: While logging in to CPBM, if you select the username from the saved username drop down list and click Enter, the login form is submitted without a password request and login fails.
- **CPBMRAL-1415**: Chargeable unit returns no values in the BSS API for closeout invoice items.
- **CPBMRAL-1139**: Customs fields are empty for terminated or deleted accounts.
- **CPBMD-1123**: Any changes made to an invoice content template are not saved.
- **CPBMD-1118**: Partial error occurs when you run a batch job.
- **CPBMD-1109**: The calendar picker is not displayed while scheduling activation.
- **CPBMD-1105**: The Tickets page does not display any results when the root user filters support tickets by selecting the All filter.
- **CPBM-914**: When resource sharing is enabled, all VMs stop if you deactivate one of the normal users.
- **CPBM-874**: Localization issues with the German language pack.
- **CPBMD-841**: The terms and conditions text box does not display the contents of the Terms and Conditions source file.
- **CPBMD-1179**: After integration of CPBM with CCP, the user cannot view VM consoles.
- **DE6840**: Revisioning of entitlements is not handled correctly.
- **DE6820**: Resource mode changes automatically on changing the service control.
- **DE6775**: Billing for a deleted tenant is not handled.
- **DE6776**: If the cloudstack_resource_id field of the table subscription_cloudstack_id_maps is null, then upgrade from CPBM 1.4.x to CPBM 2.1 fails.
- **DE6800**: You can add a VPC in CPBM 2.1.1.
- **DE6807** : When you have more than one service type, the UI does not display correctly the resource types when you change the service category.
- **DE6760**: When you upgrade from CPBM 1.4.x to CPBM 2.1.0, any taxes and discounts that were incurred before the system upgrade are not applied or included in the

appropriate billing period.

- **DE6780:** You get an error on accessing a retired product that has a logo.
- **DE6572:** A trial user is unable to convert to a retail account type.
- **DE6891:** If the number of products is large, the utility rate card does not display the product prices nor can you edit the product prices.

Known issues and Limitations

- The CloudPortal Business Manager database installation fails if you use the same server and username as CloudPlatform.
- Server startup fails on a setup migrated from CloudPortal Business Manager 2.0 to 2.1, when it has cloud-portal encryption enabled.
- If you delete a user and create another user with the same name as the deleted user, then a warning message pop-up will be shown when you enable cloud services for the user.
- Billing invoices are generated for terminated accounts.
- "Email invoice as PDF" option is not available in the UI for Root user and Finance Admin.
- If the billing job is not executed for more than a day, it may result in incorrect billing invoices.
- When you reconfigure a subscription (change a subscription), the "Your Selection" pane displays template UUID instead of template name.
- In Internet Explorer 9, the download template link does not work. As a workaround, use any other browser or http download tools to download the template.
- In pay-as-you-go mode, resource type's SINGLETON constraint is ignored.
- The invoice details section may not display name and unit of retired products.
- Numerical Values in Fusion Charts on the dashboard will not be displayed in the region format of the user.
- When Account and User Lifecycle operations (such as update, terminate) are performed, CloudPortal Business Manager will not invoke update, unregister handlers for OSS services.
- When CloudPortal Business Manager is started first time, CloudPortal Business Manager server logs will contain liquibase changeset messages at ERROR log level. You can ignore these messages.
- For a Power User and Normal User, Edit My Profile does not work if the phone number is not specified.

Warning: If you upgrade from CPBM 1.4.x version to CPBM 2.1.x version and if you have used "Default" system channel as the default channel, then a Trial Account or a Retail

Account with promotional code cannot be created. **Workaround:** After the upgrade, set the channel code to a non-null and unique string (for example, DEFAULT).

CCP Connector Issues:

- Ensure only numeric value is provided for custom disk size when you provision a new volume.

Localization Issues:

- While enabling a cloud service, some UI pages do not display localized text.
- Email templates include wrong strings in Simplified Chinese and Japanese.
- VPC names with localized characters are not supported.
- “**Payment Received**” email template in Simplified Chinese and “**Reset Password**” email template in Japanese contain garbled characters.
- Invoice PDF content template is available only in English.

Limitations:

- OSS settings cannot be changed after a customer account is created in CloudPortal Business Manager.
- Minimum length of user names in CloudPortal Business Manager is 5.
- To issue credit to an account, the account has to be in Active state.
- Localization is not supported for the experimental connectors: OpenStack, CloudPortal Services Manager, and Apache CloudStack.

Upgrading CloudPortal Business Manager

The supported upgrade paths to CloudPortal Business Manager are as follows:

- CPBM versions 1.4.3 through 1.4.7 to CPBM 2.1.x.
(for information on the upgrade steps, refer to [Migration Guide](#))
- CPBM 2.0.0 to CPBM 2.1.x.
- CPBM 2.1.0 to CPBM 2.1.x.
- CPBM 2.1.1 to CPBM 2.1.x

Important Notes:

- In CPBM 2.1.1, a new directory `connectors_ext` has been created under `/usr/share/vts3/` directory. This directory contains the `com.citrix.cpbm.connector.ext.plan` and the `connector-ext.properties` file. Customers need to deploy their connectors into this directory and respective plan needs to be edited. Contents of `com.citrix.cpbm.plan` and `com.citrix.cpbm.connector.plan` should not be modified.
- In CPBM 2.1.0, if you had deployed your own connectors (other than CPBM's own connectors namely, ACS, CCP) under the `connectors/` directory, then move those connector bundles to the `connectors_ext` directory and edit the plan appropriately.

Example:

1. Move your connector artifact (for example):

```
<artifact type="bundle" name="<bundle.artifact.id>" version="[<x.x.x>, <x.x.x>]"/>
```

from `com.citrix.cpbm.connector.plan` located at `/usr/share/vts3/connectors` and add it to `com.citrix.cpbm.connector.ext.plan` located at `/usr/share/vts3/connectors_ext` after this line:

```
<artifact type="configuration" name="connector-ext"/>
```

2. Copy the connector OSGi service bundle in the Virgo's `/usr/share/vts3/connectors_ext` directory.
- As part of the upgrade, the contents of the `/usr/share/vts3/patches/` directory will be removed as the newer version will include all the fixes. After the upgrade, it will contain the `patches.plan` along with the `patches.properties` file. It is recommended that you back up the patches folder before the upgrade.

Note: Do not delete the `patches.properties` and `connector-ext.properties` files.

Steps to upgrade CloudPortal Business Manager 2.0.0 to CloudPortal Business Manager 2.1.x:

1. Download the CloudPortal Business Manager v2.1.x tar ball from the CloudPortal Business Manager download site.

2. Expand the downloaded tar ball and change your current directory to the newly created folder:

```
# tar -xvf CloudPortal-2.1.x-1-centos.tar.gz  
# cd CloudPortal-2.1.x-1-centos/
```

3. Stop the CloudPortal Business Manager server:

```
# service cloud-portal stop
```

4. Before starting the upgrade, ensure that you have backed up your server and database.

5. Run the install script and select the upgrade option.

```
# ./install.sh  
Welcome to the Citrix CloudPortal Business Manager Installer. What would you like to do?  
D) Install the database server  
U) Upgrade CloudPortal Business Manager  
C) Integrate with CloudStack  
Q) Quit  
> U
```

6. Start the CloudPortal Business Manager server:

```
# service cloud-portal start
```

Note: Server startup fails on a setup migrated from CloudPortal Business Manager 2.0.0 to CloudPortal Business Manager 2.1.0 when it has cloud-portal encryption enabled.

Installation Guide

This guide provides detailed instructions for installing the CloudPortal™ Business Manager.

System requirements
Installing and configuring CPBM
Verifying - Starting and stopping CPBM

System Requirements

System requirements for the server:

- 4 vCPU , 8 GB RAM with 40 GB HDD
- Operating System (any of the following):
 - CentOS 5.6, 6.0, 6.2, or 6.3
 - RHEL 6.3
- Java 1.6
- MySQL Server 5.1.x
- Internet connectivity: The following features require internet connectivity for the system to be able to use them
 - CAPTCHA verification
 - Gravatar
 - Installation

Browser support

The following browsers are supported:

- Microsoft Internet Explorer 8 and 9
- Mozilla Firefox 19
- Google Chrome 25
- Apple Safari 5/iOS 6.0.1

Installing and Configuring CloudPortal Business Manager

Download

Download the CloudPortal Business Manager software.

System Preparation

All the operations listed below should be run as root.

Unarchive Install Package

To unarchive the install package tarball, run (where x.y.z is the version you have downloaded):

```
# tar -xvf CloudPortal-x.y.z-centos.tar.gz
```

Then navigate into the newly created CloudPortal-x.y.z-centos folder:

```
# cd CloudPortal-x.y.z-centos
```

Installing MySQL

The MySQL database may be installed on the same node as CloudPortal Business Manager or a different node.

Note: You must have a MySQL database administrator tune the MySQL database for optimal performance. MySQL's default configuration will use a small amount of RAM and does limited amount of caching. CloudPortal uses the InnoDB storage engine.

1. Run the `./install.sh` script and choose option "D" to install the MySQL database server.

Accept the license agreement before selecting the "D" option.

```
# ./install.sh
Welcome to the Citrix CloudPortal Business Manager Installer. What would you like to do?
D) Install the database server
I) Install & Configure CloudPortal Business Manager
C) Integrate with Citrix CloudPlatform
Q) Quit
> D
```

2. When prompted, provide a user name and password. You can set these to any desired value that is valid for a MySQL database. This step creates a new user account that CloudPortal will use to access the database.

Note: Database password is mandatory.

3. Exit the installation setup:

Welcome to the Citrix CloudPortal Business Manager Installer. What would you like to do?

D) Install the database server

I) Install & Configure CloudPortal Business Manager

C) Integrate with Citrix CloudPlatform

Q) Quit

> Q

4. Stop mysql service:

```
service mysqld stop
```

5. Change the MySQL DB encoding format to UTF-8:

Edit the file /etc/my.cnf and add the following lines under the [mysql] and [mysqld] sections:

```
[mysql]
default-character-set=utf8
```

```
[mysqld]
default-character-set=utf8
default-collation=utf8_unicode_ci
character-set-server=utf8
collation-server=utf8_unicode_ci
```

```
default-time-zone='+00:00'
max_connections=200
```

The time zone should be synchronized with the time setup for CloudPortal Business Manager (both must use the same timezone).

6. Start the MySQL service.

```
# service mysqld start
```

Note: Use the following command to check if the database encoding format is UTF-8:

```
mysql>show variables like "%char%";
```

```
mysql>show variables like "%coll%";
```

Language Settings for Linux

The default language setting for CloudPortal Business Manager is English.

To set the language

1. Open the file `/etc/sysconfig/i18n`
2. Set the value `LANG="en_US.UTF-8"`

Installing CloudPortal Business Manager

For each CloudPortal Business Manager node, run the `install.sh` script from the package. Select option I:

Accept the license agreement before selecting the "I" option.

```
# ./install.sh
Welcome to the Citrix CloudPortal Business Manager Installer. What would you like to do?
D) Install the database server
I) Install & Configure CloudPortal Business Manager
C) Integrate with Citrix CloudPlatform
Q) Quit
> I
```

Enter Y if you want Central Authentication Service (CAS) installed along with CloudPortal Business Manager, and N if you want to install only CloudPortal.

```
Would you like to install CAS with CloudPortal (Y/N)?
> Y
```

If yes, refer to [Enabling CAS authentication](#).

Installation begins. CloudPortal Business Manager checks for the Java version in the system and installs the required version (1.6) if it is not available.

JAVA_HOME path is automatically updated.

Note: If you install Java manually, update JAVA_HOME path in the `/etc/init.d/cloud-portal` script.

It is recommended to run all servers on UTC. This server is currently running on IST.

```
Do you wish to change this server to run on the UTC timezone (y/N)?
```

Select the required option.

CloudPortal Business Manager installation is complete.

The Configuration Tool starts automatically once the installation is complete.

If you have skipped the configuration step, use the command below to start the Configuration Tool.

You've successfully installed CloudPortal Business Manager.
Now proceed to configure the CloudPortal Business Manager.

Welcome to the Citrix CloudPortal Business Manager Configuration Tool.

- D) Install the database server
- G) Configure CloudPortal Business Manager
- C) Integrate with Citrix CloudPlatform
- Q) Quit

> G

1. Provide CloudPortal public server information:

- public host name - <FQDN or an IP address>
- public host port
- public host protocol

Ensure that the hostname provided is a valid hostname and is resolvable either in /etc/hosts or in DNS.

2.

Do you want to configure CloudPortal startup Encryption Key [y/N]:

Configure the CloudPortal Start-up Encryption key. Length of the encryption key must be **16 characters**. For security reasons, you need to provide the encryption key during CloudPortal start-up. This key will be used to encrypt password fields in cloud.properties file.

Note: If you select the encryption key option, then you must remember the encryption key because you will need to enter the key every time you start the CloudPortal Business Manager. The start-up CloudPortal option will be disabled as mentioned below:

```
[root@localhost CloudPortal-2.1.0-1-centos]# chkconfig --list cloud-portal
cloud-portal 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

However, if you do not select the encryption key option, you will NOT be asked to enter the key when you start CloudPortal. The following startup cloud-portal options are enabled as mentioned below running mode. So, when you reboot the system, Cloud-Portal service will start by default in the following mentioned running modes (3,4,5).

```
[root@localhost CloudPortal-2.1.0-1-centos]# chkconfig --list cloud-portal
cloud-portal 0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

3. Provide CloudPortal database configuration details:

- CloudPortal Database encryption key. Length of the encryption key must be **16 characters**.

- Host name or IP
- Database user

Note: Enter the user name that you have specified while installing MYSQL database.

- Database password - <Your Password used for cloud_portal DB user>

Note: Database password is not visible. Enter the password that you have specified while installing MYSQL database.

4. Provide mail sender (SMTP/SMTPS) configuration details:

- Mail protocol - smtp
- Mail host
- Mail port
- Mail user
- Mail password

Note: If SMTPS is enabled, ensure that the following lines are present in cloud.properties:

```
mail.smtps.auth=true  
mail.debug=false  
mail.smtp.starttls.enable=true  
mail.smtp.starttls.required=true
```

5. Do you want to configure LDAP [y/N]:

By default, it is N. If you select the option "y", then provide the following information:

- LDAP Host Name[localhost]
- LDAP Base[dc=cloud,dc=com]
- LDAP Userdn[cn=Manager,dc=cloud,dc=com]
- LDAP Password

6. Enter the tax percentage:

Tax Percentage[10.00]:

This is the tax percentage applied while billing the customers/tenants for the use of cloud services.

7. Provide all the active currencies (maximum six active currencies):

For example: USD, INR, GBP, EUR

8. Select one default currency for your installation from the active currency code displayed.

9. Enter the recaptcha configuration details (you can proceed with the default values, if required):

- Recaptcha public key
- Recaptcha private key
- Recaptcha server

You can generate your own recaptcha keys (public key and private key) from <http://www.google.com/recaptcha> using your Google account.

10. Quit the installation:

Welcome to the Citrix CloudPortal Business Manager Configuration Tool.

```
D) Install the database server
G) Configure CloudPortal Business Manager
C) Integrate with Citrix CloudPlatform
Q) Quit
> Q
```

The initial setup of CloudPortal Business Manager is completed.

You can start CPBM using the command:

```
service cloud-portal start
```

(If you have configured the CPBM startup encryption key, you will be prompted for the key.)

Enabling Network Security for Administrator Login

CloudPortal Business Manager provides a security mechanism using which login to CloudPortal Business Manager for administrators can be restricted to only a fixed set (or range) of IP addresses.

1. Log in as root to the host running CloudPortal Business Manager.
2. Edit the file `/usr/share/vts3/repository/prop/cloud.properties` and set value to the property "network.ip.address.range" :

For example: `network.ip.address.range=10.147.18.1/24,10.103.1.1/24.`

You can specify comma separated list of CIDR. An empty value allows all ranges of IP addresses.

3. Save the file.
4. Restart CloudPortal Business Manager.

```
# service cloud-portal restart
```


Enabling CAS Authentication

CloudPortal supports the use of Central Authentication Service (CAS) for single sign-on, which enables access to multiple password-protected systems after logging in just once. To make this work, you need to update CAS with the location of the cloud_portal database and list the services with which CAS will be used.

1. Log in as root to the host running CloudPortal Business Manager.
2. Edit the file `/usr/share/vts3/repository/prop/cloud.properties` and set these parameters.
 - Un-comment the following property to enable CAS. By default, native authentication is enabled.

`vmops.authentication.service=cas`
 - `cas.login.url`: Full public URL (including `http://` or `https://`, hostname, optional port, and path) of the login page displayed to the user for single signon. For example, `http:// mypublichostname:8080/cas`.
 - `cas.logout.url`: Full public URL of the page where a user can log out of single sign-on. For example, `http://mypublichostname:8080/cas/logout`.
 - `cas.service.url`: Full public callback URL used by CAS to redirect back to CloudPortal Business Manager. For example, `http://mypublichostname:8080/portal/j_spring_cas_security_check`.
 - `cas.validator.url`: Full private URL used in internal calls from CloudPortal Business Manager server to CAS server to validate the CAS ticket. This should be a URL that is not accessible from outside your installation. If using HTTPS, this site must have a valid certificate issued by a trusted certificate authority. For example, `http://myprivatehostname:8080/cas`.
3. If any other application needs to be authenticated via CAS, then edit the `/usr/share/vts3/repository/prop/deployerConfigContext.xml` file. Refer to [Adding a new service to CAS](#).
4. Restart CloudPortal Business Manager.

```
# service cloud-portal restart
```

Integration with CloudPlatform

On each CloudPlatform server, run the `./install.sh` script and select Integrate with CloudPlatform from the menu. This will place files within CloudPlatform that are needed for single sign-on integration with CloudPortal. Note that after the files are copied over for integration, you will no longer be able to access the CloudPlatform management server using `http://<hostname>:8080/client`. It is recommended to access the CloudPlatform management UI by first logging into CloudPortal and clicking the "Launch Cloud Console" button on the dashboard. As an alternative, the CloudPlatform management UI can be accessed directly by using `http://<hostname>:8080/client/?direct=true`.

```
# ./install.sh
Welcome to the Citrix CloudPortal Business Manager Installer. What would you like to do?
D) Install the database server
I) Install & Configure CloudPortal Business Manager
C) Integrate with Citrix CloudPlatform
Q) Quit
> C
```

Multinode Installation

For multinode installation:

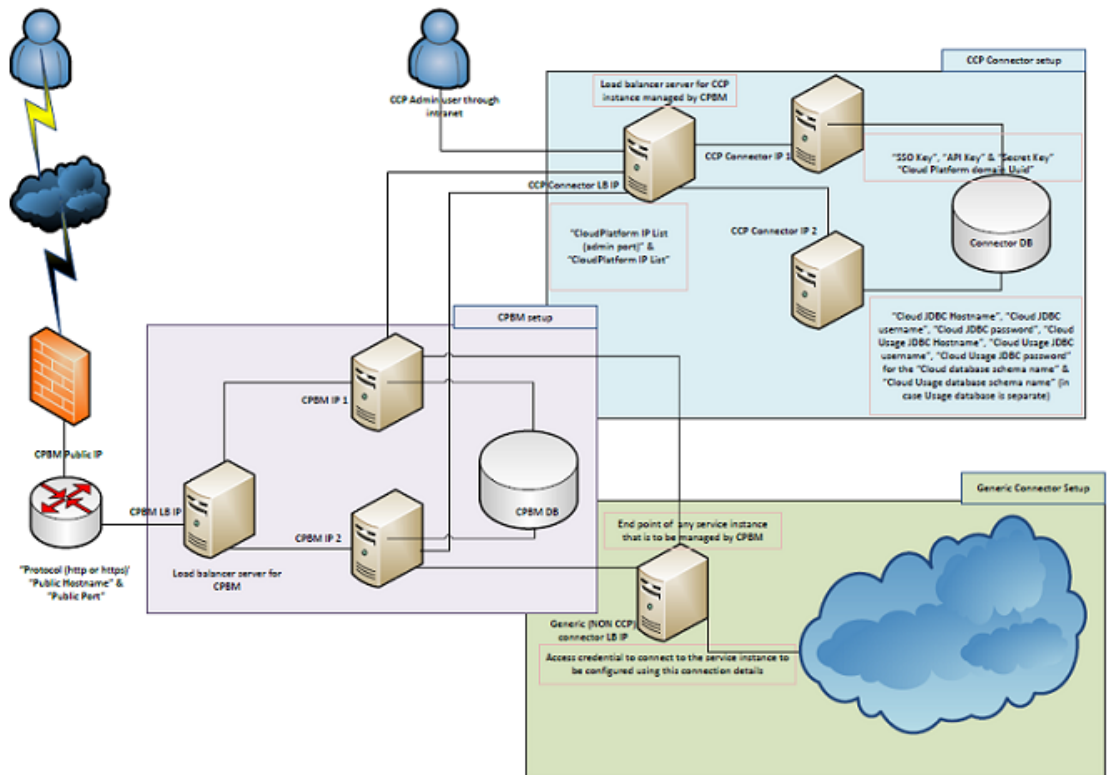
Note: Installation of database server is not required for subsequent installation of CPBM because all the CPBM servers use the same database server.

1. Install CPBM on the required servers
2. After installation, make sure that configuration values provided under "Citrix CloudPortal Business Manager Configuration Tool" are same across all the servers
3. Log in to the OS console of each CPBM server and change the following lines in `/usr/share/vts3/repository/prop/cloud.properties`

```
vmops.jms.url=tcp://<Your IP or hostname>:9125
vmops.jms.client.url=failover:(tcp://<your IP or hostname>:9125,
tcp://< CPBM Server 1 IP or hostname>:9125,
tcp://< CPBM Server 2 IP or hostname>:9125)
```

where `<your IP or hostname>` is the IP/hostname of the server from where you are entering the value and you must specify the IP/hostname of all the CPBM servers separated by comma.

Make sure that port 9125 is open on all the CPBM servers.



Setting up a Proxy Server

In order to add a proxy server to front CloudPortal servers, the steps given below must be followed.

The proxy server can be added in a variety of ways. The steps given below illustrate one typical technique using Apache HTTP Server running on CentOS 5.x and CentOS 6.x.

- On the Apache server, install the module that provides SSL and TLS support.
yum install httpd mod_ssl
- Create the file /etc/httpd/conf.d/cloud.conf, and add proxying (and, if needed, load balancing) directives

For a two-node installation where Apache and CloudPortal are run on separate nodes, use rules like the following. Replace CloudPortalNode with the private hostname or IP of your own machines.

HTTP Proxy configuration:

```
ProxyPreserveHost on
ProxyPass /portal http://CloudPortalNode:8080/portal
ProxyPass /cas http://CloudPortalNode:8080/cas
ProxyPassReverse /cas http://CloudPortalNode:8080/cas
ProxyPass / http://CloudPortalNode:8080/portal
```

AJP Proxy configuration:

```
ProxyPass /portal ajp://CloudPortalNode:20410/portal
ProxyPass /cas ajp://CloudPortalNode:20410/cas
ProxyPass / ajp://CloudPortalNode:20410/portal
```

Note: CloudPortalNode will display 8009 as the default value. Edit the file /usr/share/vts3/config/tomcat-server.xml and change the value of the port from 8009 to 20410 at this line:

```
<Connector port="20410" protocol="AJP/1.3"
redirectPort="8443"
URIEncoding="UTF-8"/>
```

For a three-node installation where Apache is on one node and CloudPortal is on two additional nodes, use directives like the following. Replace the CloudPortalNode placeholders below with the private hostnames or IPs of your own machines.

```
<Location /portal>
Header add Set-Cookie "rte=.{BALANCER_WORKER_ROUTE}e; path=/portal; HttpOnly"
env=BALANCER_ROUTE_CHANGED Header edit Set-Cookie (.*) "$1; Secure"
</Location>
<Proxy balancer://portalcluster>
BalancerMember ajp://CloudPortalNode1:20410 route=ps1
BalancerMember ajp://CloudPortalNode2:20410 route=ps2
</Proxy>
ProxyPass /portal balancer://portalcluster/portal stickysession=rte
```

If SSL is enabled on the proxy server, then add the below configuration to redirect all requests made over HTTP to HTTPS:

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]
```

Recommended settings: These settings are to mitigate attacks pertaining to POODLE (Padding Oracle On Downgraded Legacy Encryption).

You need to edit the Apache configuration located at /etc/httpd/conf.d/ssl.conf.

1. Add the following line to your Apache configuration with other SSL directives.
 - For httpd version 2.2.23 and newer, specify all protocols except SSLv2 and SSLv3:

```
SSLProtocol ALL -SSLv2 -SSLv3
```
 - For httpd version 2.2.22 and older, only specify TLSv1. This is treated as a wildcard for all TLS versions:

```
SSLProtocol TLSv1
```
2. Once done, restart your Apache HTTP Server:

```
sudo service httpd restart
```

In order to check that the connection does not go through via SSLv3, run:

```
openssl s_client -connect <ip-address>:<port> -ssl3
```

- Add the following lines to configure CAS

```
<Location /cas>
Header add Set-Cookie "rte=.%{BALANCER_WORKER_ROUTE}e; path=/cas; HttpOnly"
env=BALANCER_ROUTE_CHANGED Header edit Set-Cookie (.*) "$1; Secure"
</Location>
ProxyPass /cas ajp://CloudPortalNode:20410/cas
```

- Add the following line to avoid CSRF related errors:

```
ProxyPreserveHost on
```

- If you modify the configuration later, such as to add more nodes, use this command to reload the configuration:

```
# apachectl graceful or # service httpd restart
```

Note: If you front the Tomcat server with an Apache proxy, set the ProxyPreserveHost parameter to ON to avoid logon issues.

Recommended settings: The KeepAlive parameter is set to ON in the /etc/httpd/conf/httpd.conf file.

- [Optional] Add the lines given below to enable compression of the response:

```
SetOutputFilter DEFLATE
AddOutputFilterByType DEFLATE text/html text/css text/plain text/xml text/json application/x-javascript
```

It is recommended that you enable compression of response at reverse proxy server (web server). It improves the page load time.

Logging In

You should now be able to access CloudPortal from the Web browser. Go to `http://<your IP or hostname>:8080/portal/`. Use the following credentials to login:

User name: root

Password: Portal123#

Configuring LDAP

Refer to [Integrating the Directory Server](#).

Verifying - Starting and Stopping CloudPortal Business Manager

To start, stop, or restart CloudPortal Business Manager, log in as the root user to the host where CloudPortal Business Manager is running and use the following commands.

Starting CloudPortal Business Manager

```
# service cloud-portal start
```

Stopping CloudPortal Business Manager

```
# service cloud-portal stop
```

Restarting (stop, then start again) CloudPortal Business Manager

```
# service cloud-portal restart
```

Administration Guide

CloudPortal™ Business Manager allows you to take up several administrative tasks. There is an exclusive Administration module where you can set various parameters and manage the administrative tasks.

When CloudPortal installation is set up, you can change the set of available configuration parameters depending on your business needs. Each CloudPortal customer can have a different set of parameters as well as have different values set for the same parameter.

The configuration parameters can be used to control everything from the URL of your company's home page to the number of failed login attempts you allow users before locking their accounts.

To modify configuration parameters, you can use the UI or edit the file `/etc/cloud/portal/cloud.properties`.

Configuring Portal Settings

1. To configure portal setting parameters, log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Settings > Configure.
5. Click **Edit** and set the following values:

Settings - Parameter Name	Description
public.protocol	Public-facing protocol used for URLs embedded in text of outgoing email.
public.host	Public-facing DNS host name (or IP address) used for URLs embedded in text of outgoing email. Ensure that the hostname provided must be a valid hostname and resolvable either in /etc/hosts or DNS.
public.port	Public-facing port used for URLs embedded in text of outgoing email.
context.path	Context path of portal server (default is /portal)
helpdesk.phone	Main Help desk phone number.
helpdesk.url	Help desk URL
errorNotification.prefix	Prefix to use in Subject line of error notification email, to indicate the origin of environment error.
default.currency	Default currency for new customer accounts. It has to be one of the active currencies in the system. Use an ISO 4217 currency code, such as USD, INR, SGD, EUR, JPY.
image.path	Path to images that are used in automatically generated emails. Specify the absolute path of the image.
search.customFields	The custom fields for tenants which can be used in advanced search.
temp.path	Path to temporary directory for placing zip files before downloading or sending in email.
default.timezone	Default preferred time zone for users. New users will have this time zone as their default time zone.

Configuring Portal Settings

images.uploadPath	Absolute path to images directory for storing custom image logos, favicon etc. Note: Set the value to a valid existing path in your server file system.
proxy.server.ipList	Proxy server/load balancer IP addresses of the Cloud Portal setup, separated by commas.
record.outgoing.email	Enables saving of all outgoing emails into database table "sent_emails".
services.datapath	This is the path where service related files like serviceDefinition.xml, images, external jsp files, and terms and conditions file are copied from the OSGi connector bundle.
company.name	Company/Service Provider name.
support.email	Support Email address that is used in links/message, on portal for assistance.

6. Click **Save** and then, click **Close**.

Configuring Billing Settings for the Portal

1. To configure your billing settings, log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Billing > Configure.
5. Click Edit and set the following values:

Parameter Name	Description
billing.logPath	Directory where file with monthly invoice run failures by tenant are recorded.
spendBudget.firstAlert	First Spend Budget alert (in percentage of Spend Budget).
spendBudget.secondAlert	Second Spend Budget alert (in percentage of Spend Budget)
spendLimit.creditAlert	Send Credit Limit alert to internal Credit personnel when this percentage of Credit Limit has been used.
billingPeriod.type	Set to 1 to bill every <x> days, 2 to bill on a certain day of the month, 3 to bill monthly on the account creation date. Set com.citrix.cpbm.portal.billing.billingPeriod.config to the desired <x> or day of the month.
billingPeriod.config	Billing day(s) for use with com.citrix.cpbm.portal.billing.billingPeriod.type. Allowed values: 1-28. For example, set to 28 if com.citrix.cpbm.portal.billing.billingPeriod.type is 1, 15 if com.citrix.cpbm.portal.billing.billingPeriod.type is 2. Leave empty if com.citrix.cpbm.portal.billing.billingPeriod.type is 3.
billing.generateBillingActivity	Setting it to Y will enable detailed logging of the usage, rating and billing activity, on a daily basis. Valid Values ("Y", "N").
invoiceHandle.sequence	Invoice Handle Sequence Name used for the sequence generation.
export.format	Path to export format file of usage details (XSL).
export.fileExtension	File extension for export file of usage details.
subscription.preAuthorizationFactor	Credit Card Pre-authorization factor.
exportFormat.pdf	Path to PDF export format file of usage details (XSL).

Configuring Billing Settings for the Portal

alerts.maxSpendAlertNoti fications	Maximum number of spend alert notifications sent on each alert trigger.
generate.InvoiceEmail	Send the invoice PDF as an attachment in the mail to the Master User and Billing Admin when the invoice generation job runs.
roundingMode.charge	Set this rounding mode if charges need to be rounded to currency precision. Value should be one of these (UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN).
roundingMode.discount	Set this rounding mode if discounts need to be rounded to currency precision. Value should be one of these (UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN).
roundingMode.tax	Set this rounding mode if taxes need to be rounded to currency precision. Value should be one of these (UP, DOWN, CEILING, FLOOR, HALF_UP, HALF_DOWN, HALF_EVEN).

6. Click **Save**.

7. Click **Close**.

Configuring Marketing Settings for the Portal

1. To configure the portal marketing setting parameters, log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Marketing Wrapper > Configure.
5. Click **Edit** and set the following parameters:

Parameter Name	Description
home.url	URL of your company home page.
marketing.url	URL of the home page of the marketing wrapper for your cloud service.
support.url	Support URL to be shown at the bottom of the CloudPortal page.
blog.url	URL of the blog of your organization to be shown at the bottom of the CloudPortal page.
forum.url	URL of the user forum to be shown at the bottom of the CloudPortal page.
contact.url	Information about how to contact your organization to be shown at the bottom of the CloudPortal page.
privacy.url	URL of privacy policy of your organization to be shown at the bottom of the CloudPortal page.
tou.url	URL of Terms of Use of your organization to be shown at the bottom of the CloudPortal page.
help.url	URL to the Help documentation.

6. Click **Save**.
7. Click **Close**.

Configuring Appearance Settings for the Portal

1. To configure the portal appearance parameters, log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Appearance > Configure.
5. Click **Edit** and set the following parameters:

Parameter Name	Definition
currency.format	Display format for the currency.
currency.precision	The maximum number of digits to show after decimal point across all currencies.
currency.roundingMode	Rounding mode to use in case of scaling a currency to a strictly lower precision. Use 0 for rounding up, 1 for rounding down, 2 for ceiling, 3 for floor, 4 for half up or 5 for half down.

6. Click **Save** and then, click **Close**.

Configuring Server Settings for the Portal

1. To configure the server settings for the portal, log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Server > Configure.
5. Click **Edit** and set the following parameters:

Parameter Name	Description
username.duplicate.allowed	Setting this to true will enable suffixes where usernames will have to be disambiguated with suffix. The suffix field will show up in login and the user will have to type it in to determine which account the username is tied to. See also login.screen.tenant.suffix.dropdown.enabled.
login.screen.tenant.suffix.dropdown.enabled	At time of login, show drop down for suffix. Values can be true or false.
signature.timeout	Validity of signature timestamp (seconds).
channel.enable.caching	Enable Channel Caching (true/false).
date.format	Date Format(yyyy-MM-ddTHH:mm:ss.SSSZ).
catalog.payAsYouGoMode	Show "Pay As You Go" Mode only, while browsing catalog. Expected values true/false.

6. Click Save and then click Close.

Configuring Internal Email Addresses

1. To configure your internal email addresses, log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Address book > Configure.
5. Click **Edit** and set the following values:

Parameter Name	Value	Description
addressbook.adminEmail	<email-id>	Admin email address to display in the From field of outgoing Emails sent to customers.
addressbook.accountApprovalEmail	<email-id>	Email address to which account activation request is sent.
addressbook.salesSupportEmail	<list of email-ids>	E-mail Distribution List for Sales Support team.
addressbook.helpDeskEmail	<email-id>	Email address to be used to support the mail2case functionality.
addressbook.financeEmail	<email-id>	Email Alias for finance team.

6. Click **Save** and then, click **Close**.

Configuring CRM Email Addresses

1. To configure Email addresses for customer relationships, log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > CRM > Notifications.
4. Click Configure and set the following parameters:

Parameter Name	Value	Description
supportEmail	<email-id>	Email address for the support team. This value is used in FROM field of outgoing emails to customers.
internalEmail	<email-id>	Email address list for the internal team. This value is used in FROM field of internal emails.
accountsOutboundEmail	<email-id>	Email address to receive the notifications on payments failure and on crossing the spend limit thresholds.
salesEmail	<email-id>	Email address to display in FROM field of email sent to customer, when Retail Postpaid accounts exceed credit limits.

5. Click **Save** and then, click **Close**.

Integrating the Directory Server

The integration of CloudPortal Business Manager with Directory server (specifically Active Directory) can be done in both modes, PUSH & PULL. CloudPortal Business Manager supports either the PUSH or the PULL mode for any of the installations. However, these modes have to be configured during deployment.

Note: No change in the mode is supported once the configuration of CloudPortal Business Manager server is completed.

Pre-configuration for PUSH Mode (even for PULL mode, if LDAPS is to be used)

1. Install the security certificate exported from directory server to security keystore - Sample Command (needs modification based on system configuration):
 - `keytool -import -alias <alias> -keystore <path-to-jre>/jre/lib/security/cacerts -file <cer-file-location-and-name>`
 - `keytool -import -alias alias -keystore /usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0.x86_64/jre/lib/security/cacerts -file certfromtheAD.cer`
2. Restart the CloudPortal Business Manager server post certificate install.

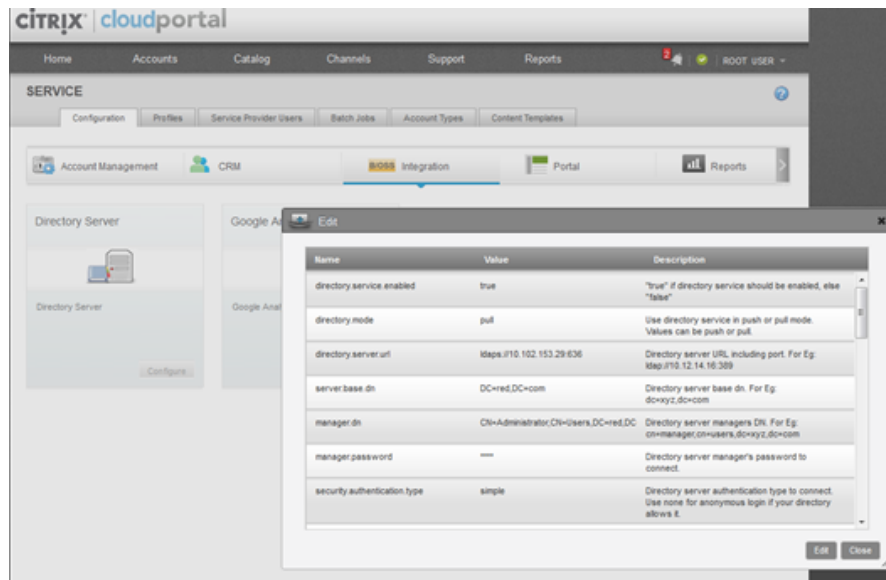
Pull Mode

In the PULL mode, CloudPortal Business Manager can import the user details from the Directory server. Once user is imported & created in the CloudPortal Business Manager, user gets an email for verification. After user verifies the email, he will be able to log into the CloudPortal Business Manager system. While logging in, user credentials are passed to Directory Server for authentication. Upon successful authentication from the Directory server, CloudPortal Business Manager allows the user to log in. CloudPortal Business Manager does not store any password in its database once this setting is enabled.

Note: In Pull mode, user password reset, forgot password, and Signup links are disabled. Any password related change needs to happen on the directory server directly.

Configuring PULL mode

1. Log in as Root user to change administration settings.
2. Go to Administration > Configuration > B/OSS Integration.
3. Click **Directory Server > Configure**.



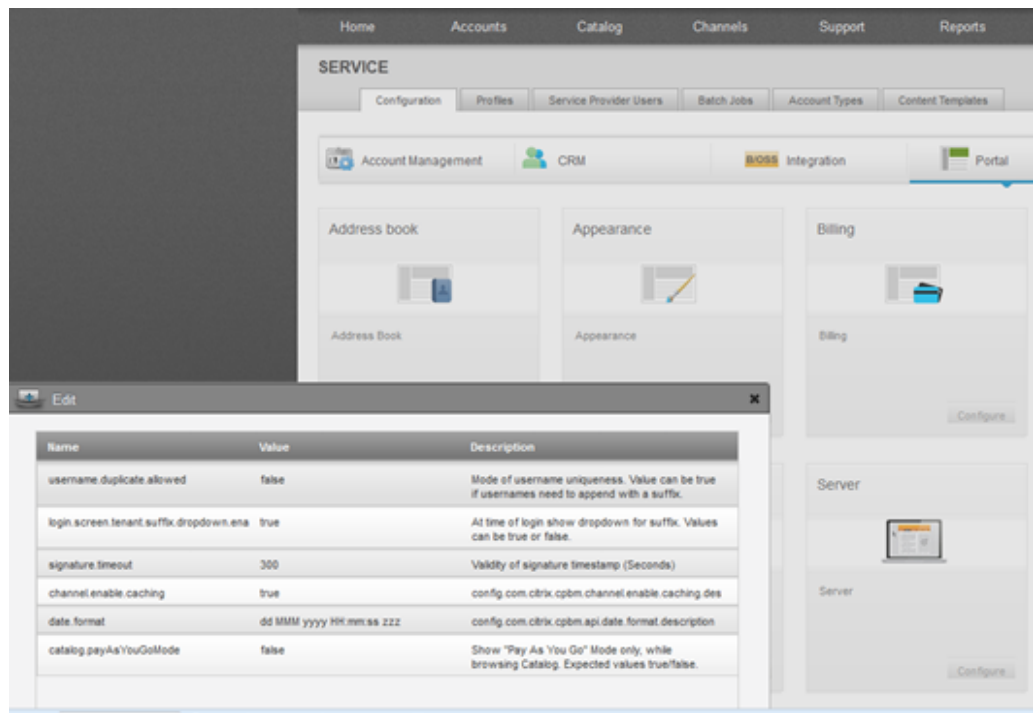
4. Change the mode to pull and directory server enable to true. Configure the Details to map to the Directory server configuration.
5. Directory server URL can be ldap or ldaps & corresponding ports (for example, ldap:389, ldaps:636).

Name	Value	Description
directory.service.enabled	true	"true" if directory service should be enabled, else "false"
directory.mode	pull	Use directory service in push or pull mode. Values can be push or pull.
directory.server.url	ldaps://10.102.153.29:636	Directory server URL including port. For Eg: ldap://10.12.14.16:389
server.base.dn	DC=red,DC=com	Directory server base dn. For Eg: dc=xyz,dc=com
manager.dn	CN=Administrator,CN=Users,DC=red,DC	Directory server managers DN. For Eg: cn=manager,cn=users,dc=xyz,dc=com
manager.password	****	Directory server manager's password to connect.
security.authentication.type	simple	Directory server authentication type to connect. Use none for anonymous login if your directory allows it.
handle.to.match.in.directory.server	(sAMAccountName={0})	Handle of directory server with which entered username will be matched. For ex. (cn={0}) will try to match with cn.
user.first.name.mapping	givenName	First name attribute name in directory.
user.last.name.mapping	sn	Last name attribute name in directory.
user.email.mapping	mail	Email address attribute name in directory.
company.name.mapping	company	Company name attribute name in directory.
country.mapping	c	Country attribute name in directory.
state.mapping	st	State attribute name in directory.
user.city.mapping	l	City attribute name in directory.
street.address.1.mapping	streetAddress	Street address 1 name in directory.
street.address.2.mapping		Street Address 2 attribute name in directory.

Any other details in the same configuration page are not used for the PULL mode.

Configuration to allow the Duplicate Username

1. Log in as Root user to change administration settings.
2. Go to Administration > Configuration > Portal.
3. Click Server > Configure.



Note: Change the setting "username.duplicate.allowed" to true. If suffix dropdown is required, change "login.screen.tenant.suffix.dropdown.enabled" to true, false for this field results in a text box appearing instead of dropdown for the user log in page.

PUSH Mode

In the PUSH mode, CloudPortal Business Manager can add the user & user details to the Directory server. Once user is created in CloudPortal Business Manager, his details will be listed in the Directory server. Upon email verification, user is asked to set the password, which is set in the Directory server. CloudPortal Business Manager supports edit user, reset password & user sign-up operations through PUSH mode. Duplicate user name & suffix features are not supported in PUSH mode as of CloudPortal Business Manager 2.1. As in case of PULL, even in PUSH mode, user passwords are not stored in CloudPortal Business Manager DB.

Note: When PUSH mode is enabled, channels on which accounts can be created are to be configured in CloudPortal Business Manager (even channels are created in Directory server).

For PUSH mode, Directory server URL has to be ldaps & port has to be 636.

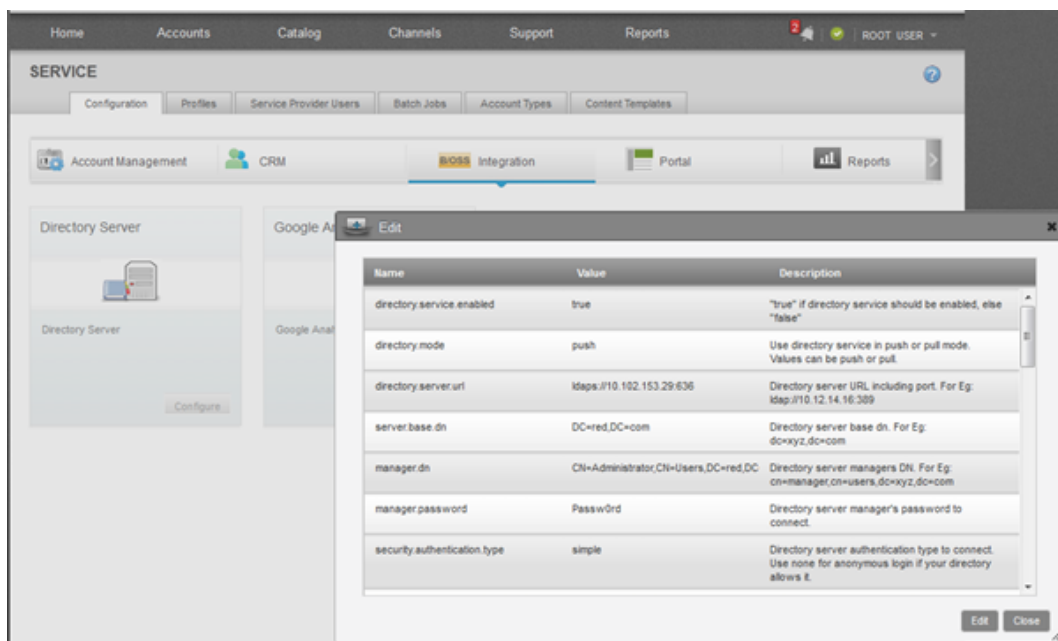
Change the first.name.mapping property to givenName which by default comes to cn to avoid multiple values of cn.

CloudPortal Business Manager does not store any password details when directory server is enabled.

Configuring PUSH Mode

1. Log in as Root user to change administration settings.

2. Go to Administration > Configuration > B/OSS Integration.
3. Click **Directory Server > Configure**.



Sample configuration for connecting the Directory server (specifically Active Directory) to CloudPortal Business Manager 2.x

If the Directory server is Active Directory, make sure the set "is.active.directory" flag to true.

Integrating the Directory Server

Name	Value	Description
directory.service.enabled	true	"true" if directory service should be enabled, else "false"
directory.mode	push	Use directory service in push or pull mode. Values can be push or pull.
directory.server.url	ldaps://10.102.153.29:636	Directory server URL including port. For Eg: ldap://10.12.14.16:389
server.base.dn	DC=red,DC=com	Directory server base dn. For Eg: dc=xyz,dc=com
manager.dn	CN=Administrator,CN=Users,DC=red,DC	Directory server managers DN. For Eg: cn=manager,cn=users,dc=xyz,dc=com
manager.password	Passw0rd	Directory server manager's password to connect.
security.authentication.type	simple	Directory server authentication type to connect. Use none for anonymous login if your directory allows it.
handle.to.match.in.directory.server	(sAMAccountName={0})	Handle of directory server with which entered username will be matched. For ex. (cn={0}) will try to match with cn.
user.first.name.mapping	givenName	First name attribute name in directory.
user.last.name.mapping	sn	Last name attribute name in directory.
user.email.mapping	mail	Email address attribute name in directory.
company.name.mapping	company	Company name attribute name in directory.
country.mapping	c	Country attribute name in directory.
state.mapping	st	State attribute name in directory.
user.city.mapping	l	City attribute name in directory.
street.address.1.mapping	streetAddress	Street address 1 name in directory.
street.address.2.mapping		Street Address 2 attribute name in directory.

user.postal.code.mapping	postalCode	Postal code attribute name in directory.
user.phone.number.mapping	telephoneNumber	Name of attribute users phone number in directory server
user.ldap.password.mapping	unicodePwd	Name of attribute users password in directory server
channel.class.type	top;organizationalUnit	Name of class of object to be created as channel in directory server. For ex. top;organizationalUnit or organizationalUnit
account.class.type	top;organizationalUnit	Name of class of object to be created as account in directory server. For ex. top;organizationalUnit or organizationalUnit
user.class.type	top;user	Name of class of object to be created as user in directory server. For ex. top;person or user
channel.dn.identifier	ou	Identifier of channel in distinguished name of this object. for ex. ou
account.dn.identifier	ou	Identifier of account in distinguished name of this object. for ex. ou
user.dn.identifier	cn	Identifier of user in distinguished name of this object. for ex. cn
user.additional.attributes		Any additional attributes in directory server which you want to set at the time of user creation. For ex. country=India
user.username.mapping	sAMAccountName	Name of attribute users username in directory server. For ex. cn or sAMAccountName etc.
is.active.directory	true	"true" if directory service is Active Directory, else "false"
is.ssl.on	true	"true" if using secured mode(ssl), else "false"
user.enablement.attributes	userAccountControl=512	Any attribute needs to be set on directory server object while enabling it. for ex: userAccountControl=512

User.additional.attributes is an optional field. If you want any key to have any particular value on directory server when user is getting created you can mention here.

User.enablement.attributes is an optional field. If you want any key to have any particular value on directory server when user is verifying himself from the activation link, you can mention here.

Make sure is.ssl.on is true if you are using ldaps protocol.

Enabling Anonymous Browsing

As a service provider, you can enable anonymous browsing so that potential customers can view the product catalog and understand the products and product bundles that are being offered.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click Administration.
3. In the Configuration pane, click Server.
4. In the Anonymous Browsing area, click Configure.
5. Click Edit and set the value of catalog.display to true.
6. Click Save and then, click Close.

Configuring Trial Management Settings

As a service provider, you can configure the settings to handle trial accounts.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. Click Configuration > Trial Management.
4. Click Configure > Edit.
5. Set the following parameters:

Parameter Name	Description
first.warning	Send first warning email to customer these many days before expiration of trial period.
second.warning	Send second warning email to customer these many days before expiration of trial period.
final.warning	Send last warning email to customer these many days before expiration of trial period.
trial.termination	Send Trial Termination email to customer these many days after expiration of trial period.
termination.threshold	Send email to customer these many days after a trial account expires, if it is not converted to a paying account.

6. Click **Save** and then, click **Close**.

Configuring Google Analytics settings

You can integrate CloudPortal Business Manager with Google Analytics and track your web traffic.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Portal > Integration.
4. Click **Configure** and then, click **Edit**.
5. Set the value of the parameter integrations.analytics to True to enable the google analytics.
6. Click **Save** and then, click **Close**.
7. Click Configuration > B/OSS Integration.
8. Click Google Analytics > Configure.
9. Click Edit and set the following parameters:

Parameter Name	Description
analytics.account	Google analytics account name.
analytics.domain	Domain name for Google analytics.

10. Click **Save** and then, click **Close**.

Configuring geo restriction

You can grant or deny access to customers based on their geographic location. CloudPortal Business Manager comes integrated with a geolocation application that is used to arrive at the geographic location of users. The country is identified from the IP address of the user. You can provide a list of countries that are either blacklisted or whitelisted. You can specify ISO 3166-1 alpha-2 codes for countries and either allow or deny users from these countries to sign up. These lists are used for users of all account types. You can specify the lists from the Account Management pane of the CloudPortal Business Manager UI.

The settings are enforced during the following events:

- When a user attempts to sign up from the Sign Up page.
- When a Root User attempts to sign up a user.

If a user tries to access the portal from a country that you have marked as blacklisted, the geolocation application rejects the login attempt and directs the user to the CloudPortal support page. A country whitelist always takes precedence over a country blacklist. For example, if a country is marked in the whitelist and also in the blacklist, the country is treated as whitelisted.

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. Click **Configuration > Account Management > Onboarding**.
4. Click **Configure**, and then click **Edit**.
5. In the Value box for the parameter `country.whitelist`, enter the ISO 3166-1 alpha-2 country codes separated by commas. This is the whitelist of countries from which users are allowed to sign in. For more information on 3166-1 alpha-2 ISO country codes, go to http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2.
6. In the Value box for the parameter `country.blacklist`, enter ISO 3166-1 alpha-2 country codes separated by commas. This is the blacklist of countries from which users are not allowed to sign in.
7. Click **Save** to save your changes and then click **Close** to close the dialog box.

Understanding B/OSS subsystems

Operational Support Systems are the systems used by service providers. It deals with the supporting processes such as maintaining network inventory, provisioning services, configuring network components, and managing faults.

Business support system refers to systems dealing with customers, supporting processes such as taking orders, processing bills, and collecting payments.

The two systems together are often abbreviated as BSS/OSS or B/OSS.

B/OSS contains processes and systems that support back-office activities within a network or customer facing activities.

This section enable you to understand some of the basic B/OSS configuration tasks in detail.

Configuring Fraud Control
Configuring Payment Gateway
Configuring Salesforce
Configuring Phone Verification

Configuring Fraud Control Settings

CloudPortal Business Manager comes integrated with a fraud control provision that can assess fraud risk associated with the user's credit card. You can determine the admissible fraud risk level for credit card users by entering values for certain parameters as shown in the table below.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Go to Administration > Configuration > Portal > Integration.
3. Click Configure and set the following parameter values.

Parameter Name	Parameter Value	Description
creditCard.fraudDetection	True/False	Set this value to True if you want to enable the credit card fraud check service. If you set this value to False, credit cards will not be put through the fraud check mechanism and will be accepted as is.
creditcardfrauddetection.threshold	Any numerical value	This is the minimum fraud threshold that you set in numerical units. The value that you set here determines the extent of fraud risk that you are willing to accept. A lower threshold value would mean that your 'trust' level is higher than that for a higher threshold value. The recommended default value for this parameter is 3.

Configuring Fraud Control Settings

creditcardfrauddetection. emailthreshold	Any numerical value greater than <integration. fraudcontrol.threshold>	Enter a value greater than <creditcardfrauddetection .threshold>. For any risk score that lies between <c reditcardfrauddetection.e mailthreshold> and <credi tcardfrauddetection.thres hold>, an Email alert for credit card fraud check violation is sent to the admin support. The recommended default value for this parameter is 25.
---	--	--

4. Click **Save** and then, click **Close**.

Configuring Salesforce.com Service Cloud Integration

CloudPortal™ Business Manager is integrated with Salesforce.com Service Cloud. This is used to log requests for support (trouble tickets) from customers. The connection between Salesforce and CloudPortal Business Manager is set up using configuration parameters as shown below.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Home > All Services.
3. Click B/OSS.
4. Select **Salesforce**. Click Configure.
5. In the Configuration window, enter the following details:
 - **Username** - Specify the username for SOAP webservice requests.
 - **Endpoint URL** - Specify the endpoint URL for SOAP webservice requests.
Note: Ensure that you provide the Salesforce API version "27" in the endpoint URL.
 - **Password** -Specify the password for SOAP webservice requests.
 - **Salesforce.com token** - A security token automatically-generated key from Salesforce.
 - **Timeout** - Set a value less than the timeout value set within Salesforce.
 - **More information required** - If set to true, it allows you to save the additional information like Tenant/user address, telephone number to salesforce.com.
6. Click Save.
7. Enable Salesforce by choosing "On" toggle button and then, click OK. An information window appears with the message - "Instance Updated successfully."

When you change the configuration settings, you have to restart the CloudPortal service.

Configuring Payment Gateway

You can integrate CloudPortal™ Business Manager with CyberSource or Authorize.Net payment gateway system.

Note: You have to sign up for a Authorize.Net or CyberSource account before you can use these payment gateways for processing your business transactions. During your payment gateway account activation and configuration, you have to specify the currencies that the payment gateway will support. Because the Authorize.Net payment gateway supports only one currency, ensure that only that currency is enabled in CloudPortal Business Manager when you use Authorize.Net. If you have multiple currencies enabled in CPBM, ensure that you use the CyberSource payment gateway.

Perform the following steps to configure the payment gateway.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Home > All Services.
3. Click B/OSS.
4. Select Payment Gateways on the left pane. CloudPortal Business Manager supports:
 - **CyberSource** payment gateway.
 - **Authorize.Net** payment gateway.
5. Select the desired payment gateway. Click Configure.

Note:

- Only one payment gateway can be enabled.
 - The payment gateway cannot be changed after an account is created.
6. Accept the Terms and Conditions to proceed.
 7. In the Configuration window, enter the following details:
 - **Server URL / Authorize.Netwsdl:** Specify the URL of the payment gateway's endpoint.
 - **Merchant name:** Specify your unique merchant name for accessing the payment gateway.
 - **Transaction Key:** Specify your unique API transaction key for accessing the payment gateway.
 8. Click Save.
 9. Enable the payment gateway by choosing "On" toggle button. An information window appears with the message - "**Instance Updated successfully.**"

Note: Once a payment gateway is enabled, you should not disable or deactivate it.

Configuring TeleSign Phone Verification

CloudPortal™ Business Manager is integrated with TeleSign service cloud for the phone verification. This is used to verify the identity of telephone users in real-time.

To configure this service:

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Home > All Services.
3. Click B/OSS.
4. Select the TeleSign. Click Configure.
5. In the Configuration window, enter the following details:
 - **Customer ID:** Specify the TeleSign customer ID.
 - **Authentication ID:** Specify the TeleSign authentication ID.
 - **Language:** Specify the language in which the user gets the verification call.
 - **Allowed Phone Types:** Specify the supported phone devices. For more information, refer to [Phone Type Codes](#).
 - **Help Desk Phone:** Specify the number of the help desk.
 - **Help Desk Email:** Specify the mail ID of the help desk.
6. Click Save.
7. Enable TeleSign by choosing "On" toggle button and then, click OK. An information window appears with the message - "**Instance Added successfully.**"

Note: When "Phone Verification" is

- **Enabled:** The user cannot edit the ISD field and the ISD number is derived from the selected country. This is because the ISD code must be the code of the country of address of the account
- **Disabled:** The user can edit the ISD field because the phone number is not verified

Phone Type Codes

The following is a list of phone type codes <TypeofPhone> returned by the TeleSign API, along with their descriptions.

Code	Description	Risk Level	Recommended Action	Detail
300	Undetermined	Medium	Flag	This is either an invalid phone number or a very new number. Because of its newness, TeleSign is not yet able to determine the phone type. This could potentially be a VOIP phone, so there is some risk involved with this type of phone number.
301	Fixed Line	Low	Allow	Fixed-line phones include traditional landline phones and VOIP phones that can be traced back to a particular address and cannot be obtained by a user outside the phone number's designated country. Because these phone numbers are traceable, they are low-risk.
302	Mobile	Medium-Low	Allow	<p>Mobile telephones are provided by companies such as Verizon, Cingular, or Sprint. Users of these phones must sign contracts, making these telephone numbers traceable. These phone numbers are generally low-risk. However, some prepaid mobile phones will be identified as Mobile.</p> <p>Internationally, phones identified as mobile may also include TETRA mobile phones, cordless access systems, proprietary fixed radio access, and fixed cellular systems.</p>

Phone Type Codes

303	Pre-paid Mobile	Medium-High	Flag	<p>These telephones can be purchased anonymously at many department stores. They work like other mobile phones, but their users are not contracted with a mobile phone company. Prepaid mobile phones generally cost between \$30 and \$100, making them unattractive for most fraudsters, but we still recommend flagging transactions made with this type of telephone due to a user's anonymity. Also, not all prepaid mobile phones will be identified as prepaid; some will be identified as Mobile.</p>
304	Toll-Free	High	Block	<p>Toll-Free numbers are numbers starting with toll-free area codes, such as (800) numbers. Although sometimes used by legitimate users calling from companies, fraudsters can easily obtain toll-free numbers that either forward to their telephone which is located in another country or toll-free numbers that terminate at an Internet voicemail box. Toll-free numbers always forward to non-toll-free numbers. For this reason, any legitimate user who provides a toll-free number and is asked to provide a non-toll-free number will be able to do so. We therefore recommend blocking users of tollfree numbers.</p>
305	Non-Fixed VoIP	High	Block	<p>Voice Over Internet Protocol (VOIP) phone numbers are Internet-based telephone numbers. Non-fixed VOIP telephone numbers can easily be obtained by users who are not located in the country associated with the telephone number. They are untraceable and disposable; some can even be obtained for free. This means that a fraudster in Romania could easily obtain a US-based telephone number using a non-fixed VOIP service and receive a call to this number in his/her home in Romania.</p>

Phone Type Codes

306	Pager	High	Block	This phone number rings to a pager. Because pagers cannot receive verification calls, it's best to block calls to these numbers. These calls are most likely requested by fraudsters who are inputting invalid phone numbers.
307	Payphone	High	Block	Calls to a payphone cannot be traced back to any one person; therefore, we recommend blocking verification calls to these numbers.
308	Invalid	High	Block	The telephone number entered is not a valid number or is a number originating on a Caribbean island (i.e. Jamaica, St. Lucia, etc.)
309	Restricted Number	High	Block	<p>The telephone number entered is a restricted number. In the US, restricted numbers include numbers that begin with 0 and numbers with the following area codes:</p> <ul style="list-style-type: none"> • 900 • 911 • 411 <p>Internationally, the following phone numbers are restricted:</p> <ul style="list-style-type: none"> • Audiotext • Carrier selection • National rate • Premium rate • Shared cost • Satellite • Short Code

Phone Type Codes

310	Personal	Medium-Low	Allow	<p>A “Personal” phone number is a phone number which allows a Subscriber to receive calls at almost any telephone number, including a mobile number. The “Personal” number forwards to another phone number, which is determined by the subscriber. Because the “Personal” number can forward to either a landline or mobile number, it’s impossible to determine what type of phone the subscriber is using. However, because “Personal” phone numbers must be obtained from a phone company, there is little risk of fraud associated with this phone type.</p>
311	Voicemail	Medium-High	Block	<p>Voicemail phone numbers will ring directly to a user’s voicemail. Although this phone number was likely purchased through a phone company and may be traceable, it will be impossible to reach the user live at this phone number. In addition, any user providing a Voicemail number should also have a valid landline or mobile phone available. Therefore, we recommend blocking this type of phone.</p>
320	Other	Medium-High	Block	<p>Phone numbers labeled “Other” are all various types of phones that originate in non-US countries. The following types of phones may be included under the “Other” category:</p> <ul style="list-style-type: none"> • Global title address • Inbound routing code • Videtext • VPN <p>Because clients should not be requesting calls to these phone types, we recommend blocking verification calls to these numbers. Any phone number entered that falls under the “Other” category was probably either entered incorrectly or was entered by a fraudster testing various phone numbers.</p>

Adding a new service to CAS

As a service provider, if any application needs to be authenticated using CAS, you can add a new service to CAS.

After enabling CAS Authentication, follow the steps given below to add new service to CAS.

1. Log in as root to the host running CloudPortal Business Manager.
2. Edit the `/usr/share/vts3/repository/prop/deployerConfigContext.xml` file as follows:
 - a. Search for the bean "serviceRegistryDao" in the xml file.
 - b. Add a new bean to the property "registeredServices" of the above bean.
 - c. The template is as follows:

```
<bean class="org.jasig.cas.services.RegisteredServiceImpl">
  <property name="id" value="<Unique id>" />
  <property name="name" value="<Name of the application>" />
  <property name="description" value="<description of the application>" />
  <property name="serviceId" value="<Callback URL used by CAS to redirect back to the new applica
</bean>
```

3. Restart the cloud-portal service.

Note: Property Id should be unique in the list of registered services only. And you need separate service for both http and https protocols.

Customizing the Service Provider Logo and Favicon

As a service provider, you can have your custom logo displayed in CloudPortal Business Manager. For the custom logo to appear, follow these steps:

1. Log in to CloudPortal as the Root User.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Settings > Configure.
5. Set the parameter `images.uploadPath` to specify the absolute path to the images directory for storing custom image logos, favicon, and so on. Set the value to a valid existing path in your server file system.

The path to upload logo on the server should have valid write permissions.

6. Go to **Customize Logo** (in the drop-down menu on the top right corner).
7. Browse to Company Logo.
8. Browse to Favicon.
9. Click **Submit**.

Custom Logo in Email Templates

For the custom logo to appear in Emails, follow these steps:

1. Log in to CloudPortal as the Root User.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Configuration > Portal.
4. Click Settings > Configure.
5. Set the parameter `image.path` to specify the path to the images that are used in automatically generated emails. Specify the absolute path of the image.
6. Restart the CloudPortal service. To restart the CloudPortal service, enter service restart cloud-portal on the VM console.

Creating Service Provider Users

Service provider users are users that are part of the organization operating the cloud. These are members of your organization who will manage various aspects of supporting the users of the cloud.

Note: All these users will have access to sensitive information across all accounts in your system and therefore should only be provided access if they have the appropriate privileges in your organization. These users cannot use the resources of your cloud services, but only operate and help your consumers with their use of the cloud.

As a Root User of your setup, you can add Service Provider users to your own account.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Go to Administration > Service Provider Users. List of all the users in your account will be displayed in the left pane.
3. Click Add New.
4. Enter the user name and email id of the user you want to create.
5. In the Profile field, select the profile name that corresponds to the new user's role in the organization. Each profile has a set of permissions associated with it.

Note: The selection of a profile is very important, because it determines what the user will and will not be able to do on the portal.

6. Select the Time Zone and Preferred Language fields.
7. Click Customize Welcome Email to change the text of the notification mail that will be sent to the new user.
8. Click Finish. The user will receive a Welcome Email that contains a link to verify his email and to set the password. The service provider user must log in through this link for the first time to confirm the validity of the email address.

Note:

- There is no expiry period for the verification link sent in the Welcome email.
- Also, the expiry period for the verification link cannot be configured or changed.

Managing Content Templates

CloudPortal™ Business Manager sends a variety of automatically generated emails to internal personnel and customers.

You can modify the content of the emails by editing the email templates in the portal. These templates support variable data substitution, and can be used to send out personalized emails. To modify the content of these emails:

1. Log in to CloudPortal Business Manager as the Root User or Ops Admin.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Content Templates > Emails.
4. Select the template you want to edit.
5. Place the cursor over the gear wheel icon and click Edit.
6. Modify the content in the editor.
7. Click **OK**.

You can click **Preview** to view the edited template.

You can also modify the font styles, header, footer, and signature in your emails. Select the respective tabs, and modify them as explained above.

To verify the changes, click the Send test mail icon at the top right corner of the screen.

Email Templates and Notification Triggers

CloudPortal Business Manager sends out email alerts and notifications to users in certain instances. Given below is a list of email alerts sent, when they are triggered, and who they are sent to.

Template name	When is the email sent	Who is it sent to
Welcome Email	On sign up	Customer Master User
Verify Email	When users sign up or new users are added	Any user
Verify Alert Email	For verification of email for Alerts	Any user
Reset Password	On request for resetting password	Any user
Trial service expiry warning	When Trial accounts are nearing expiry	Master user
Trial service expired	When Trial account expires	Master user
Trial service expired notification	For expired trial accounts	Sales alias
Credit Limit Crossed Service Suspended Notification	For service suspension caused by credit limit breach	Sales alias
Spend Budget Warning	When the spend budget threshold is reached	Any user
Credit Limit Credit Notify	For credit limit alert	Billing update alias
Credit Limit Breach Alert	On breaching the credit limit and if the Breach Action is in Alert / Alert internal	Sales alias
Credit Limit Crossed Service Suspended	When the final credit limit is breached and the service is suspended.	Master user
Payment Received	On receipt of payment	Master user
Payment Failed	On payment failure	Master user
Payment Failed Notify	On breach of payment failure alert threshold	Billing update alias
Payment info change requested	On payment information change request	Sales support and Finance alias
Deposit Received	On receipt of deposit	Master user
Support Ticket Notify	On creation of ticket	Help desk alias

Email Templates and Notification Triggers

Corporate Account Request	On new corporate account request	Sales support and Finance alias
Credit Exposure Marker Level	When credit exposure marker is crossed	Sales alias
Send Promotional Token	Invoked by customization done by service provider	Applicant of Trial account
Account Restrict	On account restriction	Master user
Account Activate	On account activation	Master user
Account Suspend	On account suspension	Master user
Account Terminate	On account termination	Master user
Account_reactivate	On account reactivation	Master user
Account Converted	On account conversion	Finance alias and Master user
Account Converted Admin	After account conversion	Finance alias
Pending Change Rejected	When pending changes are rejected	Master user
VM Creation	On VM creation	Any user
Report	On send report request	Customer user
Job Failed	On failure of any scheduled job	admin_support alias
Destroy VM Failed	When destroy virtual machine command fails	vmops_error_notification_to alias
Orphaned VM Created	When orphaned virtual machines are created	vmops_error_notification_to alias
Credit card Fraud Check Event	When MaxMind credit card fraud check is enabled and credit card transaction fails	admin_support_email alias
Account Activation Request	When manual activation is required	approval_alias
Device Fraud Detection Event	When ThreatMetrix fraud check is enabled and transaction is deemed unsafe	admin_support_email
Credit Limit Warning	When the credit limit threshold is reached	Customer user
Invoice Email	When invoice is generated	Customer user
User Deactivate Email	When Master user deactivates a user in his account	Customer user
User Activate Email	When Master user activates a user in his account	Customer user
Internal Approval	Sent by workflow activity, when a task is created from the workflow that requires an action to be taken	Recipient email address is configurable

Email Templates and Notification Triggers

Tenant Admin Approval	When admin approves the user account creation request	Customer user
New Resource Type	When new resource type is detected in system	Owner of the resource and Master user
Incompatible Resource Type	When the resource type is not compatible with existing subscription	Owner of the resource and Master user
Missing Product	When there is a missing product for usage data in system	admin_support
Provision Resource Failed	When resource provisioning fails	user
Subscription Activated Admin	On activation of subscription	product bundle owner/sales support list
Subscription terminated Admin	On termination of subscription	product bundle owner/sales support list
Subscription Activated	On activation of subscription	Customer user
Subscription terminated	On termination of subscription	Customer user
Job Warning	When any scheduled job status is "warning"	admin_support alias
VM Password Reset	On resetting the password for the Virtual Machine	User and Master user

Sending test mails

CloudPortal Business Manager sends a variety of automatically generated Emails to internal personnel and customers. You can send test mails to check the recipient addresses and the content. You can also modify the font styles, header, footer, and signature in your emails. To verify the changes, you can use the send test mail functionality.

1. Log in to CloudPortal Business Manager as the Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Content Templates > Emails.
4. Select the email template. If required, modify the template as needed.
5. Place the cursor over the gear wheel icon and click **Send test mail**.
6. Click **OK**.

Modifying the Header, Footer, and Signature in content templates

As a service provider, you can modify and format the header, footer, and signature that appear in the content templates.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Content Templates > Includes.
4. Select the language from the dropdown list.
5. To modify the header, click Email Header.
6. Place your cursor over the manage icon (that looks like a gear wheel) and click Edit.
7. Make the changes as required and click **OK**. You can click Preview to view the modified header. You can modify the footer and signature by selecting the respective option and following Steps 4 - 6.

Modifying Invoice Templates

You can modify the invoice templates to suit your needs.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Content Templates > Invoices.
4. Select the language from the dropdown list.
5. Select the invoice template you want to edit.
6. Place your cursor over the manage icon (that looks like a gear wheel) and click Edit.
7. Make the changes as required and click **OK**. You can click Preview to view the edited template.

Objects and Properties in Invoice PDF template

The following table provides a list of available objects and properties that can be referenced in the Invoice PDF template. The properties can be directly used. For example, `#{dateFormat}` for the property `dateFormat`. The objects in the template are used to get the property values within them for the template. For example, (`$ tenant.name`): to get the name of the tenant, use the object `tenant`.

Template name	Object	Properties
INVOICE_PDF_TEMPLATE	AccountStatment : : accountStatment	dateFormat
Parent Object		
AccountStatment	Tenant :: tenant SalesLedgerCreditRecords :: salesLedgerCreditRecords Invoice :: subscriptionInvoices Invoice :: renewalInvoices	billingPeriodStartDate billingPeriodEndDate balanceForwardAmount credits newCharges (subscriptionCharges + utilityCharges + taxes - discounts) finalCharges subscriptionCharges utilityCharges discounts taxes

The following table provides details of the properties/child objects:

Object	Child Objects	Properties
SalesLedgerCreditRecords		transactionAmount createdAt

Objects and Properties in Invoice PDF template

Invoice		<p>serviceStartDate</p> <p>serviceEndDate</p> <p>rawAmount</p> <p>discountAmount</p> <p>subTotal</p> <p>taxAmount</p> <p>amount</p>
Tenant	<p>Channel :: sourceChannel</p> <p>AccountType :: accountType</p> <p>User :: owner</p> <p>Address :: address</p> <p>Address :: secondaryAddress</p> <p>CurrencyValue :: currency</p> <p>TrialAccount :: trialAccount</p> <p>DepositRecord :: deposit</p>	<p>name</p> <p>spendLimit</p> <p>prepaidBalance</p> <p>initialDeposit</p> <p>spendBudget</p> <p>syncBillingAddress</p> <p>state</p> <p>memo</p> <p>paymentAlertsCounter</p> <p>accountSuspensionCounter</p> <p>usageQuery</p> <p>remoteAddress</p> <p>isManualActivation</p>
Address		<p>street1</p> <p>street2</p> <p>city</p> <p>state</p> <p>postalCode</p> <p>country</p>

AccountType		name displayName description paymentModes trial paymentInfoRequired autoPayRequired instantpayRequired notionalBilling manualActivation manualRegistrationAllowed defaultSelfRegistered defaultRegistered maxUsers accountRestrictionGracePeriod preAuthRequired depositRequired
CurrencyValue		currencyCode currencyName rank longCurrencyName sign altSign active conversionFactor

Modifying Content Template Styles

The content templates have styles defined to help you maintain uniformity in the emails. You can modify the styles to suit your needs. You can also modify the custom information such as support email, support phone, and so on. These changes get reflected across all templates.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Content Templates > Styles.
4. Select the language from the dropdown list.
5. To modify the style defined, click Styles.
 - a. Click Edit and make the changes as required.
 - b. Click OK to save your changes.
6. To modify the custom information, click Custom.
 - a. Click Edit and make the changes as required.
 - b. Click OK to save your changes.

Modifying the Approval Templates

The approval templates provide the content for the different approval mails that are sent. The service provider can specify the content that is added to such approval mails, by modifying the template content.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon in the top right corner next to ROOT USER.
3. Click Administration > Content Templates > Approvals.
4. Select the language from the dropdown list.
5. Select the invoice template you want to edit.
6. Place your cursor over the manage icon (that looks like a gear wheel) and click Edit.
7. Make the changes as required and click **OK**.

Profile Management

A profile is a set of permissions that is saved under a name, such as Administrator or Manager. Every user of your portal is assigned a profile, including yourself. The profile is assigned when the user is added to the portal.

Profiles are useful for saving a frequently-used set of permissions so that they can be easily granted to a user in a single step. This is easier and less error-prone than setting each individual permission for every user.

Profiles also ensure that multiple users with equivalent duties and needs are all granted the same permissions. For example, you might want to grant the "Create Customer Account" permission for the Sales profile but not for the Help Desk profile. This section describes about roles and how you can create and manage profiles.

What are roles
Understanding CloudPortal Business Manager Profiles
Manage Profiles

What are roles

Roles are customary functions that are created to accomplish certain tasks. The following list is the default set of roles available in CloudPortal Business Manager.

Service Provider roles

- **Account create:** This role has the ability to create, read, update, and delete any customer account.
- **Account manage:** This role has the ability to view and update information on any customer account.
- **User:** This role is the default role for any registered user.
- **User create manage:** This role has the ability to create, read, update, and delete a user belonging to any customer account. This role depends on "account create" or "account manage" role and on "config manage" to create/manage service provider users.
- **Finance create manage:** This role has the ability to create, read, update, and delete financial information of any customer account (Account). For example, make payment, issue credit. Depends on "account create" or "account manage" role.
- **Ops Comm:** This role has the ability to create, read, update, and delete status updates that are published through the Cloud Health page.
- **Profile create manage:** This role has the ability to create, read, update, and delete profiles on the CloudPortal.
- **Config manage:** This role has the ability to view and update global configuration properties. Depends on "user create manage".

Note: This role is available only for the Root user profile.

- **Product create manage:** This role has the ability to create, read, update, and delete any product.
- **Tickets Manage:** This role has the ability to create, read, update, comment on, and close support tickets across all customer (tenant) Accounts.
- **Reports Admin view/download/email:** This role has the ability to view/generate reports.

Customer Roles

- **Account User create manage:** This role has the ability to create, read, update, and delete a user in customer account.
- **User:** This role is the default role for any registered user.
- **Billing Admin manage:** This role has the ability to manage tenant billing.

What are roles

- **Account manage:** This role has the ability to view and update information on his/her own customer account.
- **User Tickets Manage:** This role has the ability to create support tickets within his/her own user account (User) and be able to read, update and comment on his/her own support tickets.
- **Account Tickets View:** This role has the ability to create, read and comment on support tickets raised by all users within his/her own customer account.
- **Account Tickets Manage:** This role has the ability to create, read, comment on, and close support tickets raised by all users within his/her own customer account.

A profile is defined using a combination of one or more of these roles. A root user is enabled with all these roles by default. You can also customize the profiles by defining their roles.

Understanding CloudPortal Business Manager Profiles

CloudPortal™ Business Manager ships with a predefined set of profiles. You cannot add or delete a profile. If you have the "Create/Manage Profile" permission, you can select which permissions to grant under each profile name. Given below are the profiles in CloudPortal Business Manager.

Service Provider Profiles

- Root
- Ops Admin
- Finance Admin
- Sales Support
- Help Desk
- Product Manager

Customer Profiles

- Master User
- Billing Admin
- User
- Power User

Managing Profiles

As Service Provider root user, you can manage your profile and also other users' profiles.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Go to Administration > Profiles and select Service Provider or Customer tab, based on the profile that needs to be updated.

You can see the default list of profiles in the left pane. The description of the available roles for each profile are visible in the right pane.

3. Select the checkboxes under **Allow** for the roles that you would like to set for the profile.
4. Click **Save**.

Note: Before changing profile permissions, make sure you have discussed it with the relevant decision makers within your organization.

Cloud Service

A cloud service is a resource that is provided over the Internet. The most common cloud service resources are Infrastructure as a Service, Platform as a Service, and Software as a Service.

A cloud service can dynamically scale to meet the needs of its users, because the service provider supplies the necessary resources for the service, there is no need for a company to deploy its own resources to manage the service.

CloudPortal™ Business Manager uses bootstrapping process to provision machines for managing the services required to deploy applications.

Onboarding and Managing a service

Onboarding a service refers to the process of:

- Deploying a cloud service connector into CloudPortal Business Manager.
- Enabling a cloud service connector and adding a cloud service instance.

Cloud Service Connector

A connector is a Java based OSGi bundle which:

- Implements the semantics prescribed by the Cloud Service Connector SDK.
- Defines the following metadata:
 - Endpoint and credential parameters
 - Account Configuration
 - Account Controls
 - Resource Metadata: Various resources that can be provisioned and metered
 - Custom UI Views for Account configuration, Subscription provisioning

The following connectors are supported:

- Citrix CloudPlatform
- **Experimental connectors:**
 - CloudPortal Services Manager
 - Windows Desktop
- **Reference connectors:**
 - Apache CloudStack
 - OpenStack

The connector implements set of hooks and CloudPortal Business Manager invokes these hooks at appropriate stages. The stages include:

1. Account Creation
2. User Creation
3. Resource provisioning
4. Product definition

5. Usage collection
6. Event collection

Deploying a cloud service connector into CloudPortal Business Manager

The cloud service connector will be deployed into VIRGO's pickup directory. If there are no errors in the connector, then, CloudPortal Business Manager extracts the information from the connector and copies the serviceDefinition.xml, logo, terms and condition files, and UI implementation files to /usr/share/cloud/portal/data directory.

Now you can enable the cloud service connector from the UI.

Enabling cloud service connector and adding an instance

To enable the cloud service connector:

1. Log in to CloudPortal Business Manager UI as Root user.
2. Go to Home > All Services.
3. Click Enable.
4. Accept the Terms & Conditions defined by the Cloud Service.
5. After enabling the cloud service, roles defined by the cloud service are mapped to appropriate CloudPortal Business Manager profiles.

Note: The service onboarding for customer and customer's users depends on these role mapping. Ensure that roles are appropriately mapped to profiles.

6. Click **Submit** and then click **Close**.

Adding an instance to the service

1. Go to Home > All Services > Cloud Services.
2. Select the Service and click Add.

The Add Service Instance wizard opens.

3. Specify the identifying information for the instance such as name, code, and description and click Next.
4. In the Configuration page, specify the details that are specific to the connector for the service.
5. Click Next.

The Default Product Selections page is displayed. This is a list of products based on defined UsageTypes.

6. Select the products that you want to create at the time of creation of the service instance.

You can also edit the name and code for each of the default products.

7. Click Next and specify the utility charges for the products that you have selected.
8. Review and confirm the details.
9. Click Add and then, click Close.

Editing an instance

1. Go to Home > All Services > Cloud Services.
2. Select the cloud service and click the cloud icon to view all instances of the selected service.
3. Click the gear wheel icon next to the instance and select Edit Configuration.
4. Make changes as required and click Save.

Workflow Management

In CloudPortal™ Business Manager, user can perform certain actions to achieve a desired business purpose. These actions can be considered as business transactions that help in achieving this business purpose. A workflow is a multi-stage process that consists of multiple phases. Every phase consists of a set of activities, (smallest unit of work in a workflow) that should be completed successfully in order to complete the phase. All phases should be completed to complete a business transaction.

A business transaction contains certain details. CloudPortal Business Manager checks these details and decides which workflow is suitable to complete the business transaction. A workflow is considered successful only if it satisfies the rule that all the activities in the configured buckets are successful. When the workflow is complete, the business transaction would also be completed successfully.

All the phases are executed in a sequential order. A phase consists of activities that are customer driven, service provider driven, and system driven.

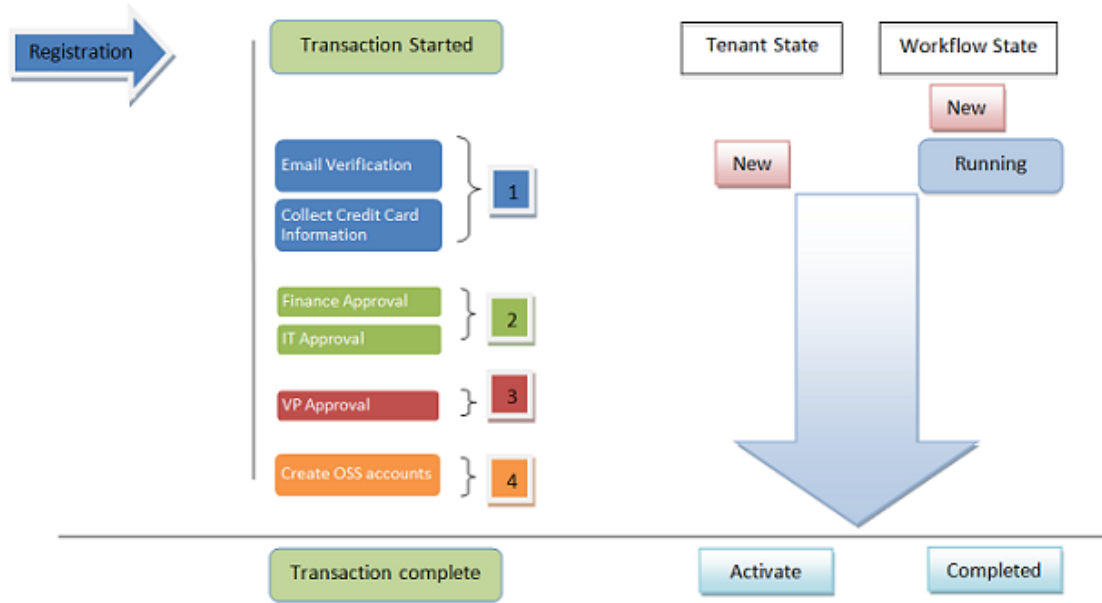
The activities in a given phase occurs in parallel. System-driven activities occur automatically. However, others require manual input for the workflow to complete. Activities have built-in capabilities to retry. You can configure the number of times an activity can retry. The activity is run for a specified number of times and gets completed only if the conditions are satisfied.

A task is created by an activity when it requires some human intervention. Each task has a role associated to it and any user with that role can act on the task if it still pending. Following are the instances in which a task is created:

- InternalApprovalActivity- Creates a task for the users with role mentioned in the property to approve or reject the business transaction.
- CollectPaymentInfoActivity - Creates a task for user having the ROLE_ACCOUNT_BILLING_ADMIN role requesting for credit card information.
- TenantOwnerApprovalActivity - Creates a task for users having the ROLE_ACCOUNT_ADMIN role to approve or reject the business transaction.

In the out-of-the-box configuration, there are pre-defined workflows. In general, activities can be customized. It is also possible to create custom activity types.

Consider an example of changing the state of an account from new to active. If this state conversion has a mandatory approval activity to be performed by an administrator, CloudPortal Business Manager generates a task for the person responsible to approve and sends a mail requesting them to approve. The related activity ensures that the task is completed by indicating that there are pending tasks for the administrator. Once the administrator checks the pending tasks and approves or rejects based on the business transaction details, the workflow proceeds to the next phase or gets completed if it reaches the last phase.



As shown in the figure below, let us consider a business transaction is raised to change the tenant state from New to Active. CloudPortal Business Manager will check for the suitable workflow which is mapped in transactionWorkflowmap.xml. This file defines the requirement to match the workflow with the business transaction and the suitable workflow is triggered. You can find the details of the workflow defined in workflows.xml file. In the workflow, there will be various activities running in parallel in various phases (represented as 1, 2, 3 and 4 in the figure). Figure 1. Example of a workflow for Tenant State change from NEW to ACTIVE

How workflows work

In CloudPortal™ Business Manager, the following types of business transactions initiate a workflow:

- Tenant activation: This transaction is triggered when a new tenant is created.
- Account type conversion: This transaction is triggered when a tenant requests conversion of his account type.
- Subscription activation: This transaction is triggered when a new subscription is created by a tenant.
- Payment information change transaction : This transaction is triggered when an account changes its payment information .

These transactions are triggered from CloudPortal™ Business Manager and may or may not initiate a workflow.

The rules or conditions to initiate a workflow depending on transaction details are defined in transactionWorkflowMap.xml.

```
<TransactionWorkflowMap>
  <Match type="paymentInfoChange" workflow="corporate-payment-info-change">
    <Rule>{#transaction.tenant.accountType.name == 'Corporate'}</Rule>
  </Match>
</TransactionWorkflowMap>
```

Important properties in this file -

- **Type:** This is an attribute to element "Match" and defines the type of business transaction on which the rule needs to be applied. Possible values are:
 - tenantStateChange Transaction
 - tenantAccountTypeConversion Transaction
 - subscriptionActivation Transaction
 - paymentInfoChange Transaction
- **Workflow:** This is an attribute to element "Match" and it is the name of the workflow that will be initiated when the defined rule is matched for the transaction. All the workflows are defined in workflows.xml.
- **Rule:** This is a sub-element under "Match" and defines the rule to be applied on the transaction.

If multiple rules are matched then first "Match" will be successful and workflow associated with this "Match" will be returned. If there is no matched workflow, then the transaction will be completed.

Workflow name returned from transactionWorkflowMap.xml is used to read the workflow definition from workflows.xml. This file has the complete definition for all the workflows, which include how many phases are there in a workflow, what all activities are there in each phase.

For example:

```
<Workflows>
  <Workflow name="account-activation-retail">
    <Bucket name="Phase-1" order="1">
      <Activity name="account-activation-retail.verifyMasterUserEmailActivity" ref="verifyMasterUserEmailActi
    </Bucket>
    <Bucket name="Phase-2" order="2">
      <Activity name="account-activation-retail.collectCreditCardInfoActivity" ref="collectCreditCardInfoActivi
      <Activity name="account-activation-retail.createOSSAccountActivity" ref="createOSSAccountActivity" />
    </Bucket>
  </Workflow>
</Workflows>
```

Important properties in this file -

- **Workflow:** This is an element which contains complete definition of a workflow.
- **Name :** This is a reference to the workflow name referenced in transactionWorkflowMap.xml.
- **Phase :** This is sub-element to "Workflow" element and contains a set of activities that need to be initiated in parallel.
- **Activity :** This is sub-element to "Phase" element and has an attribute "ref" which refers to the bean, that need to be invoked for this activity. the bean reference of the activity is defined in applicationContextworkflow-customizations.xml.

Workflow definition fetched from "workflows.xml" are used to instantiate the workflow into the database. When a workflow is initiated then it is in NEW state. Different states of a workflow are given below and are defined based on the status of activities in the workflow:

- **New** - Indicates that the workflow is yet to be initiated.
- **Running** - Indicates that there are pending activities in this workflow.
- **Completed** - Indicates that all the activities are successfully completed and the workflow was successful.
- **Failure** - Indicates that the workflow is partially complete with an error message, as one or more activities in the workflow could not be completed due to business logic failure. A business logic failure could be rejection of associated task or has reached the threshold of number of retries etc.
- **Error** - Indicates that the workflow is partially complete with an error message, as one or more activities in the workflow could not be completed due to some issue.

Note: If a workflow fails, you can restart the workflow. In that case, the workflow will reset the erroneous or failed activity.

Different states of an activity are given below:

- New - Activity is yet to begin.
- In-Progress - Activity is running.
- Waiting - User input is required.
- Success - Activity has been successfully completed.
- Failure - Activity was declined or rejected.
- Error - There has been some unexpected behavior or exception that needs to be fixed.

Workflows available in the CloudPortal Business Manager out of the box are as follows:

- Trial account activation
 - Phase one - Identity verification
 - Verify master user email activity
 - Phase two - Accounts creation
 - OSS Accounts creation activity
- Credit card account activation (For retail accounts)
 - Phase one - Identity verification
 - Verify Master User Email Activity
 - Phase two - Billing verification and account creation
 - Collect credit card information activity
 - OSS Accounts creation activity
- Manual account creation
 - Phase one - Identity verification
 - Verify Master User Email Activity
 - Phase two - Approval and account creation
 - Finance admin approval activity
 - OSS Accounts creation activity
- Account conversion to manual activation
 - Phase one
 - Finance admin approval activity
- Account conversion to credit card payment
 - Phase one
 - Verify credit card information activity
- Subscription activation with pre-authorization

How workflows work

- Phase one
 - Pre authorize the subscription
 - Payment information change for manual activation account
 - Phase one
 - Finance admin approval activity
- After the instantiation of a workflow, it is picked by workflow engine.

Viewing workflow status

Tenant scope workflow

For a Service Provider

A service provider user can view tenant scope workflows for any tenant by following the below steps:

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Accounts.
3. Click All Accounts.
4. Select an account on the left side pane. You will see the details of the account on the right side pane.
5. Click History tab on the right side pane. You can view the workflows for this account.
6. Click View under Workflow. You can see the status of the selected workflow in the Workflow Status window.

For a tenant user

The master user can view all tenant scope workflows by navigating to Company Setup >> History.

Note: An active tenant scope workflow is visible as a red bar under the primary menu.

Subscription activation workflow

For a service provider user

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Accounts.
3. Click All Accounts.
4. Select an account on the left side pane. You will see the details of the account on the right side pane.
5. Click the gear wheel icon on the right side pane.
6. Select View Billing History from the list. Usage and Billing dashboard is displayed.
7. Click Subscription. You can see Workflow option next to Status field.

Note: This option is visible only if a workflow exists for a subscription.

For a tenant user

Any tenant user can view subscription activation workflow by following below steps:

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click Home.
3. Select Usage & Billing>> Subscriptions.
4. Select any subscription. If there is any workflow created for the subscription, then you will see a View Workflow link in the Status field.

Available activities and their properties

All the activities that can be used in a workflow are defined in the `applicationContext-workflow-customizations.xml` file.

Available activities and their properties are listed in the table below:

Activities	Properties
VerifyMasterUserEmailActivity: To check whether master user's email has been verified.	<ul style="list-style-type: none"> <code>failureThresholdInHours</code>: Number of hours for which an activity will wait for a user to verify his email before activity errors out.
CollectCreditCardInfoActivity: Creates a pending task for the tenant to enter credit card information.	<ul style="list-style-type: none"> <code>failureThresholdInHours</code>: Number of hours for which the task will remain pending before activity errors out.
collectPaymentInfoActivity: Creates a pending task for the tenant to enter payment information.	<ul style="list-style-type: none"> <code>failureThresholdInHours</code>: Number of hours for which the task will remain pending before activity errors out.
CreateOSSAccountsActivity: Creates account on the OSS services enabled in the system.	<ul style="list-style-type: none"> <code>noOfRetries</code>: Maximum number of attempts to complete the activity. <code>timeBetweenRetriesInHours</code>: Number of hours a system will wait before retrying to complete the activity.
InternalApprovalActivity: Creates a pending task to approve or reject the business transaction for one of service provider user based on the role mentioned.	<ul style="list-style-type: none"> <code>taskType</code>: Type of approval to distinguish from other internal approval activities. <ul style="list-style-type: none"> <code>ui.task.type.<value of the task type>.name</code>: Name of the <code>i18n</code> property which defines the name to the task. <code>ui.widget.task.type.<value of the task type>.name</code>: This takes tenant name as an argument and name of the task appears along with the tenant name. <code>actorRole</code>: The pending task will be created for service provider users having this role. <code>toEmails</code>: The email address to which a transaction notification email will be sent. This can be CSV of EmailAlias or email id; combination is allowed <code>noOfRetries</code>: Maximum number of email notifications that will be sent. <code>timeBetweenRetriesInHours</code>: Interval in the time when the two notification emails were sent.
verifyDepositActivity: Checks whether a deposit has been made for a tenant.	<ul style="list-style-type: none"> <code>failureThresholdInHours</code>: Number of hours for which an activity will wait for the service provider to make a deposit for the tenant before the activity errors out.

Available activities and their properties

<p>SubscriptionPreAuthActivity: Preauthorizes subscription amount against the available credit card through payment gateway.</p>	<ul style="list-style-type: none"> • noOfRetries: Maximum number of attempts to complete the activity. • timeBetweenRetriesInHours: Number of hours a system will wait before retrying to complete the activity.
<p>TenantOwnerApprovalActivity: Creates a pending task for the master user to approve or reject a business transaction.</p> <p>Note: The task created for this activity gets auto approved if the creator of the subscription has the role " actorRole ", as defined in the bean.</p>	<ul style="list-style-type: none"> • taskType: Type of approval to distinguish from other internal approval activities <ul style="list-style-type: none"> • ui.task.type.<value of the task type>.name : Name of the i18n property which defines the name to the task. • ui.widget.task.type.<value of the task type>.name: This takes tenant name as an argument and name of the task appears along with the tenant name. • noOfRetries: Maximum number of email notifications that will be sent. • timeBetweenRetriesInHours: Number of hours a system will wait before retrying to complete the activity. • actorRole: Role which a user must have to view and act on the task.
<p>verifyCreditCardInfoActivity: Verifies whether credit card information has been added to the account.</p>	<ul style="list-style-type: none"> • failureThresholdInHours: Maximum amount of time in hours a system will wait before the activity errors out.

Managing your tasks

A user can view all tasks if the user has the required role for which the task is created.

Following task types are provided by CloudPortal™ Business Manager as default for POPUPs.

- FINANCE_APPROVAL
- TENANT_ADMIN_APPROVAL

To view a task, perform the following steps:

1. In the main menu, click All Tasks. You can view all the tasks on the left side pane
2. Select a task on the left side pane. You will see the details of the task on the right side pane.

You can filter the list of tasks to view all the tasks or to see if there are any pending or completed tasks.

Pending actions are also displayed on dashboard under My Tasks widget and you can directly click the task to act on it.

Service provider user can view tasks for any tenant by following below steps:

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click Accounts.
3. Click All Accounts.
4. Select an account on the left side pane. You will see the details of the account on the right side pane.
5. Click Tasks tab on the right side pane. You can view the pending tasks.

Any user can take action on the visible tasks. Follow below steps to act on any pending task:

1. View the task by following the steps mentioned in the above section.
2. If the task is in pending state, then you will see Click to Complete link. Click it.

The Review and Confirm window appears.

3. Enter your observation in the Memo field.
4. Click Reject or Approve to take action on the task accordingly.

Note: If you enter more than 255 characters in the Memo field while rejecting a Finance Approval task, then you get an error.

Account Management

Accounts or tenants are customer billable entities. An account is the entity for which usage is aggregated and invoices are generated. Accounts are created for the channel or selling avenue to which they belong. Every channel has its accounts or customers. Typically, there is one account per customer. Each account has users, who are actual people who interact with the system and consume resources. CloudPortal Business Manager tracks resource usage at a user level, but billing is done at an account level.

An account can be created by the administrator or by a customer through self-registration. Once the account is created, the customer identity is verified. Depending on the account type, verification of billing information and further approvals may be required. Once these activities are completed, the account gets activated.

As a service provider, you can manage your customer accounts. There are a host of actions that you can perform such as manage users of an account, edit customer account details, issue credit to customers, view billing history of accounts, change account state, and so on.

This section explains how to manage accounts:

Account Types
Account Lifecycle
Account Properties
Configuring Accounts
Managing Accounts
Monitoring Accounts

Account Types

The Account Type usually reflects the payment arrangement that has been agreed upon with the customer and the relationship that the service provider has with the customer. The signup process varies in parts depending on the Account Type.

CloudPortal Business Manager provides a special account type called SYSTEM. The System account type represents the Service Provider and root user and can be customized in an installation. In addition, CloudPortal Business Manager offers the following three default account types:

- **Retail Account:** This account type is periodically billed to a credit card on file.
- **Corporate Account:** These accounts are assigned to corporates with established credit. This account type requires additional processing and hence additional time to activate.
- **Trial Account:** This is a Trial account type.

As a service provider, you need not use these default account types. You can create custom account types that suit your business requirements, during the CloudPortal Business Manager implementation.

CloudPortal Business Manager assigns default parameter values to accounts that you create. You can edit the default account type parameter values for account types that you have defined. The System account type properties are view-only and cannot be edited.

Account Lifecycle

An account can be moved automatically or manually through various states from Active to Terminated, not necessarily in the order shown below. After the account has reached the final Terminated state, the customer can no longer interact with CloudPortal Business Manager, but the account data is retained. If you want to completely remove the account, see [Removing Accounts](#). To handle payment failures when the autopay feature is in use, see [Handling payment failures](#).

State	Description
New	The account is just created and has not yet been activated by the user.
Active	Operational. The customer can log in, create users, enable services and manage and start using resources.
Restricted	The only customer user who can log in is the Master User. Existing services continue to run but the customer cannot provision any new service.
Suspended	All services are shut down and cannot be restarted. The only customer user who can log in is the Master User. The customer cannot start any new services, but can view and modify financial information; for example, provide a new credit card number.
Terminated	The customer relationship has ended. The account still exists but is disabled and the customer cannot use it. All services are shut down and removed. The account's users are all deleted. All domains for this customer are deleted from the cloud. However, the account is not deleted from CloudPortal Business Manager. The account data is retained in the CloudPortal Business Manager database.

Account Properties

All accounts have properties and default values are defined for each of these properties based on the account type. Each new account, of an account type, is created with these default values. You can modify these values when you want the changes reflected in all accounts of an account type. You can also over-ride some of these values for individual accounts.

The various account properties are grouped into the following:

Account Details

Parameter	Parameter Definitions
Description	Description of the account type.
Max users	Maximum number of users that are allowed for this account type.

Onboarding Control Parameters

Parameter	Parameter Definition
Self Registration Allowed	Set this to True if you want to allow self-registration by users registering from the Sign Up page. Self and Manual registration values can be True simultaneously.
Manual Registration Allowed	Set this to True if you want to allow manual registration, that is, if you want an administrator to be able to sign up new users. Manual and Self registration values can be True simultaneously.
Manual Activation	Set this to True if you do not want to allow the account to be activated automatically, and want to allow only administrators to activate the account.
Default Self Registered	This is an add-on to the Self Registration Allowed parameter. The prerequisite for setting this parameter to True is that Self Registration Allowed must be set to True. Set 'Default Self Registered' to True if you want this account type to be selected by default.
Default Registered	This is an add-on to the Manual Registration Allowed parameter. The prerequisite for setting this parameter to True is that Manual Registration Allowed must be set to True. Set 'Default Registered' to True if you want this account type to be selected by default.
Account Restriction Grace Period	Numerical value in days. Set the grace period that users must get after the payment due date to make any due payment. The account gets restricted if no payment is made within the account restriction grace period. If the account has the "Auto Pay" flag enabled then the grace period starts after reaching the payment failure threshold. A "-1" value signifies that the grace period has no limit. This is the only required field in this group.
Initial Deposit Required	Set this value to True if Initial Deposit is a mandatory requirement (for this account type).
Payment Info Needed	Set this to True if you want to save the payment information on file.

Service Control Parameters

The service control parameters are specific to an instance of the cloud service that the service provider is providing.

Billing Control Parameters

Parameter	Parameter Definition
Notional Billing	Set this to True if you want the rated usage to get accrued to the credit balance.
Auto Payment Needed	Set this to True if automatic payment is required. The prerequisite for setting this value is that Payment Info Needed must be True. If set to True, all subscriptions are charged immediately on subscribing and payment is deducted from the user.
Credit Exposure Breach	The action to be taken when an account breaches its credit limit. The options are: No Action, Alert Internal, Alert, Auto-pay, Suspend, and Restrict.
Extend Credit	Set this to True if no credit balance is required for using the account.
Credit Card PreAuthentication Required	<p>CloudPortal Business Manager uses your credit card on file to obtain approval from your issuing authority (bank or other financial institution). The amount for which approval is sought is equal to your invoice amount. The expected charge on your credit card is secured (also called 'held') by your approving authority. Your approval authority then allocates an approval code with which CloudPortal Business Manager realizes the transaction amount.</p> <p>Note: Note that pre-authorized transactions appear with a minus sign on your credit card detail till such transactions are realized.</p>

Credit Exposure Parameters

This is a list of supported and available currencies.

Configuring accounts

As a service provider, you can add your customer accounts, edit account properties, and configure the onboarding settings. You can configure access based on geographic locations and email domains. You can also configure lockout settings and payment failure settings for the accounts.

Editing default account properties
Configuring onboarding settings
Configuring account lockout settings
Configuring payment failure settings
Configuring account resource limits
Enabling initial deposit for an account type

Editing default account properties

As a service provider, you can assign default parameter values to the customer accounts that you create. Each new account, of a particular type, is created with these default values. You can edit the account properties and specify new values for the parameters from the Account Types tab of CloudPortal Business Manager.

The various account parameters are grouped into the following types:

- Details
- On-boarding Controls
- Service Controls
- Billing Controls
- Credit Exposure

For more information on the parameters, refer to [Account Properties](#).

You cannot edit the properties of the Service account type.

1. Log in to the CloudPortal Business Manager UI as Root User.
2. Click the drop-down menu icon in the top right corner next to ROOT USER, and then click **Administration**.
3. Click **Account Types** and select the account type whose property parameter values you want to modify.
4. Click **Details**. This page displays the description of the account type and the default number of users that are allowed for this account type.
5. Click **Edit**.
6. Specify the maximum number of users allowed for the account type and click **OK**.
7. Click **On-boarding Controls**. You can view the list of the editable parameters grouped in this category.
8. Click **Edit**, make the changes as required, and then click **OK**.
9. Click **Service Controls**. You can view the list of the editable parameters grouped in this category.
10. Click **Edit**, make the changes as required, and then click **OK**.
11. Click **Billing Controls**. You can view the list of the parameters grouped in this category.
12. Click **Edit**. You can edit only two parameter values in the Billing Controls category.

13. If required, modify the action to be taken when an account breaches its credit limit, specify if credit card pre-authentication is required, and then click **OK**.
14. Click **Credit Exposure**. This is the numerical currency value that denotes the credit limit for this account type. The prerequisite for setting this value is that **Extend Credit** must be True. You see a list of supported and available currencies.
15. Click **Edit**, make the changes as required, and then click **OK**.

Configuring Onboarding Settings

You can configure the onboarding settings for customer accounts by specifying whitelist and blacklist for countries and email domains and the default channel name to be used on account signup. You can specify the ISO 3166-1 alpha-2 codes for countries and specify email domains to grant or deny access to customers. You can manage your onboarding settings from the Account Management pane of the CloudPortal Business Manager UI.

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. In the Configuration tab, click **Account Management**.
4. In the Onboarding area, click **Configure**, and then click **Edit**.
5. Configure the geolocation settings by specifying the whitelist and blacklist for countries.
6. Specify the whitelist and blacklist for email domains.
7. Specify the default channel name to be used on account sign up. By default, the first channel, that you configure, is considered as the default channel. All accounts that are created by self-registration are created under this default channel. However, you can specify any channel as the default channel.
8. Click **Save** to save your changes and then click **Close** to close the dialog box.

Also see [Configuring access based on geographic location](#) and [Configuring access based on email domains](#).

Configuring access based on geographic location

You can grant or deny access to customers based on their geographic location. CloudPortal Business Manager comes integrated with a geolocation application that is used to arrive at the geographic location of users. The country is identified from the IP address of the user. You can provide a list of countries that are either blacklisted or whitelisted. You can specify ISO 3166-1 alpha-2 codes for countries and either allow or deny users from these countries to sign up. These lists are used for users of all account types. You can specify the lists from the Account Management pane of the CloudPortal Business Manager UI.

The settings are enforced during the following events:

- When a user attempts to sign up from the Sign Up page.
- When a Root User attempts to sign up a user.

If a user tries to access the portal from a country that you have marked as blacklisted, the geolocation application rejects the login attempt and directs the user to the CloudPortal support page. A country whitelist always takes precedence over a country blacklist. For example, If a country is marked in the whitelist and also in the blacklist, the country is treated as whitelisted.

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. Click **Configuration > Account Management > Onboarding**.
4. Click **Configure**, and then click **Edit**.
5. In the Value box for the parameter `country.whitelist`, enter the ISO 3166-1 alpha-2 country codes separated by commas. This is the whitelist of countries from which users are allowed to sign in. For more information on 3166-1 alpha-2 ISO country codes, go to http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2.
6. In the Value box for the parameter `country.blacklist`, enter ISO 3166-1 alpha-2 country codes separated by commas. This is the blacklist of countries from which users are not allowed to sign in.
7. Click **Save** to save your changes and then click **Close** to close the dialog box.

Configuring access based on email domains

You can grant or deny access to customers based on their email domains. Users who attempt to sign up with an Email domain that you mark as blacklisted will have their sign up process aborted with a message 'Your email domain is not supported. Kindly contact Support.' You can specify a whitelist and blacklist for the email domains to either allow or deny users to sign up. These lists are used for users of all account types. You can specify the lists from the Account Management pane of the CloudPortal Business Manager UI.

The settings are enforced during the following events:

- When a user attempts to sign up from the Sign Up page.
- When a Root User attempts to sign up a user.
- When a Root User or a regular user attempts to edit an existing email domain.

If a user tries to access the portal from an email domain that you have marked as blacklisted, the application rejects the login attempt and directs the user to the CloudPortal support page. An email domain whitelist always takes precedence over an email domain blacklist. For example, If an email is marked in the whitelist and also in the blacklist, the domain is treated as whitelisted.

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. Click **Configuration > Account Management > Onboarding**.
4. Click **Configure**, and then click **Edit**.
5. In the Value box for the parameter emailDomain.blacklist, enter a comma separated list of email domains for which you want to deny access.
6. In the Value box for the parameter emailDomain.whitelist, enter a comma separated list of email domains for which you want to allow access.
7. Click **Save** to save your changes and then click **Close** to close the dialog box.

Configuring account lockout settings

CloudPortal Business Manager can implement an account lockout to prevent unauthorized or invalid access. You can configure account security settings and specify the number of failed login attempts before a user account gets locked and the number of failed login attempts before a CAPTCHA challenge has to be met for every additional login attempt. CAPTCHA challenges are issued to prevent unauthorized login attempts by automated or malicious software. This prevents brute force attacks.

You can specify values for the `logins.lockThreshold` and `logins.captchaThreshold` properties. The `logins.lockThreshold` property determines the number of failed login attempts after which a customer's account is locked. After customers cross the number of login attempts indicated by this property, their account gets locked. They have to request for a reset of the password and unlock their user account.

The `logins.captchaThreshold` property determines the number of failed login attempts that customers are allowed before the sign in process requires a CAPTCHA challenge to be met. After customers have crossed the number of login attempts indicated by this threshold, they will be required to satisfy a CAPTCHA challenge for every additional login attempt. If they fail to clear the CAPTCHA challenge, the login will not proceed.

1. Log in to CloudPortal Business Manager UI as Root user
2. Click the drop-down menu icon in the top right corner next to ROOT USER, and then click Administration
3. Click Account Management
4. In the Security pane, click **Configure**, and then click Edit
5. Specify the values for `logins.lockThreshold` and `logins.captchaThreshold`
6. Click Save to save your changes and then click Close to close the dialog box.

Configuring payment failure settings

CloudPortal Business Manager uses a feature called auto-pay where the system automatically changes the credit card of a customer when an invoice is due for payment. You can configure your settings to handle payment failures, from the Account Management pane of CloudPortal Business Manager UI. These settings are applicable for those accounts where the Auto Payment Needed parameter is set to True.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. Click **Configuration > Account Management > Thresholds**.
4. Click **Configure**, and then click **Edit**.
5. Specify the settings to handle payment failures.

Parameter Name	Description	Action taken
thresholds.maxPaymentAttempts	<number>	The number of times CloudPortal attempts to collect payment before taking any corrective action.
paymentFailure.suspensionThreshold	<number>	When payment is not received, CloudPortal waits these many days before suspending the account and sending an email to the customer.
paymentFailure.terminationThreshold	<number>	When payment is not received, CloudPortal waits these many days before terminating the account and sending an email to the customer. The value should not be less than paymentFailure.suspensionThreshold.

6. Click **Save** to save your changes and then click **Close** to close the dialog box.

Configuring account resource limits

You can configure the number of email addresses that a user can provide to receive notifications. This is in addition to the primary email alert address that the user has already specified. The secondary email addresses can be added by editing the user details. This setting is applicable to all account types that are created by the service provider.

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT user in the top right corner and click Administration.
3. In the Configuration pane, click Account Management.
4. In the Resource Limits area, click Configure and then, click Edit.
5. Specify the number of address that a user can specify for receiving email notifications.
6. Click Save to save your changes and then click Close to close the dialog box.

Enabling initial deposit for an account type

As a service provider, you can decide what account types require an initial deposit for account activation and enable the initial deposit setting.

1. Customize the account activation workflow to include the activity `verifyDepositActivity` for account type for which we want to enable initial deposit. For more information on customizing workflows, refer to [Customizing Workflows](#).
2. Log in to the CloudPortal Business Manager UI as Root user.
3. Click the drop-down menu icon in the top right corner next to ROOT USER, and then click Administration.
4. Click Account Types and select the account type for which you want to enable initial deposit.
5. Click On-boarding Controls > Edit.
6. Select the check box for Initial Deposit Required.
7. Click OK.

Managing accounts

As a service provider, in addition to adding your customer accounts and configuring them, you have to manage the accounts. You can edit your customer accounts, change account states, issue credit to customers, and manage users of an account.

Adding accounts

As a service provider, you can create accounts for your customers based on the market segments they belong to. Suppose your sales team has landed a deal with a new customer who wants to use your cloud service. After receiving the contact details of the customer and other information from sales, you create an account. Once you create accounts, they can utilize the resources and can be billed accordingly. You can create accounts from the Accounts tab of the CloudPortal Business Manager administration UI.

You must have account create permission to add an account.

1. Log in to the CloudPortal Business Manager administration UI as Root user.
2. Click **Accounts > Add New Account**. The Add New Account wizard opens.
3. Select the channel with which this account will be associated.
4. Click the type of account that you want to create. For more information on account types, see [Account Types](#).
5. Click **Next**.
6. Specify the information as requested by the wizard.
7. Review and confirm the details and click **Next** to add the account.

The account activation depends on the onboarding settings configured for the account type. Identity verification, billing details verification, and other approvals (as required) are obtained before account gets activated.

Creating self-registered accounts

Customers can create their own accounts from the CloudPortal Business Manager UI. If the service provider enables self registration for a particular account type, a customer can create a user account of that account type.

Self-registration must be enabled for the account type.

When you create an account by self registration, it is associated with the default channel specified. You can also create an account that is associated with a specific channel. If there is a promotional code associated with a channel, you have to add both the promotional code and the channel code to the end of the URL of the first page of the "Create a new account" wizard. The account is created in the channel specified by the channel code.

For example, if the promotional code is "abc123" and the channel code is "usa" and the promotional code "abc123" is valid for channel "usa", add this information to the end of the URL: `&promocode=abc123&channelcode=usa`. The account is created in the channel `usa`.

If the promotional code is associated with a single channel, you need to provide the channel code (for example, `&channelcode=usa`) in the URL extension.

If the promotional code is associated with multiple channels and one of those associated channels is the default channel, the account is created in the default channel. If none of the above conditions are satisfied, then account creation fails.

1. Open the service portal and click **Sign up now!**
2. Select the type of account that you want to sign up for.
3. Specify the account details as requested by the Create a new account wizard and click **Submit**.

Depending on the settings specified by the service provider, an email is sent to the Email address that the user has provided. The Email contains a link that the user must click to activate the account. The user is asked to enter and confirm a password. After the user enters and confirms a new password, the user is shown the login page. The user can then log in to the portal with these credentials. As an administrator, if you notice that the user has not activated the account, you can send a re-verification email to the user.

Editing accounts

As a service provider, you can edit customer account information such as address details. In addition, you can specify the discount and credit limit for the account, and the number of users that are allowed for the account. You can edit account information from the Accounts tab of the CloudPortal Business Manager UI.

You must have root or administrative privileges to edit accounts.

1. Log in to the CloudPortal Business Manager UI as Root User.
2. Click **Accounts**.
3. Select the account that you want to edit and click the edit icon.
4. Make the changes as required.
5. Click **Save** to save the changes you have made or click **Cancel** to close the dialog box without saving your changes.

Removing accounts

When you remove a customer account, the Master User for the account is removed. All the other users are removed when the account is terminated. An account ceases to appear in the All Accounts page after it is removed, while accounts of all other states appear in the All Accounts page.

You must have Account Create permission to remove an account.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click **Accounts > All Accounts**.
3. Select the account that you want to remove.
4. Move the cursor over the gear wheel icon. Click **Change State**, and then click **Terminated**.
5. Click **Save**.
6. Select the account that you have just terminated.
7. Place the cursor over the manage icon (that looks like a gear wheel), click **Remove**, and then click **Confirm**.

Changing account states

You can activate an account and also change the account state to restricted, suspended, or terminated and thereby progressively limit customer privileges. You can change an account state from the CloudPortal Business Manager UI.

You must have the necessary permissions to change account states.

1. Log in to the CloudPortal Business Manager administration UI as Root user.
2. Click **Accounts** and select the account whose state you want to change.
3. Move the cursor over the manage icon (that looks like a gear wheel) and click **Change State**.
4. Select the required account state and specify the reason for doing so in the Notes box.
5. Click **Save** to save the change you made or click **Cancel** to close the dialog box without saving the change.

Payments and refunds

The payments tab of the Usage and Billing section of the customer dashboard lists all payments made by or on behalf of the customer. As service provider, you can view the payment activity, record payments for corporate accounts, and issue charge back or credits to the customers, and cancel payments.

Recording payments

A root user or a user with Finance Create/Manage permissions can make/record payments on behalf corporate customers. Also, the master users of retail accounts can make payments.

To record/make a payment for a corporate account:

1. Log in to CloudPortal Business Manager as root user.
2. Click **Accounts > All Accounts**.
3. Select the desired customer in the list and click the dashboard icon.
4. Click **Usage and Billing > Payments** in the customer dashboard.
5. Move your cursor over the gear icon and click **Record Payment**.
6. Enter the amount and click **OK**.

Making refunds

Who Can Make a Refund

If you have the Finance Create/Manage permission, you can make a refund or cancel a payment.

How to Make a Refund

There are two ways to give money back to customers: credits and charge backs/cancel payments. For retail accounts you can issue charge back and for corporate accounts you can cancel payments.

Credit

A credit adds funds to the customer's balance in CloudPortal Business Manager. Any future charges incurred by the customer will be paid for first out of this credit balance.

To issue a credit:

1. Log in to CloudPortal Business Manager as root user.

2. Click **Accounts > All Accounts**.
3. Select the desired customer in the list.
4. Move your cursor over the gear icon and click **Issue Credit**.
5. Enter the amount and the reason for doing so, and click **Issue Credit**.

Issue charge back or cancel payments

A charge back returns funds through the customer's credit card at the request of the bank that issued the card. The charge back is processed through the same payment gateway that is used for routine invoice payments. For corporate accounts, the service provider can cancel payments made.

To issue a charge back or cancel a payment:

1. Log in to CloudPortal Business Manager as root user.
2. Click **Accounts > All Accounts**.
3. Select the desired customer in the list.
4. Click the dashboard icon.
5. Click **Usage and Billing > Payments** in the customer dashboard.
6. Select the desired payment and click **Issue Charge Back** for a retail account or click **Cancel Payment** for a corporate account.
7. Click **OK**.

Note: For new and terminated corporate accounts, you cannot cancel payments.

Issuing credit to customers

You can add funds to a customer's balance in CloudPortal Business Manager. Any future charges incurred by the customer will be paid for, first out of this credit balance.

You must be authorised to issue credit to a customer.

1. Log in to the CloudPortal Business Manager UI as Root user or as a user with the appropriate permission.
2. Click **Accounts > All Accounts**.
3. Select the customer account to whom you want to issue credit.
4. Place your cursor over the manage icon (that looks like a gear wheel) and click **Issue Credit**.
5. Specify the credit amount and enter the reason for issuing the credit.
6. Click **Issue Credit**.

Managing users' information of a customer account

You can view the user information of a customer account, edit the user information, deactivate a user or remove a user. You can manage the user information such as user details, reset user passwords and preferences, email addresses for notifications.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the **Accounts** tab. You can view all accounts along with the account details.
3. Select the account whose user information you want to manage.
4. Place your cursor over the manage icon (that looks like a gear wheel) and click **View Users**. You can view all users of that account.
5. Select the user whose information you want to manage.
6. Place your cursor over the manage icon (that looks like a gear wheel) and click **Edit**.
7. Make the desired changes. You can modify any of the user properties such as time zone, language, user details, and user password.
8. To add an image, click **Manage your profile image at Gravatar** and follow the instructions to upload a photo to gravatar.com.
9. To change the time zone and preferred language, follow these steps:
 - a. Click **Edit** in the Preferences pane.
 - b. Select the new time zone and preferred language and click **Save**.
10. To change the user details, follow these steps:
 - a. Click **Edit** in the Details pane.
 - b. Make the required changes and click **Save**.
11. To request the user for a password reset, follow these steps:
 - a. Click **Reset Password** in the Login Information pane.
 - b. Click **Confirm** to send a mail to the user to reset the password.
12. To deactivate or remove the user, follow these steps:
 - a. In the selected user page, click the gear wheel icon and then click the appropriate action.
 - b. Click **Confirm** to confirm your action or click **Cancel** to cancel your action.

Changing the primary email address of a user

As an administrator, you can change the existing primary email address of a user, if required, and provide a new email address as the primary email address.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Accounts.
3. Select the account of the user whose primary email address you want to edit.
4. Place your cursor over the manage icon (that looks like a gear wheel), click View Users and select the user.
5. Place your cursor over the manage icon (that looks like a gear wheel) and click Edit.
6. Click the Notification Preferences tab and provide the new email address.
7. Verify the secondary email address.
8. Navigate to User > Edit > Email Addresses.
9. Click **Make Primary**.

Managing services for a customer account

You can manage settings, user access and resources delivered for each of the enabled services for a customer account. As a service provider, you can enable a particular service instance for a customer account for the usage of the users and also manage the resources of the account.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click **Accounts**. You can view all accounts along with the account details.
3. Select the account (except accounts whose state is new) whose services you want to manage.
4. Place your cursor over the manage icon (that looks like a gear wheel) and click **Services**.
5. Click **Enable** if you want to enable a cloud service connector and add a cloud service instance.
6. Click **Manage Resources** to manage the resources for the account.

Monitoring accounts

As a service provider, you can monitor your customer accounts. You can view and monitor customer account customer account information such as active users and subscriptions, service health, notifications, and customer spend. You can review and monitor the billing history of a customer account. You can also monitor and manage the portal resources of the customer accounts.

Viewing customer account dashboard

You can view the dashboard of a customer account and monitor the account. You can view account information such as the number of active users and subscriptions, tasks, notifications, tickets, service health, and spend information.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Accounts**. You can view the list of all accounts.
3. Select the account whose dashboard you want to view.
4. Click the dashboard link that appears as a clock icon. The dashboard of the customer account is displayed.

Viewing customer account information

You can view the customer account details and manage the account information. You can filter and view the accounts based on account state, search for accounts based on company name, master user and account ID.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click **Accounts**. You can view the list of all accounts along with the account details.
3. To view the account information of a customer, either select the account or search for the account by specifying the company name, master user, or account ID.

Viewing account users

You can view the users that are associated with a customer account and review the user information.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click **Accounts**. You can view the list of all accounts along with the account details.
3. Select the account whose user information you want to view.
4. Place your cursor over the manage icon (that looks like a gear wheel) and click **View users**. You can view the list of all users of the selected customer account. By default, the master user of the account is selected and the master user details are displayed in the right pane. You can select a user and view the user details.

Viewing billing history of an account

You can view the billing history details of an account such as credits issued, discounts, subscription charges, renewal charges, current balance, and so on. You can also view the invoices and payments received.

You must have permission to view billing history of an account.

1. Log in to CloudPortal Business Manager as Root user or as a user with the appropriate permission.
2. Click **Accounts**. You can view the list of all accounts along with the account details.
3. Select the account whose billing information you want to view.
4. Place your cursor over the manage icon (that looks like a gear wheel) and click **View Billing History**.
5. To view the billing activity details, subscriptions, invoices, and payments, click the appropriate tab.

Operating your cloud

The CloudPortal™ Business Manager interface provides user friendly dashboard which allows an easy access to manage all the activities on the portal. This section describes how to operate the portal.

Understanding the dashboard
Service Health
Notifications
Tickets
Jobs and Job Management
Reports

Understanding the dashboard

The dashboard provides a summary view which varies depending on which user is logged in. When your CloudPortal installation is first set up, the gadgets and displays that are included in the dashboard for each user in your company are customized depending on your business needs.

Who Can Use the Dashboard

Every user, no matter their role, has a dashboard. The content varies depending on which user is logged in. For example:

- The Master User of a customer account can see the users in that customer account.
- Customers can see spend vs budget and spend by type reports.
- The service provider's root user or Ops Admin user sees reports on new registrations and top customers.

How to Use the Dashboard

- Log in to CloudPortal, and the dashboard is the first screen that is displayed.
- At any time while using CloudPortal, you can return to the dashboard by clicking Home > Dashboard.
- In Customer Summary, you can view graphs of statistics about customer activity. Click the tabs New Registrations, Top Customers to see different summary graphs. Click Complete Report to get more details on the reports.
- In Service Health, you can get an at-a-glance summary of your cloud infrastructure's status. Click View Details to find out more.
- In Notifications, see a summary list of recent alerts about events in your cloud. Click **View all notifications** to see the notifications for events and issues that are visible to your user role. Click **Manage Notification Preferences** to configure where your notifications will be delivered:
- In My Tasks, you can see all the pending tasks (if any).
- In Tickets, you can see the summary of the support tickets based on the ticket status.

Launching the console

As a service provider, you can manage your cloud service instances. You can access the cloud service instance UI and provision, view, and manage the cloud infrastructure, configuration settings, and so on. You can log in with administrator privileges.

1. Log in to CloudPortal Business Manager as Root user.
2. Click Launch Console > cloud service instance.

The console opens in a different tab and you can view the administrator dashboard.

Service Health

The status of your portal can be summarized in an overall rating called "health." The health status is set manually. It is visible to customer users and users within your organization who have the required permissions. Service health is a quick way to let your own personnel and customers know how the system is doing. Most users can see the service health.

You must have root or administrative privileges to set the service health.

1. Log in to the CloudPortal Business Manager UI as Root User.
2. Click Home, and you will see the Service Health icon in your dashboard.
3. Select an entry in the status history list to see more details about that event.
4. Click Add Status to change the current status of the service.
5. Select the Service Instance that the status applies to.
6. Select the degree of severity from the Type list: Issue for the yellow icon, Disruption for the red icon, or Resolution for the green icon. Provide a descriptive Subject and Description to communicate the status with users. When finished, click Add Status.

Timezone Setting for Scheduled Maintenance

To ensure that scheduled maintenance activities get created correctly, set the appropriate timezones for users.

Timezone Setting in the CloudPortal UI

1. Log in to CloudPortal as the root user.
2. Go to Administration > Configuration > Server.
3. Filter the Server parameters by Portal.
4. Set the desired value for `vmops.portal.default.user.timezone`. For example, set Japan if you want the timezone setting to be JST.

Timezone Setting in the Database

If you have several users in an account, and if you wish to run a single SQL query to set the timezone for these users in the database, you can change the the timezone setting in the database. Timezones are set in the `user_time_zone` column of the `cloud_portal.users` table in the database.

Note: In addition to running the SQL query above, you must change the default timezone in the UI as explained above to ensure that new accounts have the correct timezone.

Adding a Scheduled Maintenance

When you need to take a zone offline for necessary modifications, you can schedule a maintenance window. The maintenance window schedule is visible to customer users and users within your organization who have the required permissions. This provides a way to let your own personnel and customers know when to expect possible service interruptions as well as improvements in the service.

1. Select Administration > Service Health.
2. Select Add Scheduled Maintenance. .
3. In Service Instance, select the instance where the maintenance will take place.
4. In Start and End, set the time window during which maintenance will take place. In each field, use the calendar control and time sliders, then click Done.
5. In Subject and Description, provide information for your users to explain what is going on during the maintenance window. Typically, either an issue is being repaired or an upgrade is being performed to provide new features.
6. Click Schedule. The new event is displayed under Scheduled Maintenance in the Service Health screen.
7. If you want to change the timing or delete the maintenance window, click Edit or Delete under Scheduled Maintenance in the Service Health screen.

Notifications

CloudPortal™ Business Manager provides a variety of notifications for both the routine events and issues that require attention. Notifications are displayed in the UI and also sent to the specified emails. Notifications come from two sources:

- CloudPortal Business Manager events and alerts, such as the creation of a new account.
- CloudStack events and alerts which are passed through from the underlying infrastructure.

Viewing Notifications

Most users at a CloudPortal service provider can view at least some of the notifications. The notifications that are displayed vary according to who is logged in. For example, a user with the Ops Comm permission will see notifications about system health, whereas a user with the Account Manage permission will see notifications related to customer activity.

Click Home, and you will see a summary of recent notifications on your dashboard.

Click the More link to display the All Notifications screen. Click the Notifications link in the top menu bar. This also displays the All Notifications screen.

If you have subscribed to receive notifications in Email, view them in your Inbox.

Subscribing to Notifications

Click the **Notifications** link in the top menu bar.

Click **Manage Notification Preferences**.

Fill in your Email address.

Click Add Email.

Configuring notification settings

CloudPortal™ Business Manager allows you to configure the notification settings to control the ways in which you can receive or post the notifications about any events or alerts. To configure the notification settings:

1. Log in as root user.
2. Select Administration .
3. Scroll using < or > in the Configuration tab and select Server.
4. Select Notifications .
5. Click Configure.
6. Click Edit.
7. Update notification.to field. This is the email address to which error notification mails will be sent.
8. Click Save.

Managing Notification Preferences

CloudPortal™ Business Manager allows you to determine how you view notifications by setting a notification preference. You can override the default by modifying the notification preference.

1. Login as a root user.
2. Click Home > Manage Notification Preferences.
3. Select the delivery channel where you want the notifications to be delivered. It can be either SMS or Email.
4. If you want to add an email address, Click Add/Delete Email Address.
5. Specify the email address you want to add and Click Add Email.

Tickets

About Tickets in CloudPortal

CloudPortal™ Business Manager provides features to help you manage customer relationships.

CloudPortal is integrated with [Salesforce](#) Service Cloud. This is used to log requests for support (trouble tickets) from customers. The CloudPortal trouble ticketing UI, allows you to view and submit trouble tickets that are stored in Salesforce. Trouble tickets that were created in either CloudPortal or Salesforce are visible in both locations.

The connection between Salesforce and CloudPortal is set up using configuration parameters, as described in [Configuring Salesforce](#).

Filing a Ticket

You can file a trouble ticket on behalf of a user, either using the CloudPortal™ Business Manager UI or directly in Salesforce.com using your Salesforce login credentials.

If you have the Account Create or Account Manage permission, as well as the Trouble Ticket Manage permission, you can file a trouble ticket on behalf of a user.

1. Login as Root.
2. Click Accounts, and select the account you want to work with.
3. In the Tickets area of the dashboard, click Submit Ticket.
4. Click Add New Ticket.
5. Click Submit.

Closing a Ticket

When the issue described in a trouble ticket is solved, it is customary to close the ticket. You can close the trouble ticket on behalf of a user, either using the CloudPortal™ Business Manager UI or directly in Salesforce.com using your Salesforce login credentials. To access the Tickets screen through a customer's Dashboard :

1. Click Accounts, Select the account you want to work with.
2. In the Tickets area of the dashboard, click View All Tickets.
3. Select the ticket you want to close.
4. Click Close.

Jobs and Job Management

Jobs are system-generated tasks to run in a given time period to accomplish certain activities like billing. Jobs are used to perform routine tasks such as tracking usage, mediation, generating invoices, account statements, and so on. You can configure custom jobs, schedule jobs, view status of jobs, and view exception reports when jobs fail.

Jobs are scheduled to run at a particular time and recur at a set interval. Batch jobs are implemented as JavaBeans (<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138795.html>) and are run using the Spring (<http://www.springsource.org/>) platform. The batch jobs are scheduled in two XML files in the CloudPortal installation path: applicationContext-jobs.xml and applicationContext-scheduler.xml. You can edit these files to change the job schedule. After editing the files, you must restart CloudPortal server for the changes to take effect.

You can create custom jobs based on your business needs. For information on configuring custom jobs, refer to the Customization SDK guide.

Types of jobs

Jobs are system-generated tasks to run in a given time period to accomplish certain activities like billing.

The jobs in CloudPortal Business Manager include the following:

Batch Job Name	Start time	Dependence on others	Seen in UI	Job successful sign	Overview	Notice mail when job fails
----------------	------------	----------------------	------------	---------------------	----------	----------------------------

Types of jobs

<p>Manager (dailyBillingJob)</p>	<p>1:00 A.M. every day</p>	<p>None</p>	<p>Yes</p>	<p>Status is displayed in UI as completed, with adjacent blue circle. Check the description in the UI in case of error status.</p>	<p>All billing processing is done by "DailyBillingJob". By default, this job is scheduled to run at 1:00 A.M. server time. There are two sub-processes in the DailyBilling job:</p> <ul style="list-style-type: none"> •First sub-process scans through all the service instances and fetches their usage data for the previous day. This data is converted into useful information which CloudPortal Business Manager can understand and is persisted. •Second sub-process scans through all the tenants, picks up the information from the previous sub-process, uses it to update or post account statements, and make payments. <p>Both of these sub-processes are transactional; if any of the service instance fails during the reporting of daily usage, it is skipped and the next service instance is requested for its usage data. Similarly, if the second sub-process fails for a tenant, say during processing payment, it is once again skipped and the second tenant is processed.</p>	<p>Yes. Mail is sent to Admin support - on error, partial error, and warning.</p>
----------------------------------	----------------------------	-------------	------------	--	---	---

Types of jobs

<p>Trial Expiry Job (trialExpiry)</p>	<p>3:10 A.M. every day</p>	<p>None</p>	<p>Yes</p>	<p>Status is displayed in UI as completed, with adjacent blue circle. Check the description in the UI in case of error status.</p>	<p>This job runs once every day and checks all trial accounts and calculates the time left for expiry of each account. Depending on the configuration, mails are sent to inform the customer of the days left until the trial expires. On expiry of the trial period, if the account is not converted to a paying account, the account is suspended. After account suspension, if the account is not still converted to a paying account within the configured number of days, then the account is terminated.</p>	<p>Yes. Mail is sent to Admin support - on error, partial error, and warning.</p>
<p>Cloud Service Event Processing Job (cloudServiceEventProcessing)</p>	<p>Every 6 minutes</p>	<p>None. This job does not run when the daily Billing job is running.</p>	<p>Seen in UI only if job fails.</p>	<p>Check the description in the UI in case of error status.</p>	<p>This job polls connectors for every service instance and provides a list of events that CloudPortal Business Manager can understand. As a service provider, you can subscribe to these events and run any process that you want, such as monitoring, and so on. This job does not run when the dailyBilling job is running.</p>	<p>No mail is sent.</p>

Types of jobs

Work Flow Job (workflowJob)	Every 5 seconds	None	Seen in UI only if job fails.	Check the description in the UI in case of error status.	This job runs every 5 seconds and is responsible for running the workflow engine which picks up any workflow in NEW and RUNNING state and runs it. Only one instance of this job runs at one moment of time. For more information on workflow engine, refer to the Workflow Management section in the Administration Guide.	Yes. Mail is sent to Admin support - on error, partial error, and warning.
Subscription Validation Job (subscriptionValidationJob)	Every 1 hour	None	Seen in UI only if job fails	Check the description in the UI in case of error status	This job runs every hour and checks if resource handle is active for each active subscription. If the resource handle is not active, it marks the subscription as free to re-provision another resource.	Yes. Mail is sent to Admin support - on error, partial error, and warning.
Billing Export Job	4 A.M. every day	None	Yes	Status is displayed in UI as completed, with adjacent blue circle. Check the description in the UI in case of error status.	<p>This job runs once daily and generates a file with the following information:</p> <ol style="list-style-type: none"> 1. Account statements posted the day before 2. Invoices posted the day before 3. Payments and credits made the day before 4. Payment failures that occurred the day before <p>This file is stored locally in the export.directory and exported to the export.URL as specified in Configuration > Billing.</p>	Yes. Mail is sent to Admin support - on error, partial error, and warning.

Types of jobs

Daily Custom Job (daily JobTrigger)	10:15 A.M. every day	None	Yes	Status is displayed in UI as completed, with adjacent blue circle. Check the description in the UI in case of error status.	This job runs once every day and generates all custom reports if any such implementation is done.	Yes. Mail is sent to Admin support - on error, partial error, and warning.
Monthly Custom Job (monthlyJobTrigger)	10:15 A.M. every first day of the month	None	Yes	Status is displayed in UI as completed, with adjacent blue circle. Check the description in the UI in case of error status.	This job runs once every first day of the month and generates all custom reports if any such implementation is done.	Yes. Mail is sent to Admin support - on error, partial error, and warning.

Viewing batch jobs

As a service provider, you can view the status of the batch jobs and also view the exception report for failed jobs. To view the status of batch jobs that are running now or have run in the past, you can use the CloudPortal Business Manager UI.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. Click **Batch Jobs** and view all jobs along with their details. For jobs that fail, you can view the exception report in the description section for the report.

Reports

The CloudPortal Business Manager reports help you get an overview of what is going on in your service. These reports are displayed in a graphical form and can be exported in various formats such as PNG, JPEG, PDF, and SVG vector.

CloudPortal Business Manager provides the following reports:

- **New Registrations:** By default, this report displays the new registrations in the current month. You can generate and view this report for a particular period by specifying the start date and end date.
- **Top Customers:** By default, this report displays the top customers in the current month. You can generate and view this report for a particular month by specifying the month and year.
- **Product Usage:** By default, this report displays the product usage in the current month. You can generate and view this report for a particular month by specifying the month and year.
- **Bundle Usage:** By default, this report displays usage by product bundle in the current month. You can generate and view this report for a particular month by specifying the month and year.
- **Custom Reports:** These are reports that you have defined based on your business needs for information, reporting, and monitoring. You can generate and view these report by selecting a report and generating it. You can also download these reports and forward them via email.

Custom Reports

When first installed, CloudPortal Business Manager provides a set of built-in reports. With the help of CloudPortal Professional Services, you can add custom reports to meet your specific reporting needs. Custom reports are generated as ad-hoc reports through the CloudPortal Business Manager UI, and can accept input parameters.

Note: The out of the box reports display the monetary values in default currency (value of the config `com.citrix.cpbm.portal.settings.default.currency`). In order to convert values from other currencies to default currencies CPBM uses conversion factor (column "conversion_factor") available in table "cloud_portal.currency_value". This conversion factor is multiplied by the amount in the corresponding currency to get appropriate value in the default currency. Service provider must update this column directly into the database.

For more information on customizing reports, refer to the Customization SDK guide.

Configuring report settings

You can configure general report settings such as the directory to which the daily registration reports are exported, specify whether aggregation is based on product or category in daily usage reports, and the email address to which reports are exported.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click the drop-down menu icon next to ROOT USER in the top right corner and click **Administration**.
3. Click **Configuration > Reports**.
4. To specify the directory to which the daily registration reports are exported:
 - a. Click **Exports > Configure**.
 - b. Click **Edit** and enter the value for `registrations.reportpath`. This is the directory where daily registration reports are exported.
 - c. Click **Save**.
5. To specify the basis for aggregation for daily usage reports:
 - a. Click **Mediation > Configure**.
 - b. Click **Edit**.
 - c. Enter the value for `aggregation.category`. This is the aggregation type of usage that is recorded in daily usage reports. It can either have P (By Product) or C (By Category) as its values.
 - d. Click **Save**.
6. To specify the email address to which the reports are exported:
 - a. Click **Notifications > Configure**.
 - b. Enter the value for `reports.email`. This is the email address to which you can send the reports.
 - c. Click **Save**.

Viewing reports

As a service provider, you can view the reports to get an overview of your service and to monitor your service offerings. You can generate reports to view trends

1. Log in to CloudPortal Business Manager UI as Root user.
2. In the Reports tab, click the report type you want to view. By default, the reports display data for the current month.
3. To generate the report for a particular period, either specify the time or month and year, and click **Submit**.
4. To generate and view custom reports:
 - a. Click **Custom Reports**.
 - b. Select the report that you want to generate and click **Generate**. You can view the report, download it, and send it via email.

Product Management

CloudPortal Business Manager has a comprehensive Product Management module. The product management feature allows a product manager to design and manage a catalog of products and product bundles. These products and bundles are then purchased and consumed by the end user.

The product manager can create products, package them into product bundles which can be purchased, and manage catalogs that can selectively expose the product bundles that are available for the end user to purchase. The product manager can also provide specialized pricing on a per channel basis across multiple currencies. The product management capabilities of CloudPortal Business Manager are accessible to any service operator user with the 'Manage Products' role.

This section describes how to create products, product bundles, product catalogs, and manage them.

Products
Product Catalogs
Product Bundles

Products

A product is a basic unit of consumption in CloudPortal Business Manager. Every product has a utility or unit price associated with it. Every product has a name, a unique code (to identify it uniquely) and a description that helps the consumer understand the product better. Each product has a unit of measure which is the unit at which this product is charged for consumption and a set of mediation rules that help CloudPortal Business Manager classify usage into the various products defined for a particular service.

The service provider may or may not explicitly provision resources based on the roles. Sometimes a particular usage, that is not explicitly defined, gets accrued in conjunction with the use of a product and can, therefore, be charged to the customer. A product is created for anything a service provider wishes to charge a different price. Also, products are defined for a specific instance of a service.

Understanding products

Products represent anything that can be charged for. The usage data generated gets converted into a form that is suitable for billing. The service provider defines one or more mediation rules for every product.

Mediation rules

When Service is integrated with CloudPortal Business Manager, the service metadata describes all the resource consumption measured by the service, and any attributes that describe the resources being consumed. A service operator can use this information to define products with rules to that determine how much of the usage collected for a specific resource is classified as usage of the defined product. This process is known as mediation and the rules are called mediation rules.

For example, we can consider the mobile service where phone calls is the measured service being consumed. The service would then return the number of minutes (or the unit of measure) of each call and who used it. We can consider this usage type as CALL. The service will also describe the call based on certain pre-determined attributes such as:

- TimeOfDay: daytime/night time/evenings/weekends
- Destination: local/long distance/international
- Origin: roaming/local

A service operator can determine, based on the business needs, if all calls are charged the same rate or if a different rates are applicable for weekends and evenings, local or long distance, and so on. The service provider would create different products for each classification. They may charge all calls the same, but have another product called daytime surcharge, which has an additional price per unit for only the daytime minutes. These classifications are decided by the service provider and will vary depending on the business needs served by each installation.

The following is an example of how mediation rules can be defined for mobile calls.

- **All calls are the same price regardless of other characteristics:**

Usage Type = CALL

Since there is no subclassification, this is all that needs to be declared. The product creation wizard enables you to create this product.

- **Evenings, nights & weekend calls and daytime calls as separate products:**

Mobile call (evenings, nights and weekends) = Usage Type = CALL (TimeOfDay excludes daytime)

Mobile call (daytime) = Usage Type = CALL (TimeOfDay includes daytime)

- **Evenings, nights & weekend calls and daytime calls as separate products for local and all long distance and international calls are the same price regardless of time of day:**

Mobile local call (evenings, nights and weekends): Usage Type = CALL (TimeOfDay excludes daytime, destination includes local)

Mobile local call (daytime): Usage Type = CALL (TimeOfDay includes daytime, destination includes local)

Mobile long distance and international calls: Usage Type = CALL (destination excludes local)

Unit of Measure

When a service measures usage, it does so in a specific unit. However, an operator may choose to price things in other scales. For example, while the mobile service measures call duration in seconds, but the operator wants to charge in minutes. To achieve this, a product can have a conversion factor which is used to convert usage for a particular product from the unit of measure of the usage to the unit of measure of the product. Services typically declare common scales to scale up or down their unit of measure. A service provider may also specify a custom conversion factor and provide a corresponding unit of measure. For example, the mobile service may measure calls in seconds and may define scales such as 6seconds, half-minute, or minute. The operator could then choose whichever unit they want to price the calls. Alternatively, they could create a custom unit called 15 seconds and specify the appropriate conversion factor.

Combining and Excluding Usage

In some cases, it may be necessary to combine multiple usage types. For example, assume that the service measures incoming calls and outgoing calls separately. The operator can either price incoming and outgoing calls separately, each having products as described earlier or charge for both incoming and outgoing calls the same way. In this case, the operator could add both usage types with the COMBINE operator to the product (with their corresponding mediation rules) and combine the two usages. Alternatively, if the service measures all calls and outgoing calls separately and if an operator wishes to charge for incoming calls separately, they could COMBINE (or add) all calls into this product and EXCLUDE (or subtract) outgoing calls to derive the value of incoming calls. The exact definitions depend on what the service measures and what the operator wants to charge for.

Usage Processing

A given source usage record will be processed as follows:

- A single usage record can match multiple products based on the rules for individual products. All matching products will accrue usage as described in the rule.
- Two rules can target the same product (this allows two source usage records to be added or combined).

Pricing

Once a product is defined, reference prices in all supported currencies may be set for the product. These prices can be overridden on a per channel basis.

Charge override functionality in channels

Every channel can override price or charge from the reference catalog for a product or product bundle. However, mediation rules, entitlements, and provisioning constraints cannot be over-ridden.

Adding a product

You can create a new product to target new resources or to replace a retired product. If it is a fresh product, you can add the product to the very first revision of the Reference Catalog and to the current revision of all the channel catalogs. The price that you set for the product remains in the channel catalogs. If you would like to change the price at the channel catalog level, you can do so only in the future plan of the channel catalog.

1. Log in to CloudPortal Business Manager as the service provider user having "Product create manage" role.
2. Click the Catalog tab and choose Plan Next from the drop-down.
3. Select the service instance for which you want to create the product.
4. Click Add New in Products dashboard to add a product.
5. Add Product wizard appears. Enter the details to define the product and select the product category from the drop down list.

Note:

- Do **not** select the checkbox "This product is replacing one or more products" if you are creating a new product.
- Select the checkbox only if you are replacing this product with an existing product.

Note: If the product does not replace one or more existing products, it gets added to the reference catalog immediately and is visible in the current view of the catalog. If the product is replacing an existing product, it gets added to the reference catalog only after activation.

6. Click **Next**.
7. Select the usage types, from the drop down list, that will impact this product. Choose the appropriate aggregation handling or operator (combine or exclude) for every usage type.
8. Click on "+" sign to enter the next usage type. After entering the details for the next usage type, click **Next**.
9. Enter the mediation rules for the usage type. The mediation rule will be applied on how usage needs to be collected. Click **Next**.
10. Specify the conversion factor for the product units and click **Next**.
11. Enter the utility or product charges for the active currencies and click **Next**.
12. Review the details you entered and click **Add**.
13. Click **Close**.

Note: The product is added to the first revision of the reference catalog. The product also appears in historical revisions of the reference catalog. However, the charges of the new product will be copied over in all the channels in their respective current revisions. The charges can be over-ridden in the channels only in the next (or future) revision.

Replacing an existing product with a new product

1. Log in to the portal as the service provider user having "Product create manage" role.
2. Click the Catalog tab and choose Plan Next from the drop-down.
3. Click on Add New button in Products dashboard to add a product.
4. Add Product wizard appears. Enter the appropriate details to define the product.
5. Select the checkbox "This product is replacing one or more products".

This product will appear in the current reference catalog only after activation. The product will appear in the channel catalogs once they are synchronized with the reference catalog.

6. Click **Next**.
7. Select the usage types, from the drop down list, that will impact this product. Choose the appropriate aggregation handling or operator (combine or exclude) for every usage type.
8. Click the "+" sign to enter the next usage type. After entering the details for the next usage type, click **Next**.
9. Enter the mediation rules from the drop down list for the usage type. The mediation rule will be applied on how usage needs to be collected. Click **Next**.
10. Specify the conversion factor for the product units and click **Next**.
11. Enter the utility or product charges for the active currencies and click **Next**.
12. Review the details you entered and click **Add**.
13. Click **Close**.

Post requirements: Activate the changes so that the product appears in the reference catalog.

Editing a product

1. Log in to CloudPortal Business Manager as the service provider user having "Product create manage" role.
2. Click the Catalog tab and choose Plan Next from the drop-down.
3. You can view the products on the left pane as shown below. Select the product that needs to be edited.
4. You can view the details of the selected product on the right pane. Place your cursor over the Manage icon (that looks like a gear wheel) and click **Edit**.
5. In the Edit Product wizard, update the product description.
6. Click **Save** after you have made all the changes and click **Close**.

Note: You cannot add or remove a usage type that is already defined for the product. In case of discriminators, only if you have one or more discriminators for the usage type, you can add more discriminators. If there were none defined, then you cannot add any new discriminator.

Retiring a product

You can either retire a product or replace it with a new product. If you only retire the product, it will remain active in CloudPortal and charge the users based on the usage that accrues in the active subscriptions. If there are any entitlements in any subscribed bundles, they will still remain for the product.

You retire an existing product and replace it with a new product. This replaced product will not be charged from next plan of a channel catalog after it syncs with the reference catalog.

You cannot retire a product, that is being used as entitlement unless it is removed from all the bundle entitlements.

Note: A channel syncs with the current plan of the reference catalog. In case of changes in the reference catalog, the channel must sync with the reference catalog (when its future plan becomes active), so that the changes that take effect in the future revision of the reference catalog appear in the future revision of the channel catalog.

1. Log in to CloudPortal Business Manager as the Root user.
2. Click the **Catalog** tab and choose "Plan Next" from the drop-down.
3. You can view the products on the left pane. Select the product that you wish to retire.
4. You can see the details of the selected product on the right pane. Place your cursor over the Manage icon (that looks like a gear wheel) and click **Retire**.
5. If you do not wish to replace the product by other products, click "No" in the "Retire Product" dialog box. The product will not appear in the future reference catalogs.

Note:

- A retired product cannot be added as an entitlement in a product bundle.
- A retired product will continue to be charged based on the usage if it is not being replaced.
- A retired product will not appear in the catalog from the future plan. Therefore, no new use of the product is possible.

Replace an existing product

1. Perform the steps 1 to 3 as mentioned in the above section "Retiring a product".
2. Select "Yes" if you wish to replace the existing product in the "Retire Product" dialog box.

Editing product pricing

Depending on your business needs, it may be necessary to edit the product prices. The changes that you make to the product prices are visible in the current reference catalog only after activation.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click **Catalog > Plan Next**.
3. Select the service instance for which you want to edit the product pricing.
4. Click **Products > Plan > Edit Charges**. The Edit Planned Charges dialog box opens.
5. Make the changes as required to the product prices and click **OK**.

You have to activate the changes to make them visible in the current reference catalog. Refer to [Scheduling activation](#).

To see a new revision of product definitions, a catalog needs to 'Synchronize' the catalog to the revision. Refer to [Synchronizing with reference catalog](#).

Price changes to existing products and bundles are not synchronized up to a channel catalog and have to be explicitly managed. When managing prices for a catalog (both products and bundles) the reference price of the product or bundle is visible to the product manager (in parentheses) in the product management UI.

Product Catalogs

A catalog is a set of product bundles and prices that can be offered to customers. CloudPortal Business Manager enables you to define multiple catalogs with different products and prices. The default catalog is called the Reference Catalog.

You can define a different catalog for each channel (which is called channel catalog). Each catalog has an associated utility rate card that helps to calculate the charge when the customer exceeds the entitlements in the purchased product bundles. For example, assume that a customer has purchased a product bundle that includes an entitlement of 20 GB of storage per month. If the amount of storage used goes above 20 GB, you can charge a certain amount for each additional 100 KB.

Reference Catalog revisions

Revising a catalog includes the following:

- Adding a product or product bundle
- Editing a product or product bundle
- Editing pricing for a product or product bundle
- Retiring a product
- Replacing an existing product with a new product

When a reference catalog is revised, every channel must synchronize with the reference catalog, so that the changes are visible in the channel. After synchronization, each channel decides when the changes become visible to the customers. In case a new product or bundle is added to the reference catalog, the new product will appear in the catalog only after the catalog is activated.

Understanding current, next, and historical revisions of the catalog

You can view the current, next, and earlier versions of your reference catalog from the Catalog tab.

Current: This refers to the current version of your reference catalog. Click Catalog > View Current to view the current version. Any changes that you make, have to activated before they become a part of the current version of the catalog.

Next: This refers to the future revision of the catalog. Any changes that you make in the reference catalog will become effective in the future (on the date when the changes are scheduled to be activated). Click Catalog > Plan Next to create a new version of the catalog.

History: This refers to all versions prior to the current version of the catalog. Click Catalog > View History, and then click the date and time stamp from the drop down list to view the version of the catalog on that date.

Product Bundles

Product bundles represent the package a customer buys. Product bundle can be compared to subscription plans. When a consumer buys a subscription plan, the system has to provision the plan (perform activities such as activate the plan, associate it with a phone, give it a phone number, and so on). Once provisioned, the consumer would make calls, send SMS messages, and consume internet data. In addition, the service may also provide other services such as broadband data or cable service.

A product bundle has a name, a code, and a couple of attributes. These include the following:

- **Recurrence:** If bundle has recurring charges and if so, what is the frequency. This is also known as charge frequency. Charge frequency can be one of the following:
 - **Monthly, with no minimum recurring charge:** Pay for what you use, along with a one time subscription charge
 - **Monthly:** Besides one time, subscription renewal charges need to be paid every month.
 - **Quarterly:** Besides one time, subscription renewal charges need to be paid every quarter.
 - **Annual:** Besides one time, subscription renewal charges need to be paid every year.
- **Publish:** Whether the bundle is published or not.
- **Service Instance:** The service instance for which the bundle is created.
- **Resource type:** The resource type selected for a bundle determines what gets provisioned for that bundle. This will determine what will actually get provisioned when the customer buys this bundle.
- **Provisioning constraints:** A resource type may be provisioned with one or more configurations. For example, internet data might have the 2G, 3G, or 4G option and phone service may be local only, local-cum-ISD, and so on. Provisioning constraints describe the qualified (or disqualified) configurations that are valid for this bundle. For example, you could define a bundle that is valid only for 2G Internet data. When a customer wishes to use 3G data, this bundle will not be presented as a possible choice for subscription (other bundles may be presented depending on their configuration). As a service provider, you will be presented with available configuration options when defining constraints and you can either explicitly include certain configurations (for example, data option includes 2G, 3G) or exclude certain configurations (for example, data option excludes 4G).
- **Business constraints:** Describe operator business rules for a particular bundle. These determine how many subscriptions of this bundle is allowed per account and per user. Note that the available business constraints are sometimes limited by what is technically feasible. For example, mobile phone plans can be subscribed only once per user, so the multiple per user option will not be available. They include the following:

- No limit: No limits at the account or user level
- User level: One per user
- Account level: One per account
- Account level: No limits

Entitlements

Entitlements determine how much of what product is included as part of a subscription to a bundle. These are described as the amount of units of one or more products included in the subscription. It is possible to include entitlements to other services as part of entitlement of a bundle. For example, three pay-per-view movies free when you buy the mobile phone plan to facilitate cross-selling. The operator can give unlimited entitlements to products (for example, unlimited internet data) that can be consumed by a given resource such as a phone plan. CloudPortal Business Manager is aware of what resources can be consumed directly by a given resource, so it will allow unlimited entitlements only to such products.

Bundle Pricing

Bundles can have two prices defined per active currency, a one time charge and a recurring charge. As with products, the actual prices can be overridden on a per channel basis.

- One time charge: This is charged immediately when the consumer subscribes to a bundle.
- Recurring price: This price is charged periodically based on the recurrence defined for the bundle. For monthly bundles, the recurring charge is pro-rated to the remainder of the billing period and charged in full, in advance, for every future billing period. For all other recurrences, the charge is, in full, in advance, and will be repeated at the specified frequency.

Revising of Products and Product Bundles

Any changes, additions or deletions made when managing products and bundles are made visible to the consumer in a controlled manner. This ensures accurate billing for the consumer based on the then prevailing rates for a product or bundle. In order to ensure accuracy and auditability, all changes made to product and bundle definitions are revised. A revision is made visible (or activated) on a date specified by the operator.

At any given time (except when the product is first installed), there is a currently active revision. In addition, there is always a revision that is being planned. Whenever a revision becomes activated, a new revision is automatically ready for planning. To activate a revision, you can either schedule the activation for a particular date or activate it immediately. When there are multiple revisions for any day (if revisions are created and activate immediately, it is possible to have multiple revisions for a given day), only the last revision for that date is used by the rating system. Therefore, the option of activating a revision immediately should be used only to correct errors, as consumers may see one price (before activating the new revision) when subscribing, but be charged another price (the new revision price) when being rated.

Channel Synchronizing

Synchronizes the channel catalog with the current plan of the reference catalog. Charges are copied for the new products that come into being. Any new bundles created can now be

consumed in the channel catalog. Also, retired products are marked retired in the channel catalog. Since syncing is a modification to the channel catalog plan, the changes will take effect only in the next plan of the channel catalog.

Adding a product bundle

Product bundles represent the package a customer buys to provision a resource. As a service provider, you have to create product bundles which include specifying a service instance, resource type, entitlements, and charge frequency. You can create product bundles for your customers from the CloudPortal Business Manager.

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click **Catalog > Plan Next**.
3. In the Products Bundles tab, click **Add New**. The Add Product Bundle wizard appears.
4. In the Bundle Description page, enter the bundle details and click **Next**.
5. Specify the bundle details such as resource type, subscriptions allowed, and charge frequency. You can select from the following based on your requirement:
 - Resource types: service bundle, compute, storage, or networking.
 - Allow subscriptions: number of subscriptions supported - no limit, one per account, any number per account, and one per user.
 - Charge frequency: monthly with no minimum recurring charges, monthly, quarterly, or annual.
6. Select the checkbox "Eligible for Trial" if this bundle is for a trial account. A trial customer will see only these bundles.
7. Clear the checkbox "Enable notification" if you do not want email notifications to be sent on the subscription or unsubscription of this product bundle. By default, whenever a user subscribes or un-subscribes to this product bundle, email notification is sent to the user who has subscribed to the product bundle, the product bundle owner, and the sales support alias. Any user who has a Product Management role can turn off this setting. The sales support alias is specified in Administration > Portal > Address book > addressbook.salesSupportEmail settings.
8. Click **Next**.
9. In the Provisioning Constraints page, set the provisioning constraints for the resource components and click **Next**.
10. Enter the bundle charges and set one-off and recurring charges to be used as reference for this bundle and click Next.
11. Review and confirm the bundle details and click **Add** to add the product bundle.
12. Select the product bundle from the list of product bundles and click the **Entitlements** tab.
 - a. Select the product from the product list.

- b. Enter the number of included units per month. If you want to provide unlimited entitlements, select the checkbox to specify it.

Note: While creating a product bundle for a **Resource Type**, only **products** which are based on usage types generated by that Resource Type can be given as **unlimited** entitlement in that bundle.

If you select the checkbox for unlimited entitlement and then select the product, you are able to give unlimited entitlement even for a product that is not set to give unlimited entitlement.

Note: Entitlement is based on subscription activation date for the first billing period. From second billing period onwards, it is based on invoice start date. For monthly, it would be as on last billing cycle's last date, but for quarterly it could be up to three months back.

Post requirements: You have to publish the bundle and activate it for the bundle to be visible to the channels

Editing a product bundle

You can edit a product bundle and modify bundle description, details, and provisioning constraints. You have to schedule the activation of these changes so that they are visible in the current reference catalog.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Plan Next > Product Bundles**.
3. Select the product bundle that you want to edit.
4. Place the cursor over the manage icon (that looks like a gear wheel), and click **Edit**.
5. Make the changes as required.
6. Click **Save** and then, click **Close**.

Publishing and unpublishing product bundles

When you create a product bundle, you have to publish it and then activate it to make it visible to the channels. When you unpublish a product bundle, it is no longer visible to the channels. However, it is still a part of the reference catalog.

1. Log in to the CloudPortal Business Manager UI as Root user.
2. Click **Catalog > Plan Next**.
3. Select the service instance for which the product bundle is created and click **Product Bundles**.
4. Select the product bundle that you want to publish or unpublish.
5. Place the cursor over the manage icon (that looks like a gear wheel), and click **Publish** or **Unpublish**, as required.
6. Click **OK**.

Editing product bundle pricing

Depending on your business needs, it may be necessary to edit the product bundle prices. The changes that you make to the product bundle prices are visible in the current reference catalog only after activation.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click **Catalog > Plan Next**.
3. Select the service instance for which you want to edit the product bundle pricing.
4. Click **Product Bundles > Plan > Edit Charges**. The Edit Planned Charges dialog box opens.
5. Make the changes as required to the product bundle prices and click **OK**.

You have to activate the changes to make them visible in the current reference catalog. Refer to [Scheduling activation](#).

To see a new revision of bundle definitions, a catalog needs to 'Synchronize' the catalog to the revision. . Refer to [Synchronizing with reference catalog](#).

Price changes to existing products and bundles are not synchronized up to a channel catalog and have to be explicitly managed. When managing prices for a catalog (both products and bundles) the reference price of the product or bundle is visible to the product manager (in parentheses) in the product management UI.

Entitlement Behavior and Calculation

Unlimited entitlement:

Only those products can be given as unlimited entitlement whose usage types are generated by resourceType of the product bundle. Please check specific connector document to check which usage types are generated by a specific resourceType. Unlimited entitlements are always subscription scoped and cannot be used across subscriptions of a user.

Limited entitlement and resource isolation is true:

Entitlements are user scoped and all entitlements of a specific product are added up for a user.

Limited entitlement and resource isolation is false:

Entitlements are account scoped and all entitlements of a specific product are added up for an account.

Scheduling activation

When you make changes to your catalog such as adding new product bundles or products to replace existing ones or edit product or bundle pricing, these changes do not appear in the current reference catalog. You have to schedule the activation of the changes to update the reference catalog.

1. Log in to CloudPortal Business Manager as the Root user.
2. Click **Catalog > Plan Next > Schedule Activation**.
3. In the Set Plan Date dialog box, enter the date by when you want to make the catalog effective. On this date, the changes become effective and are visible in the reference catalog.
4. Click **OK**.

Channels

Channels represent groups of accounts. Every account is tied to a specific channel. Operators may choose to segment their accounts in any way they wish. Channels could represent geographies or they could represent market segments. Channels are associated with a catalog. A catalog contains channel specific prices for all the products in the system along with an operator determined set of bundles that are available to that catalog, which can also have channel specific prices. Consumers may purchase only the bundles available in the catalog that is configured for their channel. Operators have to explicitly add the set of bundles that are available to a catalog. This is done using the add bundle action. Only published bundles that are visible to the catalog can be added. Any changes made to a catalog have to be explicitly activated at which point it is visible to consumers associated with the corresponding channel.

Channel Catalog

Every channel has a catalog that lists the bundles being sold in that channel. You can add any number of bundles to a catalog.

Note: Only published bundles can be added in a channel catalog.

Channel Currencies

You may want to sell offerings only in a subset of currencies active in the system.

Note: You can add currencies that we want to support in the channel, but once the channel catalog is active, the currency can be supported only in the future revision.

Product and Bundle Prices

There are separate revisions of product and bundle prices in the channel catalog. When a channel is created, prices of all the active products are copied in the channel. As regards a bundle, the prices are copied over when you attach a bundle to the channel catalog. There is a separate price revision which is independent of the price revision in the reference catalog.

Channel Synchronizing

To see a new revision of product and bundle definitions and prices, a catalog needs to synchronize the catalog to the revision. This allows the catalog to see all the changes made in the definitions. Once synchronized, an operator may then override prices before activating a new revision of the catalog. Price changes to existing products and bundles are not synchronized with the channel catalog and have to be explicitly managed. When managing prices for a catalog (both products and bundles) the reference price of the product or bundle is visible to the product manager (in parenthesis) in the product management UI.

After synchronizing, retired products are marked retired in the channel catalog.

Note: Product Mediation Rules and Bundle Provisioning constraints are modified only in the reference catalog. If you synchronize, only then can you get the changes reflected in the channel catalog.

Adding a channel

You can add a channel to represent a market segment and serve a particular customer base. You can add channels from the CloudPortal Business Manager UI.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels > Add New**. The Add Channel wizard opens.
3. Specify the channel details and the currencies in which the products and product bundles will be available in the channel.
4. Review and confirm the channel details, click **Add** and then, click **Close**.

Editing a channel

You can modify channel details and the currencies that a channel supports from the CloudPortal Business manager UI.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels** and select the channel that you want to edit.
3. Place the cursor over the manage icon (that looks like a gear wheel) in the top right corner, and click **Edit Channel**.
4. Make the required changes.
5. Click **Save** and then click **Close**.

Deleting a channel

You can delete a channel when it is no longer serves a market segment and there are no customer accounts associated with the channel. You can delete a channel if no active tenants exist in the channel.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels** and select the channel that you want to delete.
3. Place the cursor over the manage icon (that looks like a gear wheel) in the top right corner, and click **Delete Channel**. This option is visible only if there are no active tenants in the channel.
4. Click **OK**.

Adding currencies to a channel

A channel can support multiple currencies. You may want to sell offerings only in a subset of currencies active in the system. As a service provider, you can specify the currencies in which the products and bundles are sold in a channel during channel creation. You can add other active currencies at any time.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels** and select the channel to which you want to add currencies.
3. In the Currencies tab, place the cursor over the manage icon (that looks like a gear wheel), and click **Add Currencies**.
4. Select the currencies and click **OK**.

Synchronizing with reference catalog

To see a new revision of product and bundle definitions, a catalog needs to 'Synchronize' the catalog to the revision. The latest version of reference catalog is then visible to the channel. On synchronization, any changes to bundle provisioning constraints get reflected in the channel catalog, retired products are not displayed in the channel catalog, and newly added bundles and pricing changes become visible to the channel.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels** and select the channel.
3. Click the **Catalog** tab in the details pane.
4. Place the cursor over the manage icon (that looks like a gear wheel), and click **Sync with Reference Price Book**.
5. Click **OK**.

Note: Price changes to existing products and bundles are not synchronized up to a channel catalog and have to be explicitly managed. When managing prices for a catalog (both products and bundles) the reference price of the product or bundle is visible to the product manager (in parentheses) in the product management UI.

Adding product bundles to a channel catalog

Each channel is created to serve a particular customer base and has its own catalog of product bundles that is specific to its customer requirements. You can add published product bundles to your channel catalog from the CloudPortal Business Manager UI.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels** and select the channel.
3. Click the **Catalog** tab in the details pane.
4. Place the cursor over the manage icon (that looks like a gear wheel), and click **Add Bundle(s) to Catalog**. You can add only published product bundles. You have to synchronize with the reference catalog to get the latest version of the reference catalog so that you can view any changes that have made to the reference catalog since the last synchronization.
5. Select the bundles that you want to add to the channel catalog and click **OK**.

Editing product and product bundle prices in channel catalog

Based on your business needs, you can provide specialized pricing on a per channel basis across multiple currencies. You can override the prices in the reference catalog on a per channel basis. When managing prices for a catalog (both products and bundles) the reference price of the product or bundle is visible to the product manager (in parentheses) in the product management UI.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click **Channels** and select the channel.
3. Click the **Catalog** tab in the details pane.
4. To edit product prices, click **Utility Rate Card**. To edit product bundle prices, click **Bundles** and select the bundle.
5. Place the cursor over the manage icon (that looks like a gear wheel), and click **Edit Pricing**.
6. Make the changes as required and click **OK**.

Scheduling activation

When you make changes to your channel catalog, you have to activate the changes so that the changes are visible to the customers of that channel. When you add bundles to your channel catalog or make any pricing changes, schedule activation to make the changes effective on the date specified.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels** and select the channel.
3. Click the **Catalog** tab in the details pane.
4. Place the cursor over the manage icon (that looks like a gear wheel), and click **Schedule Activation**.
5. Specify the date on which the revision of the catalog becomes effective and click **Proceed**.

Previewing the channel catalog

A channel catalog contains channel specific prices for all the products in the system along with an operator determined set of bundles that are available to that catalog, which can also have channel specific prices. As a service provider, you can view the product catalog that is specific to any channel. You can view the catalog in the same way as any user of that channel sees it.

1. Log in to CloudPortal Business Manager as Root user.
2. Click **Channels** and select the channel.
3. Click the **Catalog** tab in the details pane.
4. Place the cursor over the manage icon (that looks like a gear wheel), and click **Preview Catalog**. You can view the catalog for that channel.

Campaigns and Promotions

A campaign is a marketing program that you create in CloudPortal Business Manager to introduce new products to customers who sign up during the campaign period to expand the customer pool, or to retain customers by giving them additional offerings.

Trial users who sign up during the campaign period are given a promotional code that they use to sign up. This is a required code that trial users should provide. These customers are assigned a Trial account type and they can sign up by entering the promotional code in the text box provided in the Sign-up page.

Other users (Retail and Corporate users) who sign up during the campaign period can also enroll for campaigns and promotions that are on offer. However, for non-trial users, the promotional code is not required in the Sign-up page.

Note: Trial users avail themselves of a full discount by default (100%), whereas the discounts on offer for non-trial users are governed by rules set by an administrator.

When the trial period expires, the account owner is notified by Email. The account's resources like running VMs, templates, snapshots, storage etc. are stopped. These resources are released into the system after a certain period of time, which can be configured.

Adding a campaign

Pre-requisite: You must have the Product Create/Manage permission to add a new campaign.

1. Log in to CloudPortal as the Root User or Ops Admin.
2. Click **Channels > Campaigns**.
3. To add a campaign, click Add New Campaign. The Add Campaign page is displayed.
4. Update the fields with the relevant details.
5. Specify the promotion code for the campaign.
6. Specify the start date and end date of the campaign. The end date decides when the campaign ends. No user can register after this date. However, if the end date is not specified, the campaign will run forever.
7. Check the **Trial** option, if you want this campaign to be eligible **only** for trial users. If you leave the Trial option unchecked, it implies that this campaign is exclusively for non-trial users.

Note: Percent Off and Amount Off options appear only when the Trial option is unchecked. If the Trial option is checked, it indicates that the campaign is exclusively for trial users, and as such the campaign offers a 100% discount.
8. Check the **Unlimited Signups** option if the campaign is not limited only to a specific number of active users.
9. Check the **Unlimited Duration** option if the promotion offer is available forever to a registered trial user.
10. If you do not select the Unlimited Duration option for the campaign, specify the **Duration of entitlement** of the campaign. This denotes the duration of the entitlement available through this particular campaign and is specified in the number of days for a trial user and number of billing periods for a non-trial user.
11. Select the channels in which you want this campaign to feature. To ensure this campaign features in multiple channels, press Ctrl key while selecting the channels.
12. For non-trial users, if you want to give a discount of a certain percentage on this campaign, select the **Percent Off** option and enter the percentage in the **Percent** text box.
13. For non-trial users, instead of a flat percentage off, you might want to give different amounts as discounts across different currencies. Select the **Amount Off** option and specify the amount of discount that you want to give in various currencies.
14. Click **Submit**.

Understanding Billing in CloudPortal Business Manager

In CPBM 2.0, the billing engine and book keeping model was fully redesigned to make it more accurate and easy to understand in addition to supporting some additional capabilities. This section describes the billing and book keeping module for CPBM 2.x.

Billing Periods
Account Statement
Invoices
Accounting for Entitlements
Subscriptions
Instant Payment

Billing Periods

From a billing perspective, CPBM reconciles its books every billing period. Depending on an implementation's need, the definition of a billing period can be changed. CPBM supports three models for defining billing periods. There are two configuration properties in CPBM that control this behavior, namely `billingMode.type` and `billingMode.param`. The billing mode determines which model CPBM uses to compute billing periods. The `billingMode.param` is interpreted differently depending on the `billingMode.type`. The different billing mode types are:

- **FIXED_DAYS:** This model defines a billing period as a fixed number of days. The `billingMode.param` in this case determines the number of days that make a billing period. This is counted from the 'anniversary date' of an account, which is the date on which the account is registered. CPBM will then compute every billing period as `x` (determined by `billingMode.param`) number of days from the end of the last billing period (or from anniversary date for the first time). The common value for `billingMode.param` is between 28 and 31.
- **FIXED_DATE:** This model defines a billing period as a calendar month ending the day before a fixed date every month. The `billingMode.param` in this case is the date on which the billing period is reconciled (in other words, the books are closed for the month ending the day before). For example, if `billingMode.param` is 1 (a common setting), then the account statements are posted on the 1st of every month. In the fixed date model, the `billingMode.param` cannot be greater than 28.
- **ANNIVERSARY_DATE:** This model is a variation on the **FIXED_DATE** model and defines a billing period as a calendar month ending the day before a fixed date every month, but the date being determined by the anniversary date (that is the date the customer registered) of each customer independently. This means customers who registered on the 5th will have their billing periods end on the 4th of every month, but a customer who registers on the 12th will have the billing period end on the 11th of every month. Customers who register on dates after the 28th get reset to the 28th starting the second month. In this model, the `billingMode.param` is ignored.

For all practical purposes, CPBM uses the term **MONTH** to be synonymous with billing period, for user friendliness.

Account Statement

Every billing period, CPBM posts an account statement that provides information about the current state of the account from a billing perspective. In the 'current' billing period, CPBM attempts to keep an accurate record of the user's spend on a daily basis. However, until the account statement is posted, the information in the current account statement should be considered subject to modification. In this state, the account statement is considered to be in a PROVISIONAL state. Once posted, the account statement is in a POSTED state. All POSTED account statements are read only and will not be modified by the system ever again.

Balance Forward

Every month, an account statement is initialized in a PROVISIONAL state with its opening balance the same as the Net Balance of the previous month's POSTED account statement, essentially carrying the previous account statement's balance forward. This amount is represented as the *Balance Forward*. This is always guaranteed to be the same as the previous month's posted account statement.

The only exception to this rule is that on the date of invoicing, there is a lag between midnight and the time the account statement is computed and posted, during which the account balance of the current month will not reflect the last day's charges. This will be corrected once the previous account statement is posted.

Payments and Credits

All payments made and service credits issued to the customer are recorded and aggregated in this section. Automated payments for the previous account statement will be credited in the current account statement, as payments are made against posted account statements only.

Charges

All charges to the account are accrued and aggregated in this section. All accrued charges are recorded as invoices in CPBM.

Invoices

Invoices have invoice line items that detail the charges that make up the invoice. In addition, each invoice, when posted will show the discounts applied and the taxes owed for that charge. The number of discount line items and the number of tax line items that are presented in the invoice will depend on the discount and tax engines configured in the system and may produce zero or more line items. Because utility prices for cloud resources can sometimes be less than the smallest currency unit (for example, microcent pricing in USD, such as 0.0032 USD per GB of bandwidth), invoice line items will be based on the appropriate precision as determined by the pricing of the purchased product. However, the billing system works only in currency precision. Therefore, all sub totals (charges, total tax, total discount, and net amount) of an invoice are in currency precision only. How these subtotals are rounded (up, down, ceiling, floor, and so on) is dependent on configuration and can be modified. The default rounding in the system is HALF_UP, which means 0.0050 and above will get rounded to 0.01.

There are three kinds of invoices. They are described in the following sections.

Subscription invoices

These are invoices for any user subscriptions to bundles. These are categorized by subscription type and are tied to the subscription and the user that owns the subscription.

In 1.x, CPBM used to create charge line items for each subscription in the invoice for the month. In 2.x, CPBM correctly creates invoices for each subscription, which details the charges for each subscription. In addition, CPBM creates an invoice for each subscription's renewal separately.

Close Out invoices

Close out invoices are posted every day and record utility spend (i.e, spend for any usage that are not covered by entitlements provided by the subscriptions to bundles) on a per user basis. Note that these invoices are also tied to specific users. In other words, there will be one close out invoice per day per user in the tenant. This allows the system to accurately record the spend of each user.

In 1.x, CPBM used to accrue only month to date information on per product basis. When consolidated at the end of the month, CPBM would apply taxes and discounts to each product's usage for the month. In 2.x, in order to ensure accurate daily spend and to ensure that daily pricing was reflected in the charges, CPBM moved to a model of tracking daily spend. Due to this, close out invoices are now recorded on a daily basis, which means while it is still possible to get actual month to date usage of a given product, that amount does not account for discounts and taxes. Instead discounts and taxes are applied on a daily basis for the aggregate of all utility spend for the day.

Debit Notes

Debit notes are applied whenever a simple amount (i.e, no taxes, discounts, and so on) need to be charged to the account. This can happen for various reasons, but out of the box, debit notes are created only when a payment is cancelled. In CPBM, you can cancel any past credit (automated, recorded, manual, or service credits). It is important to note that while payments and credits for past account statements can be cancelled, the debit note will always appear as a charge in the current account statement. When automated payments to credit cards are cancelled, they are also charged back to the credit card.

Renewal Charges

In addition to the invoices mentioned, invoices for any renewals that are due in the current billing period are created at the beginning of the billing period (or immediately in case the subscription itself occurred in the middle of this billing period). These renewal invoices are posted when they become due. Monthly renewals are due at the end of the previous month, and therefore get posted just before the account statement is posted. Renewal invoices for all other renewal periods (such as quarterly or annual) are posted on the day they are due. Renewal invoices will be cancelled if the subscription is cancelled before the renewal becomes due. Once posted, the customer owes the amount of this invoice and this cannot be revoked or canceled. For example, if you subscribed for a quarterly subscription on the 15th of January (assuming fixed date billing mode, with date set to the first of every month), in the beginning of March, that is March 1, the system will create the renewal invoice for that subscription. That invoice will be posted on the 15th (i.e, as of the 14th). Cancelling the subscription before March 15th will cancel the renewal invoice, but canceling it on the 15th and beyond will not. The customer will owe the amount after that point (even if the actual payment will be done after the account statement is posted on April 1).

This allows the customer to see the renewal charges that will be posted to the account in any given month, even if they only become due later, so they have an opportunity to cancel the subscription.

Accounting for Entitlements

Entitlements are accounted either to usage accrued to a given subscription or in a consolidated manner at the user level. Whether usage is accrued to a subscription or to the user is dependent on the specific product being entitled and the specific type of the bundle being subscribed to. For example, VM running time, can be accrued to a virtual machine, but data transfer is usually not attributable to the virtual machine as it shares the network on which the traffic is being generated and accounting for data transfer is done at the router. So, when purchasing a virtual machine, data transfer entitlements in the associated bundle will be added to the user's overall entitlements for data traffic, whereas when purchasing a network, the data transfer on that network will be tied specifically to that subscription.

Note: This above model of accruing entitlements is different from CPBM 1.x, where entitlements were accrued at the account level and not on a per user basis. In 1.x this meant that unused entitlements from one user's subscriptions spilt over to other users. That implied, that CPBM could not keep accurate track of individual spend of a given user, as it got diluted with entitlement data from other users.

Price Selection

Since prices can be revisioned in CPBM, it is important for CPBM to charge appropriately based on pricing changes. For all utility pricing, CPBM will pick up the latest prices on a daily basis to apply to any utility charges. This allows operators to change the prices and have the updated prices passed on to the consumer by the next day. For all subscriptions, pricing will be the latest price on the date the renewal invoice is posted. Note that this means that if the prices change between renewal invoice generation and posting, the new price will be reflected at the time of posting (but at the time of posting only).

Prorating

For monthly invoices only, subscription charges are always prorated when subscriptions are created to pay only for duration of the month that the subscription was active. This is done at the granularity of a single day. In addition, all entitlements will also be prorated accordingly.

Subscriptions

Terminating Subscriptions

Subscriptions can be terminated. When a subscription is terminated, there is no proration of charges, and no credit is given. This is because subscriptions are charged in advance for a committed period of time. However, since the subscription is fully paid for that period, all entitlements (except unlimited entitlements that are dependent on the resource existing) that are eligible for that period continue to be accounted for during billing.

Changing Subscriptions

Subscriptions can be changed to shift from one bundle to another. In this case, the system will treat this action as terminating the first subscription and subscribing to the new subscription. Note that this means that while the new subscription will be prorated for the remainder of the month, if monthly and there will be no credit for unused entitlements from the original bundle. However, as before, entitlements will also be provided based on what the original bundle provided in addition to any prorated entitlements that will be available from the new bundle.

Instant Payment

When instant pay is turned on for any account type, the first invoices for any bundle subscription will be paid for immediately. This is applicable only to account types that require payment information that can be used to automatically charge the customer. These payments will be accrued in the current account statement as well since it was made in the current billing period.

BSS API Guide

The BSS API Guide is a task-oriented explanation of how to write, preprocess, compile, and link database applications using the BSS Applications Programming Interface (API).

In This Section

Intended Audience
Overview of BSS API
REST Architecture
BSS API data model
Simulating PUT and DELETE
Custom Actions
Requesting data formats
Standard query parameters
Root API path and versioning
Sample responses
Localization
Roles
Using the API
Resources
Known issues

Intended Audience

This guide is for the software developers who want to develop customer portal for CPBM that can interact with BSS API.

To use this guide, you should have a working knowledge of the following subjects:

- Knowledge of JSON and/or XML data serialization formats
- Knowledge of a programming language for example, Java
- Knowledge of REST web services

Overview of BSS API

Citrix® -CloudPortal™ Business Manager Business Support System (BSS) API (henceforth referred to as the BSS API) is a platform that allows developers to create new management solutions or integrate existing systems with CloudPortal. The BSS layer focus towards customers and business partners. This BSS API provides a ReSTful (Representational State Transfer) web service interface. The BSS API supports four possible actions (Get, Put, Post, and Delete) on exposed resources and supports responses in either XML or JSON response formats. This API represents the set of business-related services exposed by the CPBM that are used to implement a cloud service.

REST Architecture

Representational State Transfer (REST) is an architecture that provides a consistent approach to request and modify the data. In this architecture, resources are stored in a data store, and when the client sends a request to the server for a particular action, the server performs the action and sends a response in the form of a representation of the specified resource. It is possible to invoke the API via HTTP or HTTPS through a certified authentication.

BSS API data model

A resource is an individual data entity with a unique identifier. The BSS API follows the conventions detailed below:

- All APIs are written in lower case.
- For a given resource, the API will reference the plural form of the word. For example, for resource Account, all APIs will reference it as accounts.
- The date value arguments will expect value in ISO 8601 format i.e., YYYY-mm-ddTHH:mm:ss.[mmm]TZ. Please refer the following document <http://www.w3.org/TR/NOTE-datetime>

Supported Operations

Operations are the actions you perform within your account. Creating, deleting or updating, retrieving. Operations are performed through the ReST web service API.

For every resource exposed, the base URL is fixed to be the name of the resource (in plural) and for referring to single entities of a given resource type, the base URL is appended with the UUID of that resource. For example, given a resource Order. The base URL will be:

`/orders`

And for a single order, the base URL will be:

`/orders/<uuid>`

Operation	<code>/orders</code>	<code>/orders/<uuid></code>
GET	Gets all the orders	Get this order
POST	Create new order	Create this order (Not supported)
PUT	Update all orders (Not supported)	Update this order
DELETE	Delete all orders (Not supported)	Delete this order

Nested Resources

For nested resources, the base URL of that resource should be appended to the parent resource base URL.

For example, for line items of order ID 32 the base URLs will be

```
/orders/32/line_items
```

```
/orders/32/line_items/21
```

Simulating PUT and DELETE

When using browsers or other clients that do not support the PUT and DELETE methods, the parameter `_method` may be used along with a POST request to request the server to treat the request as a PUT or DELETE.

For example, to simulate a delete of order 21 you would use:

```
POST/orders/21?_method=DELETE
```

Custom Actions

Custom actions are avoided as much as possible. However, certain operations are exposed as custom actions. Custom actions on a given resource are appended to the base URL and are documented along with the details of each Resource.

For example, say an execute action for an Order 23 will be represented as

`/orders/23/execute`

Requesting data formats

By default, the BSS APIs will return JSON. To request a specific format, the API call can be appended with a suffix, currently only `.json` and `.xml`, which will return JSON and XML respectively. Alternatively, the BSS API will also support the `Content-Type` header to set format. When both `Content-Type` and the suffix are provided, the suffix will take precedence.

Standard query parameters

Where supported, the API uses standard query parameters to support search, filter, pagination and expansion. All APIs will use these parameters when they support the specified functions.

These parameters are detailed in the table below:

Parameters	Description
q	search string for a search, typically used in the list calls (e.g. GET /orders)
page	Used to paginate, represents page number
pagesize	No. of items to return per page
expand	Request expansion of nested entities (Expand is optional comma separated list of one or more inner objects of REST resource)
order	asc desc. Order by specified field in ascending / descending order
<name>=<value>	Filter by field specified by name with value provided.
sort	<field>, field to sort

Example for expand

Suppose we have object "foo", which contains list of "bar" Objects. foo object expanded view:

name

fooid

uuid

bar

bar Object expanded view

bar Object expanded view

name (minimal)

displayname

address

pincode

country

emailed (minimal)

phone (minimal)

address Object expanded view

name (minimal)

street1

street2

city (minimal)

state

Response for the query listFOO, gives the expanded representation of foo, but minimal representation of bar object.

In request query, if there is no expand, such as `http://<IP:Port/api/listFOO ?>`, then the response will be:

```
{
  "count":1,
  "foolist":[
    {
      "uuid":"9fc7754c-6d46-11e0-a026-065287aed31a",
      "name":"SERVICE",
      "fooid":"00000000",
      "barlist":[
        {
          "name":"SYSTEM",
          "emailid":"abc@xyz.com",
          "phone":"213412341"
        }
      ]
    }
  ]
}
```

In request query, if `expand = bar`, such as `http://<IP:Port/api/listFOO?expand=bar >` then the response will be :

```
{
  "count":1,
  "foolist":[
    {
      "uuid":"9fc7754c-6d46-11e0-a026-065287aed31a",
      "name":"SERVICE",
      "fooid":"00000000",
      "barlist":[
```

```
{
  "name":"SYSTEM",
  "displayname":"System",
  "address":{
    "city":"cityname"
  },
  "pincode":"213123",
  "Country":"someCountry",
  "emailid":"abc@xyz.com",
  "phone":"213412341"
}
]
```

In request query, if Address along with "bar" needs to be expanded, then the query should contain `expand = bar.address`, such as

< `http://<IP:Port/api/listFOO?expand=bar.address` >, then the response will be:

```
{
  "count":1,
  "foolist":[
    {
      "uuid":"9fc7754c-6d46-11e0-a026-065287aed31a",
      "name":"SERVICE",
      "fooid":"00000000",
      "barlist":[
        {
          "name":"SYSTEM",
          "displayname":"System",
          "address":{
            "stree1":"1st street",
            "street2":"street2",
            "city":"cityname",
            "state":"state"
          },
          "pincode":"213123",
          "Country":"someCountry",
          "emailid":"abc@xyz.com",
          "phone":"213412341"
        }
      ]
    }
  ]
}
```

Multiple expand:

Suppose "qux" is another object in foo along with "bar", that can be expanded as well, then the query should be `expand=bar,qux` as in:

< `http://<IP:Port/api/listFOO?expand=bar,qux` >

Root API Path and Versioning

The root path will be `/api/orders` which in turn hit the latest version of the API. All earlier versions will be available as `/api/<version no>/orders`.

APIs will have only one major revision number. All changes within a given version will remain backward compatible. Backward compatible changes include:

1. Addition of new entities and custom actions.
2. Support for additional verbs for resources that previously did not support them.
3. Addition of new fields in responses.
4. Mandatory request parameters may be made optional.

Sample Responses

BSS API supports two formats as the response to an API call. The default response is JSON. To request a specific format, the API call can be appended with a suffix, currently only .json and .xml which will return JSON and XML respectively.

Customization of date format is also supported and this can be done by setting the date format from the Administration > Configuration > Portal > Server page.

Sample JSON response:

Request URL:

`http://10.102.153.50:8080/portal/api/accounts?expand=owner&q=retail&page=1&pagesize=5&sort=name&order=asc&_=<timestamp>&apiKey=<apikey>&signature=<generatedSignatureUsingSecretKey>` or `http://10.102.153.50:8080/portal/api/accounts.json?expand=owner&q=retail&page=1&pagesize=5&sort=name&order=asc&_=<timestamp>&apiKey=<apikey>&signature=<generatedSignatureUsingSecretKey>`

Response:

```
{
  "accounts": [
    {
      "uuid": "6477e245-db16-4863-9b2f-aef2cb984d21",
      "name": "abhait@retail",
      "accountId": "AA000011",
      "state": "ACTIVE",
      "sourceChannel": {
        "uuid": "0ed2ca51-2421-4df7-971e-70d7a2d3321d",
        "name": "channel",
        "code": "channel",
        "description": "channel"
      },
      "accountType": {
        "name": "RETAIL",
        "displayName": "Retail",
        "description": "Retail customer type. Credit card will be charged automatically when payments are due"
      },
      "spendLimit": 1000,
      "initialDeposit": 0,
      "spendBudget": 0,
      "owner": {
        "firstName": "Abhai",
        "email": "k.abhai@citrix.com",
        "lastName": "Kumar",
        "phone": "91-9379682039",
        "username": "abhait@retail",
        "uuid": "3651d8ba-ffdd-46ec-88ae-22b70ff0c204",

```

```
"address":{
  "city":"sbc",
  "country":"IN",
  "postalCode":"560075",
  "state":"KAR"
},
"apiKey":"XUyCQVb7S_WOGsDiEBG1wqNKyCqJmk6e97ZnOycU8-8HEaJjqztEdZOpXF2cb1EOzZ-Vr1423i",
"countryCode":"91",
"gravatarUrl":null,
"profile":{
  "name":"Master User",
  "description":"Master user"
},
"enabled":true,
"secretKey":"3FRXeqBtzu61aoGEVDPxyK4T1MXAE8dwJxb8TwX7qld8BccrKgfwlqFOTFy9iLTyM-ZTDntXtD",
"spendBudget":0,
"title":null,
"timeZone":"GMT"
},
"address":{
  "city":"sbc",
  "country":"IN",
  "postalCode":"560075",
  "state":"KAR"
},
"secondaryAddress":null,
"currency":{
  "currencyCode":"USD",
  "currencyName":"US Dollar"
},
"memo":null,
"trialAccount":null,
"tenantExtraInformation":{
  "anniversaryDate":"24 Apr 2013 12:00:00 UTC",
  "nextInvoiceDate":null,
  "discountPercent":"0.0000"
},
"deposit":null,
"remoteAddress":"10.252.240.188",
"removed":null,
"removedBy":null,
"imagePath":null,
"faviconPath":null,
"usernameSuffix":null,
"users":[
  {
    "firstName":"Abhai",
    "lastName":"Kumar",
    "username":"abhai@retail",
    "uuid":"3651d8ba-ffdd-46ec-88ae-22b70ff0c204"
  },
  {
    "firstName":"Normal",
    "lastName":"Kumar",
    "username":"retail@normal",
    "uuid":"a2856513-b3bd-4034-b9a2-5bc4fe74164d"
  }
]
```

```
    },
    {
      "firstName":"Power",
      "lastName":"Kumar",
      "username":"retail@power",
      "uuid":"48f123c4-4ece-425f-a320-73f212262fca"
    },
    {
      "firstName":"Billing",
      "lastName":"Kumar",
      "username":"retail@billing",
      "uuid":"b4c0a44e-3060-4e5c-b290-3ea37329ae3a"
    }
  ]
},
{ "accountId":"AA000001",
  "state":"ACTIVE",
  "sourceChannel":{
    "uuid":"0ed2ca51-2421-4df7-971e-70d7a2d3321d",
    "name":"channel",
    "code":"channel",
    "description":"channel"
  },
  "accountType":{
    "name":"RETAIL",
    "displayName":"Retail",
    "description":"Retail customer type. Credit card will be charged automatically when payments are du
  },
  "spendLimit":1000,
  "initialDeposit":0,
  "spendBudget":0,
  "owner":{
    "firstName":"Retail",
    "email":"d.aditya@citrix.com",
    "lastName":"Account",
    "phone":"91-9886319282",
    "username":"Retail",
    "uuid":"0f1a4222-7d96-4d7e-946a-6cfb05e57200",
    "address":{
      "city":"Bangalore",
      "country":"IN",
      "postalCode":"560001",
      "state":"KAR"
    },
    "apiKey":"retailapi123",
    "countryCode":"91",
    "gravatarUrl":null,
    "profile":{
      "name":"Master User",
      "description":"Master user"
    },
    "enabled":true,
    "secretKey":"retailsecret123",
    "spendBudget":0,
    "title":null,
    "timeZone":"GMT"
```

```

    },
    "address":{
      "city":"Bangalore",
      "country":"IN",
      "postalCode":"560001",
      "state":"KAR"
    },
    "secondaryAddress":null,
    "currency":{
      "currencyCode":"USD",
      "currencyName":"US Dollar"
    },
    "memo":null,
    "trialAccount":null,
    "tenantExtraInformation":{
      "anniversaryDate":"17 Apr 2013 12:00:00 UTC",
      "nextInvoiceDate":null,
      "discountPercent":"0.0000"
    },
    "deposit":null,
    "remoteAddress":"10.252.249.72",
    "removed":null,
    "removedBy":null,
    "imagePath":null,
    "faviconPath":null,
    "usernameSuffix":null,
    "users":[
      {
        "firstName":"Retail",
        "lastName":"Account",
        "username":"Retail",
        "uuid":"0f1a4222-7d96-4d7e-946a-6cfb05e57200"
      }
    ]
  },{"..."},"..."},"..."},
]
}

```

Sample XML response:

Request URL:

http://10.102.153.50:8080/portal/api/accounts.xml?expand=owner&q=retail&page=1&pagesize=5&sort=name&order=asc&_=<timestamp>&apiKey=<apikey>&signature=<generatedSignatureUsingSecretKey>

Response:

```

<?xml version="1.0"?>
-<root>
-<accounts>
-<account>
<uuid>6477e245-db16-4863-9b2f-aef2cb984d21</uuid>
<name>abhair@retail</name>

```

```
<accountId>AA000011</accountId>
<state>ACTIVE</state>
-<sourceChannel>
<uuid>0ed2ca51-2421-4df7-971e-70d7a2d3321d</uuid>
<name>channel</name>
<code>channel</code>
<description>channel</description>
</sourceChannel>
-<accountType>
<name>RETAIL</name>
<displayName>Retail</displayName>
<description>Retail customer type. Credit card will be charged
automatically when payments are due.</description>
</accountType>
<spendLimit>1000.0000</spendLimit>
<initialDeposit>0.0000</initialDeposit>
<spendBudget>0.0000</spendBudget>
-<owner>
<firstName>Abhai</firstName>
<email/>
<lastName>Kumar</lastName>
<phone>91-9379682039</phone>
<username>abhai@retail</username>
<uuid>3651d8ba-ffdd-46ec-88ae-22b70ff0c204</uuid>
-<address>
<city>sbcb</city>
<country>IN</country>
<postalCode>560075</postalCode>
<state>KAR</state>
</address>
<apiKey>XUyCQVb7S_WOGsDiEBG1wqNKyCqJmk6e97ZnOycU8-
8HEaJjqztEdZOpXF2cb1EOzZ-
Vr1423iNajNJYV2gXA</apiKey>
<countryCode/>
<gravatarUrl/>
-<profile>
<name>Master User</name>
<description>Master user</description>
</profile>
<enabled>>true</enabled>
<secretKey>3FRXeqBtz61aoGEVDPxyK4T1MXAE8dwJxb8TwX7ql
d8BccrKgfwlqFOTFy9iLTyM-ZTDntXtDHaeNr8t6u00w</secretKey>
<spendBudget>0.0000</spendBudget>
<title/>
<timeZone>GMT</timeZone>
</owner>
-<address>
<city>sbcb</city>
<country>IN</country>
<postalCode>560075</postalCode>
<state>KAR</state>
</address>
<secondaryAddress/>
-<currency>
<currencyCode>USD</currencyCode>
<currencyName>US Dollar</currencyName>
```

```
</currency>
<memo/>
<trialAccount/>
-<tenantExtraInformation>
<anniversaryDate>24 Apr 2013 12:00:00 UTC</anniversaryDate>
<nextInvoiceDate/>
<discountPercent>0.0000</discountPercent>
</tenantExtraInformation>
<deposit/>
<remoteAddress>10.252.240.188</remoteAddress>
<removed/>
<removedBy/>
<imagePath/>
<faviconPath/>
<usernameSuffix/>
-<users>
-<user>
<firstName>Abhai</firstName>
<lastName>Kumar</lastName>
<username>abhai@retail</username>
<uuid>3651d8ba-ffdd-46ec-88ae-22b70ff0c204</uuid>
</user>
-<user>
<firstName>Normal</firstName>
<lastName>Kumar</lastName>
<username>retail@normal</username>
<uuid>a2856513-b3bd-4034-b9a2-bc4fe74164d</uuid>
</user>
-<user>
<firstName>Power</firstName>
<lastName>Kumar</lastName>
<username>retail@power</username>
<uuid>48f123c4-4ece-425f-a320-73f212262fca</uuid>
</user>
-<user>
<firstName>Billing</firstName>
<lastName>Kumar</lastName>
<username>retail@billing</username>
<uuid>b4c0a44e-3060-4e5c-b290-3ea37329ae3a</uuid>
</user>
</users>
</account>
+<account>
+<account>
+<account>
</accounts>
</root>
```

Localization

The BSS API will respect the Content-Language header and return any human readable data in the appropriate language.

Note: This is currently not supported, responses will be in default language entered, but it is a good practice to pass the appropriate Content-Language header, for when the API will support this feature.

Roles

The following roles are supported:

- **ROLE_ACCOUNT_CRUD** (Account create): This role has the ability to create, read, update, and delete any customer account.
- **ROLE_ACCOUNT_USER_CRUD** (Account User create manage): This role can create, read, update, and delete a user in his/her own customer account only.
- **ROLE_PROFILE_CRUD** (Profile create manage): This role can create, read, update, and delete profiles on the CloudPortal.
- **ROLE_ACCOUNT_MGMT** (Account manage): This role has the ability to view and update information of any customer account.
- **ROLE_ACCOUNT_BILLING_ADMIN** (Billing Admin manage): Tenant Billing Admin.
- **ROLE_FINANCE_CRUD** (Finance create manage): This role can create, read, update, and delete financial information of any customer account. For example, make payment, issue credit, and so on.

Using the API

Before you begin to use the BSS API, you should have:

- Access to the CloudPortal server you wish to integrate with.
- Both the API Key and Secret Key for a user

Authentication

Every REST request against the BSS API requires the inclusion of a specific authorization through API Key and Secret Key and timestamp. The authentication process requires various HTTP Header parameters to be submitted with a request.

Note: Timestamp (T) is system current time in milliseconds. By default, timestamp is valid for 5 minutes. This can be configured on a need basis.

Requests that need security have to include a signature. This signature changes with every requests and includes elements from the request.

User needs to pass the APIkey & append the encoded signature (using secret key & parameters) to the generated URL.

Every API request has the following format: Base URL+Base API Path+ REST API Path+Parameter String+Signature

where,

Base URL: Base URL to the CPBM Server. `http://localhost:8080`

Base API Path: Path to the base API Servlet that processes the incoming requests.
`/portal/api`

REST API Path: Path to CPBM rest api controller. `/foo`

Generating the signature

The signature is built by applying a keyed-HMAC (Hash Message Authentication Code) on parameter string prefixed by REST API path. The secretKey is provided as a parameter to the keyed-HMAC method.

1. For each field-value pair (as separated by a '&') in the Parameter String, URL ,encode each value so that it can be safely sent via HTTP GET.

Note: Make sure all spaces are encoded as "%20".

2. Lower case the entire Parameter String and sort it alphabetically via the field for each field-value pair. =1368420672402&apikey=mivr6x7u6bn_sdahobpjnejpgest35exq-jb8cg20yi3yaxxcgpyuairmfi_ejtvwz0nukkjbpm3y2bcikwfg
3. Prefix the parameter string with the REST API path as shown below: /foo_=1368420672402&apikey=mivr6x7u6bn_sdahobpjnejpgest35exq-jb8cg20yi3yaxxcgpyuairmfi_ejtvwz0nukkjbpm3y2bcikwfg
4. Take the prefixed Parameter String and run it through the HMAC SHA-1 hashing algorithm with the user's Secret Key.

After reconstructing, the final URL looks like:

```
http://localhost:8080/portal/api/foo? _=1368420672402&apiKey=mivr6x7u6bn_sdahobpjnejpgest35exq-jb8cg20yi3yaxxcgpyuairmfi_ejtvwz0nukkjbpm3y2bcikwfg&signature=Cxx1DN401BjmXU%2Fcaik8RAPo02xU%3D
```

Sample code to generate the signature:

```
public String signRequest(String parameterString, String SecretKey) {
    try {
        Mac mac = Mac.getInstance("HmacSHA1");
        SecretKeySpec keySpec = new SecretKeySpec(SecretKey.getBytes(), "HmacSHA1");
        mac.init(keySpec);
        mac.update(parameterString.getBytes());
        byte[] encryptedBytes = mac.doFinal();
        return URLEncoder.encode(Base64.encodeBytes(encryptedBytes), "UTF-8");
    } catch (Exception ex) {
        logger.debug(ex);
    }
    return null;
}
```

Channel: While it is possible to invoke the API via HTTP or HTTPS, it is strongly recommended that it be executed only through HTTPS with mutual certificate based authentication. This allows the CloudPortal Business Manager to establish a trusted link with the calling portal.

Resources

The following sections detail the exposed data structure view of the resources and supported actions on these resources. All the resources support an expanded view and minimal view of the data structure. The expanded view is an optional comma separated list of one or more inner objects of the REST resource.

Accounts
User
Subscriptions
AccountStatements
Invoices
Payments
Cloud Services
Catalogs
Events
Service health status
Service Maintenance activities
Channels
Currencies
Usage
Profiles
Account types
Countries

Accounts

Accounts can be accessed by the Root/Service provider users having access to create/update/delete/list.

Expand view

- uuid(minimal)
- name(minimal)
- accountId(minimal)
- state(minimal)
- spendLimit
- initialDeposit
- spendBudget
- syncBillingAddress
- memo
- paymentAlertsCounter
- accountSuspensionCounter
- remoteAddress
- removed
- removedBy
- imagePath
- faviconPath
- usernameSuffix
- customFieldMap
- isManualActivation
- maxUsers
- sharedCloudAccount
- tenantHandles

- users
- allUsers
- debugBilling
- sourceChannel (minimal)
 - uuid(minimal)
 - name(minimal)
 - description(minimal)
 - code(minimal)
 - type
 - imagePath
 - removed
 - removedBy
 - channelHandles
 - catalog
 - name
 - description
 - code
 - createdBy
 - updatedBy
 - supportedCurrencies(minimal)
 - removed
 - removedBy
 - channel
- accountType
 - name(minimal)
 - displayName(minimal)
 - description(minimal)
 - paymentModes
 - trial

- paymentInfoRequired
- autoPayRequired
- instantpayRequired
- extendCredit
- selfRegistrationAllowed
- manualRegistrationAllowed
- notionalBilling
- manualActivation
- creditExposureBreach
- defaultSelfRegistered
- defaultRegistered
- enableSecondaryAddress
- maxUsers
- accountRestrictionGracePeriod
- preAuthRequired
- depositRequired
- accountTypeCreditExposureList
- owner (minimal)
 - uuid(minimal)
 - firstName(minimal)
 - * lastName(minimal)
 - * email(minimal)
 - * phone(minimal)
 - * username(minimal)
 - * title
 - * address
 - * profile
 - * spendBudget

- * currentUsage
- * spendLimit
- * customFieldMap
- * locale
- * spendAlertSubscriptions
- * countryCode
- * apiKey
- * secretKey
- * userHandles
- address
 - city(minimal)
 - * state(minimal)
 - * postalCode(minimal)
 - * country(minimal)
 - * street1
 - * street2
- secondaryAddress
 - city(minimal)
 - * state(minimal)
 - * postalCode(minimal)
 - * country(minimal)
 - * street1
 - * street2
- currency
 - currencyCode(minimal)
 - currencyName(minimal)
 - rank
 - longCurrencyName
 - sign

- altSign
- flag
- fractionalUnit
- fractionalUnitActive
- numberToBasic
- active
- conversionFactor
- trialAccount
 - state(minimal)
 - expiryDate(minimal)
 - lastChange
- tenantExtraInformation
 - anniversaryDate(minimal)
 - nextInvoiceDate(minimal)
 - discountPercent(minimal)
 - paymentMode
 - paymentInfo
 - paymentInfoEffectiveChangeDate
- deposit
 - receivedOn(minimal)
 - amount(minimal)
 - tenant
 - recordedBy

Minimal View

- uuid
- name

Operations	Path	Method	Sort parameter	q parameter
List	/accounts	GET	Anything	uuid, name, accountId, memo, usernameSuffix

Create	/accounts	POST		
Get	/account	GET		
Get(Via Root)	/accounts/<uuid>	GET		
Update	/account	PUT		
Update(Via Root)	/accounts/<uuid>	PUT		
Delete	/accounts/<uuid>	DELETE		
Change State	/accounts/<uuid>/change/state	POST		
Change Account Type	/accounts/<uuid>/change/account/type	POST		

Parameters are optional unless specified as required.

List

Response Tags: minimal representation of accounts

Error Codes:

Role: Account manage or Account create

Request Parameters	Required/Optional
expand	
page (default=1)	
pagesize(default=10)	
q (Free text searchable query on 'uuid', 'name', 'accountid', 'usernameSuffix' and 'memo' fields)	
sort	
order (asc, desc)	
state {NEW,ACTIVE,LOCKED,SUSPENDED,TERMINATED}	
sourcechannelid	
accounttypeid {3,4,5 or any valid custom account type id}	
currencycode (eg: INR)	
suffix	
accountid	
name	
ownername	
charrange	
pattern	

count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	
---	--

Note: API behavior: When a service provider user requests to list accounts using /accounts, then all accounts are listed except the service provider user account.

Create

Response Tags: minimal representation of object

Error Codes:

Role: Account create

Request Parameters	Required/Optional
accountname	Required
sourcechannelid	Required
accounttypeid (3,4,5 or any valid custom account type id)	Required
currencycode (eg: INR,USD)	Required
locale (eg. en_US)	
usernameprefix	Required if duplicate username is enabled. Required in case of same user allowed across tenants is enabled
ownerfirstname	Required
ownerlastname	Required
owneremail	Required
ownerusername	Required
countrycode	Required
state	may be required based on country
city	Required
street1	Required
street2	
zipcode	Required
isdcode	Required
phonenumber	Required
promocode	

GET

Response Tags: minimal representation of object

Error Codes:

Role: Account manage or Account create or Owner

Request Parameters	Required/Optional
expand	
uuid	Required

Note: API behavior:

- expand = users, owner are blocked.
- Any user can see minimum details of other users in his account.
- This operation is only for customer users. If a service provider user attempts this Get request, then that user gets an Access Denied message.

Get (via Root)

Response Tags: minimal representation of object

Error Codes:

Role: Account manage or Account create or Owner

Request Parameters	Required/Optional
expand	

Note: API behavior:

- Any user can see minimum details of other users in his account.
- This operation is only for service provider users. If a customer user attempts this Get request, then that user gets an Access Denied message.
- When service provider user requests the account details using /accounts/uuid, then the given uuid account details are listed. If the given account is service provider account uuid, then a blank response is returned.

Update

Response Tags: minimal representation of object

Error Codes:

Role: Account create or Account User create manage

Request Parameters	Required/Optional
uuid	Required
companyname	
discount	
creditlimit	
userslimit	
countrycode	
state	
city	
street1	
street2	
zipcode	
issyncwithprimary	
secondarycountrycode	
secondarystate	
secondarycity	
secondarystreet1	
secondarystreet2	
secondaryzipcode	

Update (via Root)

Response Tags: minimal representation of object

Error Codes:

Role: Account create or Account User create manage

Request Parameters	Required/Optional
companyname	
discount	
creditlimit	
userslimit	
countrycode	
state	
city	
street1	
street2	
zipcode	

issyncwithprimary	
secondarycountrycode	
secondarystate	
secondarycity	
secondarystreet1	
secondarystreet2	
secondaryzipcode	

Delete

Response Tags: minimal representation of object

Error Codes:

Role: Account create

Request Parameters	Required/Optional
uuid	Required

Change State

Response Tags: minimal representation of object

Error Codes:

Role: Account create

Request Parameters	Required/Optional
uuid	Required
newstate	Required
statechangebecause	Required
memo	Required

Change Account Type

Response Tags: minimal representation of object

Error Codes:

Role: Owner

Request Parameters	Required/Optional
uuid	Required

Accounts

	Required
accounttypeid	
creditcardtype	
creditcardnumber	
creditcardexpiration	
creditcardcvv	
nameoncard	
countrycode	
state	
city	
street1	
street2	
zipcode	
initialpayment	

Users

Users can be accessed by a Master / Root user and the users having permissions. If the access is through Root, then the format will be: /account/<account uuid>/users/<useruuid>. If the access is by a Master or Power user, then the format will be: /users/uuid.

Expand View

- firstName
- email
- lastName
- phone
- username
- uuid
- address
 - city
 - country
 - postalcode
 - state
- apiKey
- countrycode
- gravatarUrl
- locale
- profile
 - name
 - description
 - scope
 - profileAuthorities
 - authority
 - authorityRequired

Users

- enabled
- secretKey
- spendBudget
- title
- timezone

Minimal View

- firstName
- lastName
- username
- uuid
- address
 - city
 - country
 - postalCode
 - state
- userHandles
 - handle
 - serviceInstanceUuid
- profile
 - name
 - description
- enabled

Operations	Path	Method	Sort parameter	q parameter
List	/users	GET	username	username
List(Via Root)	/accounts/{uuid}/users	GET		
Create	/users	POST		
Create(Via Root)	/accounts/{uuid}/users	POST		
Get	/users/<uuid>	GET		
Get (Via Root)	/accounts/{tenantUuid}/users/{userUuid}	GET		
Update	/users/<uuid>	PUT		

Users

Update(Via Root)	/accounts/{tenantUuid}/users/{userUuid}	PUT		
Delete	/users/<uuid>	DELETE		
Delete(Via Root)	/accounts/{tenantUuid}/users/{userUuid}	DELETE		
State Change	/users/<uuid>/changestate	POST		
State Change(Via Root)	/accounts/{tenantUuid}/users/{userUuid}/changestate	POST		

Parameters are optional unless specified as required.

List

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
expand	
page(default=1)	
pagesize(default=10),	
q	
sort	
orderby (ASC/DESC)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List (Via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
uuid	
expand	
page (default=1)	

pagesize(default=10),	
q	
sort	
order (ASC/DESC)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Note: If the requested parameter for the pagesize is greater than one, there may be a discrepancy of one count in the response returned when the listUsers() API is called from root. This is because the API call removes the portal user from the list and therefore the response returned displays all the service provider users except the portal user.

Create

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
username	Required
email	Required
firstname	Required
lastname	
profile	
timezone	
locale	

Create(Via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
uuid	Required
username	Required
email	Required

firstname	Required
lastname	
profile	
profile	
timezone	
locale	

GET

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
expand	
uuid	Required

Get (via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
expand	
userUuid	Required
tenantUuid	Required

Update

Response Tags: minimal representation of object

Error Codes:

Role: Account create or Account User create manage or Profile create manage

Request Parameters	Required/Optional
uuid	Required
firstname	

lastname	
timezone	
locale	
isdcode	
phonenumber	

Update (via Root)

Response Tags: minimal representation of object

Error Codes:

Role: Account create or Account User create manage or Profile create manage

Request Parameters	Required/Optional
tenantUuid	Required
userUuid	Required
firstname	
lastname	
timezone	
locale	
isdcode	
phonenumber	

Delete

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
uuid	Required

Delete

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
userUuid	Required
tenantUuid	Required

State Change

Response Tags: minimal representation of object

Error Codes:

Role: Account create

Request Parameters	Required/Optional
uuid	Required
state	Required

State Change (Via Root)

Response Tags: minimal representation of object

Error Codes:

Role: Owner

Request Parameters	Required/Optional
userUuid	Required
tenantUuid	Required
state	Required

Subscriptions

Subscriptions can be accessed by the Master user/Power user/Normal user to list all the subscriptions for a given tenant/user. It supports reconfiguration and provisioning.

Note: Following actions are not supported as of now:

- "update" action, so there is no way to update custom fields, However re-configuring subscription is allowed and please refer to "reconfigure" API for the same.
- "delete" is not supported because CPBM does not do hard delete, "terminate" is provided in support of terminating subscription.

In order to change the state of subscription from ACTIVE to EXPIRED, please refer to "terminate" API, other state transitions are not allowed.

Some basic concepts related to subscription related APIs-

- configurationData (JSON string of configuration data depends on Cloud Service)
- filters (value of filters defined for underlying Cloud Service) 3. context (comma separated values of context obtained from Cloud Service)

Expand View

- uuid
- activationDate
- state
- productBundle
 - name
 - description
 - code
- tenant
 - uuid
 - name
- newSubscription
- preAuthTransId
- preAuthAmount
- configurationData (varies as per Cloud Service, an example for an IaaS is given below)
 - hostName

Subscriptions

- displayName
 - group
 - diskOfferingUuid
 - keypair
 - diskSize
 - keyboard
 - networkIds
 - securitygroupIds
 - zoneUuid
 - templateUuid
 - serviceOfferingUuid
- terminationDate

Minimal View

- uuid
- activationDate
- state
- terminationDate

Operations	Path	Method	Sort parameter	q parameter
List	/subscriptions	GET		
List (Via Root)	/accounts/{uuid}/subscriptions	GET		
Create	/subscriptions	POST		
Create (Via Root)	/accounts/{uuid}/subscriptions	POST		
Get	/subscriptions/{uuid}	Get		
Get (Via Root)	/accounts/{tenantUuid}/subscriptions/{subscriptionUuid}	GET		
Terminate	/subscriptions/{uuid}	DELETE		

Terminate(Via Root)	/accounts/{tenantUuid}/subscriptions/{subscriptionUuid}	DELETE		
Reconfigure	/subscriptions/{uuid}/reconfigure	PUT		
Reconfigure(Via Root)	/accounts/{tenantUuid}/subscriptions/{subscriptionUuid}/reconfigure	PUT		
Provision	/subscriptions/{uuid}/provision	PUT		
Provision(Via Root)	/accounts/{tenantUuid}/subscriptions/{subscriptionUuid}/provision	PUT		

Parameters are optional unless specified as required.

List

Response Tags: minimal representation of object

Error Codes:

Role: User should have read permission on returned subscription object

Request Parameters	Required/Optional
state (Possible values are "ACTIVE", "CANCELED", "EXPIRED", "NEW")	
tenantuuid	
userid	
productbundleid	
page	
pagesize	
sort	
order(ASC, DESC)(Default-ASC)	
expand	

count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	
---	--

List (Via Root)

Response Tags: minimal representation of object

Error Codes:

Role: User should have read permission on returned subscription object

Request Parameters	Required/Optional
uuid	Required
state	
tenantuuid	
useruuid	
productbundleid	
page	
pagesize	
sort	
order(ASC, DESC)(Default-ASC)	
expand	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Create

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
tenantparam	Required

productbundleid	
serviceinstanceuuid	Required
resourcetype	Required
configurationdata (JSON string of configuration data)	Required
filters	
context	
provision	
expand	

Create(Via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
uuid	Required
tenantparam	
productbundleid	
serviceinstanceuuid	Required
resourcetype	Required
configurationdata	
filters	
context	
provision	
expand	

Get

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
expand	
uuid	Required

Get (Via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
expand	
tenantUuid	Required
subscriptionUuid	Required

Terminate

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
uuid	Required

Terminate (Via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
tenantUuid	Required
subscriptionUuid	Required

Reconfigure

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
--------------------	-------------------

Subscriptions

uuid	Required
serviceinstanceuuid	Required
configurationdata	Required
filters	
context	
targetsubscriptionuuid	Required
expand	

Reconfigure(Via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
tenantUuid	Required
subscriptionUuid	Required
serviceinstanceuuid	Required
configurationdata	Required
filters	
context	
targetsubscriptionuuid	Required
expand	

Provision

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
uuid	Required
resourcetype	Required
expand	

Provision (Via Root)

Response Tags: minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
tenantUuid	Required
subscriptionUuid	Required
resourcetype	Required
expand	

Accountstatements

Master user can list a particular account statement.

Expand view

- uuid
- billingPeriodStartDate
- billingPeriodEndDate
- state
- balanceForwardAmount
- credits
- renewalCharges
- subscriptionCharges
- utilityCharges
- discounts
- taxes
- finalCharges
- postedAt
- settled
- amountToBeSettled
- dueInvoicesPaid

Minimal View

- uuid
- billingPeriodStartDate
- billingPeriodEndDate
- state

Operations	Path	Method	Sort parameter	q parameter
------------	------	--------	----------------	-------------

List	/accountstatements	GET		
List(Via Root)	/accounts/<uuid>/accountstatements	GET	BillingPeriodStartDate, Id	
Get	/accountstatements/<uuid>	GET		
Get (Via Root)	/accounts/{tenantUuid}/accountstatements/{accountStatementUuid}	GET		

Parameters are optional unless specified as required.

List

Response Tags: minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
expand	
page ,(default=1)	
pagesize(default=10)	
q	
sort	
order(ASC, DESC)(Default-ASC)	
state={PROVISIONAL,POSTED}	
date :In ISO 8601 format(YYYY-mm-ddTHH:mm:ss). This is date between start date and end date of the statement	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List(Via Root)

Response Tags: minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
uuid	Required
expand	
page ,(default=1)	
pagesize(default=10)	
sort	
order(ASC, DESC)(Default-ASC)	
state={PROVISIONAL,POSTED}	
date :In ISO 8601 format(YYYY-mm-ddTHH:mm:ss) This is date between start date and end date of the statement (optional)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

GET

Response Tags: minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
expand	
uuid (Also, 'current' can be passed in place of uuid to get current provisional account statement)	Required

Get (via Root)

Response Tags: minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
tenantUuid	Required

Accountstatements

expand	
accountStatementUuid (Also, 'current' can be passed in place of uuid to get current provisional account statement)	Required

Note:

For normal users API calls, we need to add more data to UserSpendSummary. Same URLs can be used for normal users also. When CMU wants to view statement for entire tenant, a flag "admin" can be passed.

Currently, Tenant details are not passed to the caller as there is no way to send only the AccountId from the Tenant.

Invoices

This allows the tenant to list the invoice for all the users or a particular user.

Expand View

- uuid
- amount
- rawAmount
- subTotal
- amountDue
- taxAmount
- discountAmount
- generatedOn
- serviceStartDate
- serviceEndDate
- postedAt
- state
- type
- invoiceHandle
- removed
- invoiceItems
 - description
 - chargeType
 - amount
 - quantity
 - product (expanded view)
 - unitPrice
 - chargeableUsage

- chargeType
- quantity
- amount
- user

Minimal View

- uuid
- amount
- rawAmount
- subTotal
- amountDue
- state
- type

Operations	Path	Method	Sort parameter	q parameter
List	/accountstatements/{uuid}/invoices	GET		Invoice Handle
List(Via Root)	/accounts/{tenantUuid}/accountstatements/{accountStatementUuid}/invoices	GET		Invoice Handle
Get	/accountstatements/{accountStatementUuid}/invoices/{invoiceUuid}	GET		
Get(Via Root)	/accounts/{tenantUuid}/accountstatements/{accountStatementUuid}/invoices/{invoiceUuid}	GET		

Parameters are optional unless specified as required.

List

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
expand	

q, (searches in invoice handle)	
state=(Provisional,Posted)	
type={Subscription,Renewal,CloseOut,DebitNote}	
username	
handle=invoiceHandle	
due = (true/false)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List(Via Root)

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
tenantUuid	Required
accountStatementUuidRequired	Required
expand	
q, (searches in invoice handle)	
state=(Provisional,Posted)	
type={Subscription,Renewal,CloseOut,DebitNote}	
user=username	
handle=invoiceHandle	
due = (true/false)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

GET

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
expand	
uuid	Required
invoiceuuid	Required

Get (via Root)

Response Tags: Minimal representation of object

Error Codes:

Roles: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
tenantUuid	Required
expand	
accountStatementUuid	Required
invoiceuuid	Required

Note:

For normal users API calls, we need to add more data to UserSpendSummary. Same URLs can be used for normal users also. When CMU wants to view statement for entire tenant, a flag "admin" can be passed.

Currently, Tenant details are not passed to the caller as there is no way to send only the AccountId from the Tenant.

Payments

Payments can be accessed by Root/Master user to list a payment for an account and also to view a particular payment.

Expand View

- uuid
- type
- transactionAmount
- memo

Minimal View

- uuid
- type
- transactionAmount

Operations	Path	Method	Sort parameter	q parameter
List	/account/payments	GET		
List(Via Root)	/accounts/<uuid>/payments	GET		
Get	/account/payments/{paymentUuid}	GET		
Get(Via Root)	/accounts/{uuid}/payments/{paymentUuid}	GET		
Create (Make Payment/Record Payment via Root)	/accounts/<uuid>/payments	POST		
Create (Make Payment/Record Payment)	/account/payments	POST		

Parameters are optional unless specified as required.

List

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
expand	
page (default=1)	
pagesize (default=10)	
accountstatementid	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Note: Will not list failed payments. The list payment call also returns the count if the parameter **count** is set to true.

List (Via Root)

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
uuid	Required
expand	
page (default=1)	
pagesize (default=10)	
accountstatementid	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Note: The list payment call also returns the count if the parameter **count** is set to true.

GET

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
uuid	Required
paymentUuid	Required

Get (via Root)

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
uuid	Required
amount	

Create (Make Payment/Record Payment via Root)

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
uuid	Required
amount	

Create (Make Payment/Record Payment)

Response Tags: Minimal representation of object

Error Codes:

Role: Account User create manage, Billing Admin manage, Finance create manage

Request Parameters	Required/Optional
amount	

Cloud Services

Cloud services can be accessed by master user to view list of enabled services.

Expand View

- uuid
- name
- service
 - serviceName
 - category
 - type

Minimal View

- uuid
- name
- service
 - serviceName

Operations	Path	Method	Sort parameter	q parameter
List all services (via root)	/accounts/{uuid}/services	GET		
List all services	/account/services	GET		
List all services (Normal user)	/users/{uuid}/services	GET		
enable a service	/users/{uuid}/services/enable	PUT		

Parameters are optional unless specified as required.

List all services

Response Tags: Minimal representation of object

Error Codes:

Role:

Request Parameters	Required /Optional
uuid	Required
expand	
category={IAAS,STASS etc.related to CLOUD type}	
enabled={TRUE/FALSE}	

List all services (Normal user)

Response Tags: Minimal representation of object

Error Codes: AccessDeniedException, NoSuchUserException

Role:

Request Parameters	Required/Optional
expand	
category={IAAS,STASS etc.related to CLOUD type}	

Enable a service

Response Tags: Minimal representation of object

Error Codes:

- Will get 400 response in the following scenario(s)
 - Invalid service instance uuid
 - Already user signed up for the given service instance
 - Master user is not signed up for this service
 - Passed User is not active
 - Master user is not active
- Will get 404 response in the following scenario(s)
 - Passed user does not exists in the system

Role:

Request Parameters	Required/Optional
serviceinstanceuuid	Required

Catalogs

Catalogs can be accessed either through the channel (for anonymous display of catalog for a specific channel) or through a customer account (get effective catalog for a given account by mapping it to the catalog corresponding to this account). This will return all the products, bundles and campaigns associated with this catalog.

Expand View

- channel
- productRevisions(List)
 - Product
 - name
 - displayName
 - uom
 - code
- productBundleRevisions(List)
 - productBundle
 - rateCardCharges(List of RCCharge)
 - rateCardCharge
 - price
 - currencyValue
 - rateCardComponent
 - isRecurring
 - rateCard
 - description
 - startDate
 - chargeType
 - chargeRecurrenceFrequency
 - name
 - displayName

- description
 - frequencyInMonths
- entitlements(List of entitlements)
 - entitlement
 - includedUnits
 - product
 - name
 - displayName
 - uom
 - code

Operations	Path	Method	Sort parameter	q parameter
Get(Via Channel)	/channels/{uuid}/catalog	GET		
Get(Via root)	/accounts/{uuid}/catalog	GET		
Get(Via account)	/account/catalog	GET		

Parameters are optional unless specified as required.

Get(Via Channel)

Response Tags: Full data structure view

Error Codes:

Role:

Request Parameters	Required
uuid	Required
expand	

Get (Via root)

Response Tags: Full data structure view

Error Codes:

Role:

Request Parameters	Required
uuid	Required
expand	

Get (Via account)

Response Tags: Full data structure view

Error Codes:

Role:

Request Parameters	Required
expand	

Events

Returns the list of latest events based on caller's authority, and markers for pagination. If markers are not provided, then it returns the pagesize number of latest events. To paginate, call this with startMarker, marker, and pagesize to return the next set of latest events. startMarker will always remain same for every subsequent pagination call and the marker varies based on response. This also returns the newEventCount which is the number of newer events than startMarker, Which are not returned in the list, and count which is the total number of events.

Note: The events API is a special API that uses markers for pagination. Unlike other APIs, here the count is an important information that has to be returned always because there is no page parameter and the user needs to know the total number of events to paginate properly.

Expand View

- events
 - source
 - scope
 - category
 - subcategory
 - severity
 - generatedAt
 - message
 - accountholder
- startMarker
- marker
- newEventCount
- count

Minimal View

- events
 - severity
 - generatedAt
 - message

- accountHolder
- scope
- startMarker
- marker
- newEventCount
- count

Operations	Path	Method	Sort parameter	q parameter
List (Events for Tenant and its Users)	/accounts/<uuid>/events	GET		
List (Events for Account) (As a Master User)	/account/events	GET		
List (Events for Tenant's User)	/accounts/<accountuuid>/users/<useruuid>/events	GET		
List (Events for Account's User) (As a Master User)	/account/users/<useruuid>/events	GET		
List (Events for a user)	/users/<uuid>/events	GET	generatedAt	message
List Events (List my events)	/events	GET	generatedAt	message
Get	/events/<uuid>	GET		

Parameters are optional unless specified as required.

List (Events for Tenant and its Users)

Response Tags: Minimal representation of all events

Exceptions: `IllegalArgumentException` (if invalid values for source,severity,category,scope), `NoSuchTenantException` (if invalid account uuid), `accessDeniedException`(if caller is unauthorized)

Role:

Request Parameters	Required/Optional
uuid (Account's uuid specified in API's path)	Required
pagesize	

q (String which will be searched in event's message)	
source (Expected values "PORTAL" (in any case) or "<uuid of service instance>" (in exact case))	
severity (Expected values "CRITICAL", "ALERT", "INFORMATION" (in any case))	
category (Expected values "SYSTEM", "ACCOUNT", "ACCOUNT_SPEND", "SERVICES" (in any case))	
scope (Expected values "SYSTEM", "CHANNEL", "ACCOUNT", "USER" (in any case))	
startmarker (optional, Used for pagination, Used along with marker, uuid of the first event from the last result set)	
marker (optional, Used for pagination, Used along with startmarker, uuid of the last event from the last result set, events older than event with this uuid will be returned in next page)	

List (Events for Account) (As a Master User)

Response Tags: Minimal representation of all events

Exceptions: `IllegalArgumentException` (if invalid values for source,severity,category,scope), `accessDeniedException`(if caller is unauthorized)

Role:

Request Parameters	Required/Optional
uuid (Account's uuid specified in API's path)	Required
pagesize	
q (String which will be searched in event's message)	
source (Expected values "PORTAL" (in any case) or "<uuid of service instance>" (in exact case))	
severity (Expected values "CRITICAL", "ALERT", "INFORMATION" (in any case))	
category (Expected values "SYSTEM", "ACCOUNT", "ACCOUNT_SPEND", "SERVICES" (in any case))	
scope (Expected values "SYSTEM", "CHANNEL", "ACCOUNT", "USER" (in any case))	
startmarker (optional, Used for pagination, Used along with marker, uuid of the first event from the last result set)	
marker (optional, Used for pagination, Used along with startmarker, uuid of the last event from the last result set, events older than event with this uuid will be returned in next page)	

List (Events for Tenant's User)

Response Tags: Minimal representation of all events

Error Codes:

Role:

Request Parameters	Required/Optional
accountuuid (Account's uuid specified in API's path)	Required
userid (required, User's uuid specified in API's path)	
pagesize	
q (String which will be searched in event's message)	
source (Expected values "PORTAL" (in any case) or "<uuid of service instance>" (in exact case))	
severity (Expected values "CRITICAL", "ALERT", "INFORMATION" (in any case))	
category (Expected values "SYSTEM", "ACCOUNT", "ACCOUNT_SPEND", "SERVICES" (in any case))	
scope (Expected values "SYSTEM", "CHANNEL", "ACCOUNT", "USER" (in any case))	
startmarker (optional, Used for pagination, Used along with marker, uuid of the first event from the last result set)	
marker (optional, Used for pagination, Used along with startmarker, uuid of the last event from the last result set, events older than event with this uuid will be returned in next page)	

List (Events for a user)

Response Tags: Minimal representation of all events

Exceptions: `IllegalArgumentException` (if invalid values for source,severity,category,scope), `NoSuchUserException` (if invalid user uuid), `accessDeniedException`(if caller is unauthorized)

Role:

Request Parameters	Required/Optional
userid (required, User's uuid specified in API's path)	Required
pagesize	
q (String which will be searched in event's message)	
source (Expected values "PORTAL" (in any case) or "<uuid of service instance>" (in exact case))	
severity (Expected values "CRITICAL", "ALERT", "INFORMATION" (in any case))	

category (Expected values "SYSTEM", "ACCOUNT", "ACCOUNT_SPEND", "SERVICES" (in any case))	
scope (Expected values "SYSTEM", "CHANNEL", "ACCOUNT", "USER" (in any case))	
startmarker	
marker	

List (Events for Account's User) (As Master User)

Response Tags: Minimal representation of all events

Exceptions: `IllegalArgumentException` (if invalid values for source,severity,category,scope), `NoSuchUserException` (if invalid user uuid), `accessDeniedException`(if caller is unauthorized)

Role:

Request Parameters	Required/Optional
uuid (required, User's uuid specified in API's path)	Required
pagesize	
q (String which will be searched in event's message)	
source (Expected values "PORTAL" (in any case) or "<uuid of service instance>" (in exact case))	
severity (Expected values "CRITICAL", "ALERT", "INFORMATION" (in any case))	
category (Expected values "SYSTEM", "ACCOUNT", "ACCOUNT_SPEND", "SERVICES" (in any case))	
scope (Expected values "SYSTEM", "CHANNEL", "ACCOUNT", "USER" (in any case))	
startmarker (optional, Used for pagination, Used along with marker, uuid of the first event from the last result set)	
marker (optional, Used for pagination, Used along with startmarker, uuid of the last event from the last result set, events older than event with this uuid will be returned in next page)	

List Events (List my events)

Response Tags: Minimal representation of all events

Exceptions: `IllegalArgumentException` (if invalid values for source,severity,category,scope)

Role:

Request Parameters	Required/Optional
--------------------	-------------------

pagesize	
q (String which will be searched in event's message)	
source (Expected values "PORTAL" (in any case) or "<uuid of service instance>" (in exact case))	
severity (Expected values "CRITICAL", "ALERT", "INFORMATION" (in any case))	
category (Expected values "SYSTEM", "ACCOUNT", "ACCOUNT_SPEND", "SERVICES" (in any case))	
scope (Expected values "SYSTEM", "CHANNEL", "ACCOUNT", "USER" (in any case))	
startmarker (optional, Used for pagination, Used along with marker, uuid of the first event from the last result set)	
marker (optional, Used for pagination, Used along with startmarker, uuid of the last event from the last result set, events older than event with this uuid will be returned in next page)	

GET

Response Tags: Minimal representation of single event

Error Codes:

Role:

Request Parameters	Required/Optional
uuid	Required

Service Health Status

Service Health Status API can be accessed by the Root/Master/normal user to view the latest as well as recent (for last 7 days from today or the given date, if passed as parameter) health status updates for all the service instances that the user has enabled.

Expand View

- serviceHealthWrapper
 - count
 - marker (this is the date which can be passed in the date parameter in the next call to get next set of status updates. e.g. if updates for dates 2013-05-02 to 2013-05-08 are listed, then returned marker will be 2013-05-01)
 - serviceHealthNotifications
 - serviceHealthNotification
 - serviceInstance(minimal)
 - latest (the latest status for this service instance even if it is older than 7 days from today or given date param. If it is older than 7 days from today or given date param, then the 'recent' list mentioned below will be empty)
 - notificationType(minimal)
 - recordedOn(minimal)
 - subject(minimal)
 - serviceInstance(minimal)
 - description(minimal)
 - recordedBy
 - plannedStart
 - plannedEnd
 - recent (list of health status for this service instance for last 7 days)

Minimal View

- serviceHealthWrapper
 - count
 - marker (this is the date which can be passed in the date parameter in the next call to get next set of status updates. e.g. if updates for dates 2013-05-02 to 2013-05-08 are listed, then returned marker will be 2013-05-01)

- serviceHealthNotifications
 - serviceHealthNotification
 - serviceInstance
 - latest (the latest status for this service instance even if it is older than 7 days from today or given date param. If it is older than 7 days from today or given date param, then the 'recent' list mentioned below will be empty)
 - notificationType(minimal)
 - recordedOn(minimal)
 - subject(minimal)
 - description(minimal)
 - recent (list of health status for this service instance for last 7 days)

Operations	Path	Method	Sort parameter	q parameter
List	/health/status	GET		subject, description

Parameters are optional unless specified as required.

List

Response Tags:

Exceptions: IllegalArgumentException (If date passed is not in proper syntax)

Request Parameters	Required/Optional
expand (possible values are : 'serviceHealthNotifications.serviceInstance', 'serviceHealthNotifications.latest', 'serviceHealthNotifications.recent')	
q (Free text Searchable query on 'subject' and 'description' fields)	
serviceinstanceuuid (list status updates only for the given service instance)	
date (if given the allowed format is 'yyyy-MM-dd') <ul style="list-style-type: none"> • If not passed, the status updates for last 7 days from today (ordered by record date descending and grouped by service instances) are returned • If a past date is passed, status updates for last 7 days from the given date (ordered by record date descending and grouped by service instances) are returned • If a future date is passed, that will result in an IllegalArgumentException 	

Service Health Status

count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	
--	--

Service Maintenance Activities

Service maintenance activities API can be accessed by the Root/Master/normal user to view all the future maintenance activities (or maintenance activities including a future date passed as parameter) for all the service instances, the user has enabled. If 'page' and 'pagesize' are not given, then 10 upcoming maintenance activities are listed by default.

Expand View

- notificationType(minimal)
- recordedOn(minimal)
- subject(minimal)
- serviceInstance(minimal)
- description(minimal)
- plannedStart(minimal)
- plannedEnd(minimal)
- recordedBy

Minimal View

- notificationType
- recordedOn
- subject
- description
- plannedStart
- plannedEnd

Operations	Path	Method	Sort parameter	q parameter
List	/health/maintenance	GET		

Parameters are optional unless specified as required.

List

Response Tags:

Exceptions: `IllegalArgumentException` (If date passed is not in proper syntax)

Request Parameters	Required/Optional
expand (possible values are : 'recordedBy' and 'serviceInstance')	
page	
pagesize	
q (Free text Searchable query on 'subject' and 'description' fields)	
serviceinstanceuuid (list status updates only for the given service instance)	
<p>date (if given the allowed format is 'yyyy-MM-dd')</p> <ul style="list-style-type: none"> • If not passed, the status updates for last 7 days from today (ordered by record date descending and grouped by service instances) are returned. • If a past date is passed, status updates for last 7 days from the given date (ordered by record date descending and grouped by service instances) are returned. • If a future date is passed, that will result in an IllegalArgumentException 	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Channels

Channels can be accessed by the root to list all the channels, a particular channel, or all channels belonging to a billing group.

Expand View

- uuid (minimal)
- name (minimal)
- code (minimal)
- description (minimal)
- catalog
 - planDate (minimal)
 - supportedCurrencies
 - currencyCode
 - currencyName

Minimal View

- uuid
- name
- code
- description

Operations	Path	Method	Sort parameter	q parameter
List	/channels	GET		name
Get	/channels/<uuid>	GET		

Parameters are optional unless specified as required.

List

Response Tags: Minimal representation of object

Error Codes:

Role:

Channels

Request Parameters	Required/Optional
expand	
page	
perpage	
q (Search parameter for a channel)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Get

Response Tags: Minimal representation of object

Error Codes:

Role:

Request Parameters	Required/Optional
expand	

Currencies

Currencies can be accessed by the root user to view all the currencies supported in a channel.

Expand View

- `currencyCode`(minimal)
- `currencyName`(minimal)
- `sign`
- `flag`

Minimal View

- `currencyCode`
- `currencyName`

Operations	Path	Method	Sort parameter	q parameter
List	<code>/channels/{channelUuid}/currencies</code>	GET		name

Parameters are optional unless specified as required

List

Response Tags: minimal representation of object

Error Codes:

Role: any role

Request Parameters	Required/Optional
expand	
count (Possible values are 'true' or 'false'(default). If set as <code>count=true</code> in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Usage

Returns usage and spend aggregated by product and users for the current billing period. Also used to return usage and spend aggregation by product for a channel and billing group. For usage for a earlier billing period, call the API with the date parameter and aggregate the usage for billing period within which the date is returned. To filter Product usage based on service instance, call API with serviceInstanceUuid parameter.

List (Aggregated product usage and spend for a given account, based on given date)

Response Tags: Aggregated product usage and spend for Account

Error Codes: illegalArgumentException (if date is not in proper format or date does not lie in a valid billing period), nosuchTenantException (if invalid account uuid)

Role:

Request Parameters	Required
uuid (Account's uuid specified in API's path)	Required
sort	
order	
date (specified in yyyy-MM-dd format and in server time zone, Usage for billing period within which the given date lies)	
serviceinstanceuuid (to filter product usage by service instance)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List (Aggregated product usage and spend for a given account, based on given date)

Response Tags: Aggregated product usage and spend for Account

Error Codes: illegalArgumentException (if date is not in proper format or date does not lie in a valid billing period)

Role:

Request Parameters	Required
sort	
order	
date (specified in yyyy-MM-dd format and in server time zone, Usage for billing period within which the given date lies)	
serviceinstanceuuid (to filter product usage by service instance)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List (Aggregated product usage and spend for a given user, based on given date)

Response Tags: Aggregated product usage and spend for User

Error Codes: illegalArgumentException (if date is not in proper format or date does not lie in a valid billing period)

Role:

Request Parameters	Required
accountUuid	Required
userUuid	Required
sort	
order	
date (specified in yyyy-MM-dd format and in server time zone, Usage for billing period within which the given date lies)	
serviceinstanceuuid (to filter product usage by service instance)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List (Aggregated product usage and spend for a given user, based on given date)

Response Tags: Aggregated product usage and spend for User

Error Codes: `IllegalArgumentException` (if date is not in proper format or date does not lie in a valid billing period)

Role:

Request Parameters	Required
uuid	Required
sort	
order	
date (specified in yyyy-MM-dd format and in server time zone, Usage for billing period within which the given date lies)	
serviceinstanceuuid (to filter product usage by service instance)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List (Aggregated product usage and spend for a given user, based on given date)

Response Tags: Aggregated product usage and spend for User

Error Codes: `IllegalArgumentException` (if date is not in proper format or date does not lie in a valid billing period)

Role:

Request Parameters	Required
uuid	Required
sort	
order	
date (specified in yyyy-MM-dd format and in server time zone, Usage for billing period within which the given date lies)	
serviceinstanceuuid (to filter product usage by service instance)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

List (Aggregated product usage and spend for a given user, based on given date)

Response Tags: Aggregated product usage and spend for calling User or all Accounts based on caller.

Error Codes: illegalArgumentException (if date is not in proper format or date does not lie in a valid billing period)

Role:

Request Parameters	Required
expand	
sort	
order	
date (specified in yyyy-MM-dd format and in server time zone, Usage for billing period within which the given date lies)	
serviceinstanceuuid (to filter product usage by service instance)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Profiles

Profiles can be accessed only by the Root to view all the profiles.

Expand View

- profile
- name
- description
- scope
- profileAuthorities
 - authority
 - authorityRequired

Minimal View

- name
- description

Operations	Path	Method	Sort parameter	q parameter
List	/profiles	GET	Anything	name

Parameters are optional unless specified as required.

List

Response Tags: minimal representation of all Profiles

Error Codes:

Role:

Request Parameters	Required
page (defaults to 1)	
pagesize (defaults to 10)	
sort	
order	
q (Searchable query on profile name)	

scope (can be tenant, tenant_admin, user, global, global_admin)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

AccountTypes

AccountTypes can be accessed by the root user to view all the account types.

Expand View

- name(minimal)
- displayName(minimal)
- description(minimal)

Minimal View

- name
- displayName
- description

Operations	Path	Method	Sort parameter	q parameter
List	/accounttypes	GET	Anything	name

Parameters are optional unless specified as required.

List

Response Tags: minimal view of AccountType Object

Error Codes:

Role: Account create

Request Parameters	Required / Optional
expand	
page (defaults to 1)	
pagesize (defaults to 10)	
sort	
order (ASC/DESC)	
q (Searchable query on country name and code)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Countries

Countries can be accessed by the root/master user to view all the countries.

Expand View

- name(minimal)
- countryCode2(minimal)
- countryCode3
- isEU
- isdCode
- stateCodes

Minimal View

- name
- countryCode2

Operations	Path	Method	Sort parameter	q parameter
List	/countries	GET	name, countryCode	name

Parameters are optional unless specified as required.

List

Response Tags: minimal representations of all countries

Error Codes:

Role:

Request Parameters	Required
expand	
countrycode (countryCode2)	
isdcode	
page (defaults to 1)	
pagesize (defaults to 10)	
sort	

Countries

order (ASC/DESC)	
q (Searchable query on country name and code)	
count (Possible values are 'true' or 'false'(default). If set as count=true in the API parameters this will return a count in response (as first level field) that represents the total number of entities matching the passed criteria irrespective of passed page and pagesize.)	

Known issues

This following section list some of the known issues:

- If the List is empty, then json response will come as {} and xml will come as empty xml document.
- Count is implemented only for accounts, events, service health notification and service health status.
- Error codes are not yet finalized.
- Validation checks on request parameters have not been completely implemented.
- Security has not been fully applied for the APIs.
- There may be a discrepancy of one count in the response returned when the List user API is called from root when the requested parameter for the pagesize is greater than one.
- A normal user cannot fetch accountstatement uuid and, therefore, cannot fetch any object which requires account statement uuid. Only the master user, billing admin, and finance admin can do that.

Customization Guide

This guide describes how to implement customization in Citrix® CloudPortal™ Business Manager. This guide is for the implementors who want to implement the customization in CPBM.

To use this guide, you should have a working knowledge of the following subjects:

- Knowledge of Java, Spring, MVC architecture, jQuery, javascript, jsp, VIRGO framework, velocity templating language, quartz, hibernate, xml, json, and liquibase.

Customization Overview

The Customization pack allows customer or professional services team to extend or customize CloudPortal Business Manager.

Customization pack comes with two GIT repositories.

- **cpbm-customization.git**: This repository has three bundles which allows customization of CloudPortal Business Manager product.
- **cpbm-ccp-connector.git**: This repository has one bundle which allows customization of user interface of CCP connector.

Bundles in cpbm-customization.git:

- citrix.cpbm.custom.model
 - This bundle consists of all the proxy model interfaces of the customizable entities. These are Tenant, User, and Subscription proxy interfaces used for custom fields.
 - This bundle also consists a package called com.citrix.cpbm.custom.model which can contain any custom model required for customization.
- citrix.cpbm.custom.common
 - All custom liquibase goes under this.
 - Custom DAOs and Services are written here.
 - Custom job is configured here.
 - Custom workflow is configured here.
 - Custom report is configured here.
 - Custom email templates
 - Topic Subscribers
- citrix.cpbm.custom.portal
 - CPBM open sourced portal UI contents like controllers, view (JSPs, javascripts, css, and so on).
 - WebSecurity xml consisting of access based control for URL paths.

Bundle in cpbm-ccp-connector.git

- cloud.cloudplatform.fragment
 - CCP connector UI contents like controllers, view (JSPs, javascripts, css, and so on).

Installing and building custom components

The four CloudPortal Business Manager components that can be customized are available in two customization repositories:

- Bundles available in `cpbm-customization.git` are:
 - Bundle `citrix.cpbm.custom.model`
 - Bundle `citrix.cpbm.custom.portal`
 - Bundle `citrix.cpbm.custom.common`
- For CPBM 2.1.0 and CPBM 2.1.1, bundle available in `cpbm-ccp-connector.git` is:
 - Bundle `cloud.cloudplatform.fragment`
- For CPBM 2.1.2, bundle available in `CPBM-<version>-ccp-connector-ui.git` is:
 - Bundle `cloud.cloudplatform.fragment`

Note: These git repositories are released along with every CloudPortal Business Manager release.

Any customer or service provider who is interested in customizing these components can follow the steps given below.

Pre-requisites for setting up the development setup:

- **GIT client:** GIT downloads are available <http://git-scm.com/downloads> and install instructions are available at <http://git-scm.com/book/en/Getting-Started-Installing-Git>
- **Maven 3.0.4:** Maven downloads and install instructions are available at <http://maven.apache.org/download.cgi>
- **JDK 6:** JDK downloads and install instructions are available at <http://www.oracle.com/technetwork/java/javase/index-137561.html>

1. Setting up remote and local GIT repositories:

For CPBM versions 2.1.0 and 2.1.1:

- a. Download the latest CloudPortal Business Manager customization repositories tar ball:

```
cpbm-customization.git.tar.gz
```

- b. Unzip the customization tar ball:

```
$ tar xvf cpbm-customization.git.tar.gz
```

- c. This will be a bare GIT repository which you will use only to pull changes. For example, call it `pristine-cpbm-customization` repo. Create another bare repository from the above repository which will be your customer specific repository where you will pull and push changes on daily basis:

```
$ git clone --bare </path/to/pristine-cpbm-customization/repo>  
xyz-cpbm-customization
```

- d. Clone a normal repository with working directory from customer specific repository on development environment:

```
$ git clone </path/to/xyz-cpbm-customization>
```

- e. Change directory: `cd xyz-cpbm-customization`

- f. Add the `pristine-cpbm-customization` as a remote into the repository on development instance. For example, call it `pristine-remote`:

```
$ git remote add pristine-remote </path/to/pristine-cpbm-customization/repo>
```

For CPBM version 2.1.2:

- a. Download the latest CloudPortal Business Manager customization repositories tar ball:

```
CPBM-<version>-cpbm-customization.git.tar.gz
```

- b. Unzip the customization tar ball:

```
$ tar xvf CPBM-<version>-cpbm-customization.git.tar.gz
```

- c. This will be a bare GIT repository which you will use only to pull changes. For example, call it `pristine-cpbm-customization` repo. Create another bare repository from the above repository which will be your customer specific repository where you will pull and push changes on daily basis:

```
$ git clone --bare </path/to/pristine-cpbm-customization/repo>  
xyz-cpbm-customization
```

Note: For some older version of GIT, you might see a warning " Initialized empty Git repository " which is fine and can be ignored. To make sure that the repository is created correctly, run "git log" from inside the repository and you should see some commits.

- d. Clone a normal repository with working directory from customer specific repository on development environment:

```
$ git clone </path/to/xyz-cpbm-customization>
```

- e. Change directory:

```
cd xyz-cpbm-customization
```

- f. Add the `pristine-cpbm-customization` as a remote into the repository on development instance. For example, call it `pristine-remote` :

```
$ git remote add pristine-remote </path/to/pristine-cpbm-customization/repo>
```

g. Create and checkout a local development branch:

- Example: To create and checkout a new branch at tag for CPBM-2.1.2 release, use the command given below:

```
$ git checkout -b <local-dev-branch-name> tag-2.1.2-GA
```

- **Note:** By creating a new branch at the tag of the CPBM release, you make sure that you are developing at the correct version of the customization SDK.

2. Once you have set up the repositories as shown above, follow the steps given below whenever a new version of `cpbm-customization repo` is released:

For CPBM versions 2.1.0 and 2.1.1:

a. Untar the customization tar ball into the existing `pristine-cpbm-customization` repository: (run the command on the remote repository)

```
$ tar xvf cpbm-customization.git.tar.gz -C  
</path/to/parent/of/pristine-cpbm-customization/repo>
```

b. From your local development repository, fetch the `pristine-remote`:

```
$ git fetch pristine-remote
```

c. Merge the master of `pristine-remote` into local master:

```
$ git merge pristine-remote/master
```

d. Push your changes into customer specific repository:

```
$ git push origin master
```

For CPBM version 2.1.2:

a. Untar the customization tar ball into the existing `pristine-cpbm-customization` repository: (run the command on the remote repository):

```
$ tar xvf cpbm-customization.git.tar.gz -C  
</path/to/parent/of/pristine-cpbm-customization/repo>
```

b. From your local development repository, fetch the `pristine-remote`:

```
$ git fetch pristine-remote
```

c. Checkout the local development branch where you intend to merge latest changes:

```
git checkout <local-dev-branch-name>
```

d. Merge the release tag on `pristine-remote` into local master:

```
$ git merge tag-<release version>
```

Example:

```
$ git merge tag-2.1.2-GA
```

- e. Push your changes into customer specific repository after resolving conflicts (if any):

```
$ git push origin <development branch>
```

3. Steps to generate custom jar files:

- a. Make required changes in any of the custom projects.
- b. Commit your changes and push them into your customer specific repository.
- c. Build the citrix.cpbm.custom.all project (without running tests):

```
mvn clean install -Dmaven.test.skip=true
```

You can now copy the new jar files into the CloudPortal Business Manager server.

To run the tests, you must create all tests schemas, as shown below, into your MySQL database.

- cloud_portal_test
- cloud_reports_test
- cloud_test
- cloud_usage_test

You can use command given below to run tests:

```
mvn clean test
```

4. Steps to deploy updated custom jar files:

- a. Stop the CloudPortal Business Manager server using the command:

```
$ service cloud-portal stop
```

- b. Go to /usr/share/vts3/config.
- c. Open org.eclipse.virgo.repository.properties.
- d. Comment the following lines:

```
default_impls.type=watched  
default_impls.watchDirectory=default_impls
```

- e. Uncomment the following lines:

```
custom_impls.type=watched  
custom_impls.watchDirectory=custom_impls
```

- f. Modify the line to have custom_impls instead of default_impls

```
chain=ext,usr,patches,connectors,default_impls,prop
```

with

with chain=ext,usr,patches,connectors,custom_impls,prop

Note: Do not modify the jars under the default_impls directory.

- g. Copy the updated custom jars from target directory of all four custom projects to:

/usr/share/vts3/custom_impls

- h. Start the CloudPortal Business Manager server using the command:

```
$ service cloud-portal start
```

Customizing UI

Customers can customize the CloudPortal Business Manager UI by modifying any of the below items in the custom bundle `citrix.cpbm.custom.portal`:

- **CSS:**

CloudPortal Business Manager provides a hook to insert customized CSS. Customer can add any CSS property into the file `citrix.cpbm.custom.portal/src/main/resources/custom/css/custom.css`. This file overrides any CSS property available in CPBM core.

- **Javascript:**

All the JavaScript files are available in custom bundle `citrix.cpbm.custom.portal` at `src/main/resources/js/`. Customers can customize these files if required.

- **JSP:**

All the JSP files are available in custom bundle `citrix.cpbm.custom.portal` at `src/main/resources/WEB-INF/jsp/` directory. CPBM uses Apache Tiles* templating framework for development of user interfaces. We have provided a hook in the form of custom tile definition (`citrix.cpbm.custom.portal/src/main/resources/WEB-INF/custom/tiles-defs.xml`) which takes precedence over default tiles definition (`citrix.cpbm.custom.portal/src/main/resources/WEB-INF/default/tiles-defs.xml`). To update any tiles view/definition, a customer can copy the view/definition from default tiles XML into the custom tiles XML and then make any change there. Customers can also add new definitions into the custom tiles XML.

- **Controllers:**

CPBM is using Spring MVC framework. And all controllers are available in custom bundle `citrix.cpbm.custom.portal` at `src/main/java/com/citrix/cpbm/portal/fragment/controllers/` directory. Every controller has one concrete implementation which extends the abstract implementation of the controller. Customers can override the method in the concrete implementation to modify any existing method or add new methods in the concrete implementation.

After making the required modification, customers can re-package the custom bundle and deploy it into CloudPortal Business Manager server. For more information on how to re-package and deploy the custom bundle, refer to [Installing and building custom components](#).

Example: To customize an image, we can put the respective image in the `citrix.cpbm.custom.portal\src\main\resources\images` folder and change the respective references in the `*.css`, `*.js`, `*.jsp` files.

Currently the jsp for footer is located at `citrix.cpbm.custom.portal\src\main\resources\WEB-INF\jsp\tiles\shared\footer.jsp`. To change the footer, you have to add the new jsp in the above location and change the

corresponding references in
citrix.cpbm.custom.portal\src\main\resources\WEB-INF\default\tiles-defs.xml.

Customizing Channel-specific CSS

Whenever you add a channel and you want to have channel-specific css, follow these steps:

1. Refer to the Customization SDK: [Customizing the UI](#).
2. Follow these steps:
 - a. In citrix.cpbm.custom.portal project
 - b. Under citrix.cpbm.custom.portal-x.x.x/custom/css directory, create a folder with name channel (if not already created). x.x.x is the CPBM version.
 - c. Create a file <ChannelCode>.css with your channel specific css under channel directory.
3. Build the citrix.cpbm.custom.portal jar and deploy into /usr/share/vts3/custom_impls directory. Refer to [Installing and building custom components](#).

Customizing Validation Rules

You can customize validation rules for predefined properties in a model or apply validation rules on custom fields for an existing model.

Configuring validation Rules or Constraints

CloudPortal Business Manager ships a bundle *citrix.cpbm.custom.common* which contains a file named *validation-constraints.xml* (path: *citrix.cpbm.custom.common/src/main/resources/validation-constraints.xml*). Any new validation constraints should be added or modified in this file. The file name should not be changed as CloudPortal Business Manager looks for this file in this bundle to bootstrap the validation constraints. As part of implementing validations, Hibernate Validator is used which is an implementation of JSR-303 specification.

Sample validation-constraint.xml

```
<constraint-mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jboss.org/xml/ns/javax/validation/mapping validation-mapping-1.0.xsd"
  xmlns="http://jboss.org/xml/ns/javax/validation/mapping">

  <bean class="com.citrix.cpbm.access.Tenant">
    <getter name="owner">
      <valid />
    </getter>
    <getter name="address">
      <valid />
    </getter>
  </bean>
  <bean class="com.citrix.cpbm.access.User">
    <getter name="firstName">
      <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
        <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
        </message>
      </constraint>
      <constraint annotation="org.hibernate.validator.constraints.Length">
        <message>javax.validation.constraints.Size.message</message>
        <element name="min">1</element>
        <element name="max">255</element>
      </constraint>
    </getter>
    <getter name="lastName">
      <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
        <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
        </message>
      </constraint>
      <constraint annotation="org.hibernate.validator.constraints.Length">
        <message>javax.validation.constraints.Size.message</message>
        <element name="min">1</element>
        <element name="max">255</element>
      </constraint>
    </getter>
  </bean>
</constraint-mappings>
```

```
</getter>
<getter name="email">
  <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
    <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
    </message>
  </constraint>
  <constraint annotation="org.hibernate.validator.constraints.Email">
    <message>javax.validation.constraints.Email.message</message>
  </constraint>
  <constraint annotation="org.hibernate.validator.constraints.Length">
    <message>javax.validation.constraints.Size.message</message>
    <element name="min">5</element>
    <element name="max">255</element>
  </constraint>
</getter>
<getter name="username">
  <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
    <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
    </message>
  </constraint>
  <constraint annotation="javax.validation.constraints.Pattern">
    <message>javax.validation.constraints.Pattern.message</message>
    <element name="regexp">[a-zA-Z0-9_@\\-\\.]+</element>
  </constraint>
  <constraint annotation="javax.validation.constraints.Size">
    <message>javax.validation.constraints.MinimumSize.message</message>
    <element name="min">5</element>
  </constraint>
  <constraint annotation="org.hibernate.validator.constraints.Length">
    <message>javax.validation.constraints.Size.message</message>
    <element name="min">5</element>
    <element name="max">255</element>
  </constraint>
</getter>
<getter name="phone">
  <constraint annotation="javax.validation.constraints.Pattern">
    <message>javax.validation.constraints.Pattern.message</message>
    <element name="regexp">(\\d+)(-)(\\d+)</element>
  </constraint>
  <constraint annotation="org.hibernate.validator.constraints.Length">
    <message>javax.validation.constraints.Size.message</message>
    <element name="min">1</element>
    <element name="max">20</element>
  </constraint>
</getter>
</bean>
<bean class="com.vmops.model.Address">
  <getter name="street1">
    <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
      <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
      </message>
    </constraint>
    <constraint annotation="org.hibernate.validator.constraints.Length">
      <message>javax.validation.constraints.Size.message</message>
      <element name="min">1</element>
      <element name="max">255</element>
    </constraint>
  </getter>
</bean>
```

```
        </constraint>
    </getter>
    <getter name="city">
        <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
            <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
            </message>
        </constraint>
        <constraint annotation="org.hibernate.validator.constraints.Length">
            <message>javax.validation.constraints.Size.message</message>
            <element name="min">1</element>
            <element name="max">255</element>
        </constraint>
    </getter>
    <getter name="state">
        <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
            <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
            </message>
        </constraint>
        <constraint annotation="org.hibernate.validator.constraints.Length">
            <message>javax.validation.constraints.Size.message</message>
            <element name="min">1</element>
            <element name="max">255</element>
        </constraint>
    </getter>
    <getter name="postalCode">
        <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
            <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
            </message>
        </constraint>
        <constraint annotation="org.hibernate.validator.constraints.Length">
            <message>javax.validation.constraints.Size.message</message>
            <element name="min">1</element>
            <element name="max">255</element>
        </constraint>
    </getter>
    <getter name="country">
        <constraint annotation="com.citrix.cpbm.validator.constraint.NotBlank">
            <message>com.citrix.cpbm.validator.constraint.impl.NotBlank.message
            </message>
        </constraint>
        <constraint annotation="org.hibernate.validator.constraints.Length">
            <message>javax.validation.constraints.Size.message</message>
            <element name="min">1</element>
            <element name="max">255</element>
        </constraint>
    </getter>
</bean>
<bean class="com.citrix.cpbm.access.Subscription">

</bean>

</constraint-mappings>
```

Built-in Constraints

Annotation	Part of Bean Validation Specification	Apply On	Use	Metadata impact
<code>javax.validation.constraints.AssertFalse</code>	yes	field/property	check that the annotated element is false.	none
<code>javax.validation.constraints.AssertTrue</code>	yes	field/property	check that the annotated element is true.	none
<code>javax.validation.constraints.DecimalMax</code>	yes	field/property. Supported types are <code>BigDecimal</code> , <code>BigInteger</code> , <code>String</code> , <code>byte</code> , <code>short</code> , <code>int</code> , <code>long</code> and the respective wrappers of the primitive types.	The annotated element must be a number whose value must be lower or equal to the specified maximum. The parameter value is the string representation of the max value according to the <code>BigDecimal</code> string representation.	none
<code>javax.validation.constraints.DecimalMin</code>	yes	field/property. Supported types are <code>BigDecimal</code> , <code>BigInteger</code> , <code>String</code> , <code>byte</code> , <code>short</code> , <code>int</code> , <code>long</code> and the respective wrappers of the primitive types.	The annotated element must be a number whose value must be higher or equal to the specified minimum. The parameter value is the string representation of the min value according to the <code>BigDecimal</code> string representation.	none
<code>javax.validation.constraints.Digits(integer=, fraction=)</code>	yes	field/property. Supported types are <code>BigDecimal</code> , <code>BigInteger</code> , <code>String</code> , <code>byte</code> , <code>short</code> , <code>int</code> , <code>long</code> and the respective wrappers of the primitive types.	Check whether the property is a number having up to integer digits and fraction fractional digits.	Define column precision and scale.

org.hibernate.validator.constraints.Email	no	field/property. Needs to be a string.	Check whether the specified string is a valid email address.	none
javax.validation.constraints.Future	yes	field/property. Supported types are java.util.Date and java.util.Calendar.	Checks whether the annotated date is in the future.	none
org.hibernate.validator.constraints.Length(min=, max=)	no	field/property. Needs to be a string.	Validate that the annotated string is between min and max included.	none
javax.validation.constraints.Max	yes	field/property. Supported types are BigDecimal, BigInteger, String, byte, short, int, long and the respective wrappers of the primitive types.	Checks whether the annotated value is less than or equal to the specified maximum.	Add a check constraint on the column.
javax.validation.constraints.Min	yes	field/property. Supported types are BigDecimal, BigInteger, String, byte, short, int, long and the respective wrappers of the primitive types.	Check whether the annotated value is higher than or equal to the specified minimum.	Add a check constraint on the column.
javax.validation.constraints.NotNull	yes	field/property.	Check that the annotated value is not null.	Column(s) are not null.
org.hibernate.validator.constraints.NotEmpty	no	field/property. Needs to be a string.	Check if the string is not null nor empty.	none
javax.validation.constraints.Null	yes	field/property	Check that the annotated value is null.	none
javax.validation.constraints.Past	yes	field/property. Supported types are java.util.Date and java.util.Calendar.	Checks whether the annotated date is in the past.	none
javax.validation.constraints.Pattern(regex=, flag=)	yes	field/property. Needs to be a string.	Check if the annotated string match the regular expression regex.	none

org.hibernate.validator.constraints.Range(min=, max=)	no	field/property. Supported types are BigDecimal, BigInteger, String, byte, short, int, long and the respective wrappers of the primitive types	Check whether the annotated value lies between (inclusive) the specified minimum and maximum.	none
javax.validation.constraints.Size(min=, max=)	yes	field/property. Supported types are String, Collection, Map and arrays.	Check if the annotated element size is between min and max (inclusive).	Column length will be set to max.
javax.validation.constraints.Valid	yes	field/property .	Perform validation recursively on the associated object.	none

Advanced CPBM Defined Constraints

Annotation	Part of Bean Validation Specification	Apply On	Use	Metadata impact
com.citrix.cpbm.validator.constraint.CreditCard	no	field/property.	check that the annotated element is a valid credit card.	none
com.citrix.cpbm.validator.constraint.DateRange	no	field/property.	check that the annotated element is in between the specified range.	none
com.citrix.cpbm.validator.constraint.IPAddress	no	field/property.	check that the annotated element is a valid IPv4/IPv6.	none
com.citrix.cpbm.validator.constraint.List	no	field/property.	check that the annotated element is one of the element defined in the list.	none

com.citrix.cpbm.validator.constraint.NotBlank	no	field/property.	check that the annotated element is not empty or blank or null.	none
com.citrix.cpbm.validator.constraint.Url	no	field/property.	check that the annotated element is valid URL.	none

Examples- Advanced Constraints

Date Range

```
<constraint annotation="com.citrix.cpbm.validator.constraint.DateRange" >
  <element name="startDate">2013-10-01</element>
  <element name="endDate">2013-12-01</element>
  <element name="format">yyyy-MM-dd</element>
</constraint>
```

List

```
<constraint annotation="com.citrix.cpbm.validator.constraint.List">
  <element name="value">
    <value>INDIA</value>
    <value>USA</value>
  </element>
</constraint>
```

Resolving Message Codes

Whenever a constraint violation happens, it is likely that we would like to support the same via customized error message. Hence, in order to achieve the same, we can add message codes to src/main/resources/OSGI-INF/l10n/resources/CustomApplicationResources.properties and corresponding error messages.

Now in case of constraint violation, error message codes would be resolved with proper error message and gets its passed to subsequent layers i.e. API/JSP/Controllers.

```
<constraint annotation="com.citrix.cpbm.validator.constraint.List">
  <element name="message">list.validator.constraint.error.message.code</element>
  <element name="value">
    <value>INDIA</value>
    <value>USA</value>
  </element>
</constraint>
```

Customizing Fields and Validations

This section describes the metadata and semantics for supporting custom validations and custom fields in CloudPortal Business Manager. Often, service operators require custom validation of tenants, users, or subscription fields. In addition, customers may require the collection and management of additional custom fields. This specification describes the metadata that a service operator provides to collect, validate, and manage custom fields, in addition to providing custom validation rules for all customer or service provider editable fields. This specification is provided by the service operator as part of customization in a file called `citrix.cpbm.custom.common/src/main/resources/validation-constraints.xml`.

Requirements:

1. Services team should be able to define Custom Fields for Tenant, User, Subscription in the Customization SDK.
2. Services team should be able to define Custom Validation for the Custom Fields in the Customization SDK.
3. Custom Field values are stored in the `custom_fields` table.
4. Supported custom fields types are text, textarea, radio, checkbox, select, hidden.

Defining custom fields for customizable objects: Tenant, User, and Subscription

Addition of custom fields is supported using dynamic proxy. This makes it possible to access custom fields objects in UI as the first level field from customizable objects.

To add a custom field for any customizable object, you need to add the getter and setter of the field into the proxy customizable object.

An example of how to add a custom field "Alternate Mobile number" to User object is given below. All the proxy customizable objects are located at `citrix.cpbm.custom.model/src/main/java/citrix/cpbm/access`.

```
package com.citrix.cpbm.access;

import citrix.cpbm.custom.field.annotation.UniqueValue;

public interface User extends UserEntity {

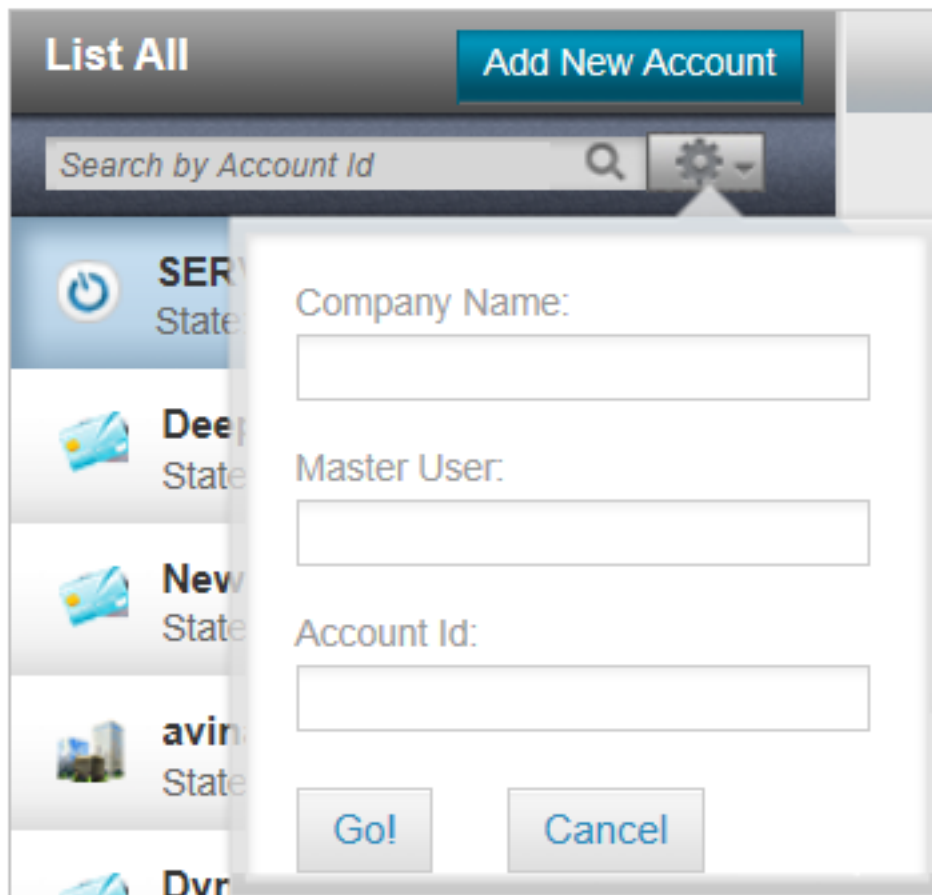
    public String getAlternatePhone();

    @UniqueValue
    public void setAlternatePhone(String phone);
}
```

As seen in the example, the custom field is annotated with `@UniqueValue`, which makes the custom field value unique across users.

You can annotate custom fields in the following four different ways:

- @UniqueValue: To mark its value unique across the customizable object.
- @Default: To set a default value for the field. For example,
@Default(value="<defaultValue>")
- @Encrypt: To encrypt the value while saving it into DB.
- @Searchable: To add the custom field into the list of searchable fields in UI.



Once you have added the custom fields into the proxy interface, you can access it in the code as follows:

```
com.citrix.cpbm.access.User user = (com.citrix.cpbm.access.User) CustomProxy.newInstance(new com.vmoop  
user.setAlternatePhone("9886123456");
```

There are three-tile views, one for each customizable object, which is used various places where we create/edit tenant, user or subscription. These three tile views are:

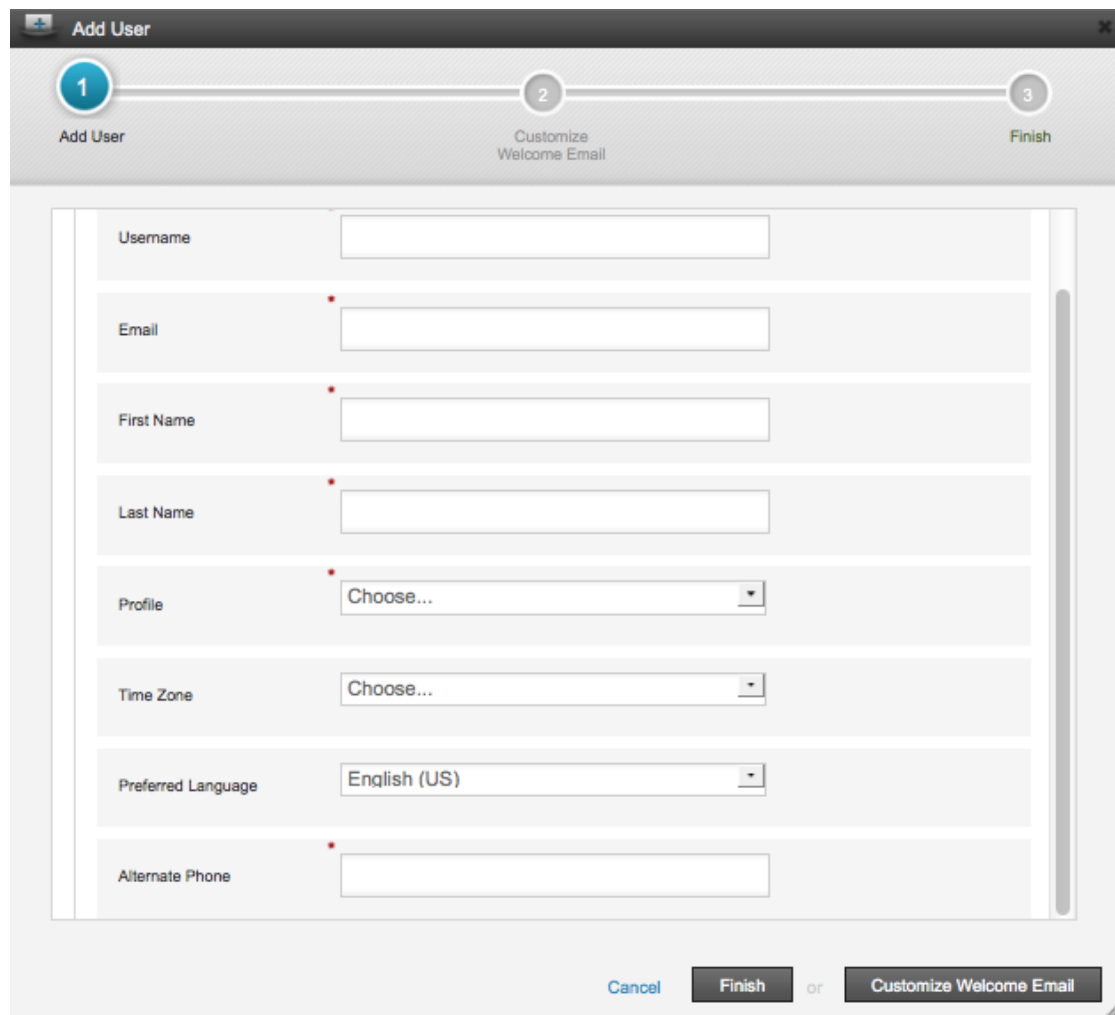
```
<definition name="tenant.custom.fields" template="/WEB-INF/jsp/tiles/shared/custom_fields_tenant.jsp"/>  
<definition name="user.custom.fields" template="/WEB-INF/jsp/tiles/shared/custom_fields_user.jsp"/>  
<definition name="subscription.custom.fields" template="/WEB-INF/jsp/tiles/shared/custom_fields_subscrip
```

For the given example, custom_fields_user.jsp looks as follows:

```

<li>
  <span class="label">
    <form:label path="user.alternatePhone">
      <spring:message code="label.newUserStep1.alternatePhone" />
    </form:label>
  </span>
  <div class="mandatory_wrapper">
    <spring:message code="label.userInfo.alternatePhone.tooltip" var="i18nAlternatePhoneTooltip" />
    <form:input cssClass="text" tabindex="114" path="user.alternatePhone" title="${i18nAlternatePhoneTool
  </div>
  <div class="main_addnew_formbox_errormsg_popup" id="user.alternatePhoneError">
    <form:errors path="user.alternatePhone"></form:errors>
  </div>
</li>

```



Create user with Custom Fields:

Adding custom field validation

Custom field validation can be added into the validation-constraints.xml as described in Validation Rules.

Sample contents of the custom_fields table:

Column Name	Sample value
id	1
prefix	test.user.custom.field
name	test.user.custom.field.AlternatePhone
value	9886123456
entity_id	9
entity_type	User
encrypt	00000000
updated_at	2013-07-13 22:15:57
updated_by	1
created_at	2013-07-13 22:13:08
created_by	1
version	0

You can configure custom field prefix by changing the following properties located at citrix.cpbm.custom.common/src/main/resources/custom.properties:

```
tenant.custom.field.prefix=test.tenant.custom.field  
user.custom.field.prefix=test.user.custom.field  
subscription.custom.field.prefix=test.subscription.custom.field
```

Customizing Resource Properties

Customers can add or update a resource property in the following places:

- Under `citrix.cpbm.custom.common`:
 - For English: `/src/main/resources/OSGI-INF/l10n/resources/CustomApplicationResources.properties`
 - For other languages: `/src/main/resources/OSGI-INF/l10n/resources/CustomApplicationResources_<languageCode>.properties`
- Under `citrix.cpbm.custom.portal`:
 - For English: `src/main/resources/OSGI-INF/l10n/resources/PortalCustomApplicationResources.properties`
 - For other languages: `/src/main/resources/OSGI-INF/l10n/resources/PortalCustomApplicationResources_<languageCode>.properties`

The language code is a valid ISO Language Code. These codes are the lower-case, two-letter codes as defined by ISO-639. You can find a full list of these codes at a number of sites, such as: <http://www.loc.gov/standards/iso639-2/englangn.html>

Typically, you should add all UI related custom resource properties in `citrix.cpbm.custom.portal` bundle and the rest in `citrix.cpbm.custom.common` bundle.

Properties defined in `citrix.cpbm.custom.portal` take precedence over other properties in the system.

Note: Single quote handling in message strings:

For each key-value pair, any single quote in a "value" string must be written as 2 consecutive single quotes to prevent the single quote from acting as a string delimiter.

Customizing Workflows

In CloudPortal™ Business Manager, user can perform certain actions to achieve a desired business purpose. These actions can be considered as business transactions that help in achieving this business purpose. A workflow is a multi-stage process that consists of multiple phases. Every phase consists of a set of activities, (smallest unit of work in a workflow) that should be completed successfully in order to complete the phase. All phases should be completed to complete a business transaction.

You can customize any existing workflow by performing any of the following:

- Move the activities into another phase.
- Change the order of the phases.
- Remove the activities from any phase.

Note: The above mentioned tasks can be edited in workflows.xml

- Add any new activity to any phase.
- Add a new phase with new activities or existing activities.

To perform the above mentioned tasks, follow the steps explained below:

How to create a new activity

1. Create an activity in *citrix.cpbm.custom* bundle by creating a new class which implements the interface "com.citrix.cpbm.workflow.activity.Activity".
2. Create a task within the activity, so that the task will be opened when the activity is started.

Consider the following example where *AddSecondaryEmailActivity* is the activity being created, and *ADD_SEC_EMAIL* is the task within that activity and its status is being checked for completion.

```
package com.citrix.cpbm.workflow.custom.activity;

import com.citrix.cpbm.core.workflow.model.BusinessTransaction;
import com.citrix.cpbm.core.workflow.model.Task;
import com.citrix.cpbm.core.workflow.model.Task.State;
import com.citrix.cpbm.workflow.activity.Activity;
import com.citrix.cpbm.workflow.activity.WorkflowActivityExecutionException;
import com.vmops.service.AuthorityService;
import com.vmops.service.UserAlertPreferencesService;

public class AddSecondaryEmailActivity implements Activity{

    private AuthorityService authorityService;
    private UserAlertPreferencesService userAlertPreferencesService;
```

```
public void setUserAlertPreferencesService(
    UserAlertPreferencesService userAlertPreferencesService) {
    this.userAlertPreferencesService = userAlertPreferencesService;
}

public AuthorityService getAuthorityService() {
    return authorityService;
}

public void setAuthorityService(AuthorityService authorityService) {
    this.authorityService = authorityService;
}

@Override
public void start(BusinessTransaction businessTransaction,
    Record activityRecord) throws WorkflowActivityExecutionException {

}

@Override
public Status run(BusinessTransaction businessTransaction,
    Record activityRecord) throws WorkflowActivityExecutionException {
    Status activityStatus;
    if (isActivityComplete(businessTransaction, activityRecord)) {
        activityStatus = Status.SUCCESS;
    } else {
        activityStatus = doActivity(businessTransaction, activityRecord);
    }

    if (activityRecord.getTask() == null && Status.WAITING == activityStatus) {
        Task action = createAction(businessTransaction);
        activityRecord.setTask(action);
    }

    return activityStatus;
}

protected Boolean isActivityComplete(BusinessTransaction businessTransaction,
    Record record) throws WorkflowActivityExecutionException {
    if(userAlertPreferencesService.getCount(businessTransaction.getTenant().getOwner())>0){
        return true;
    }else
        return false;
}

public Task createAction(BusinessTransaction businessTransaction)
    throws WorkflowActivityExecutionException {
    Task action = new Task("ADD_SEC_EMAIL", businessTransaction.getTenant(), businessTransaction,
        businessTransaction, State.PENDING, authorityService.findByAuthority("ROLE_ACCOUNT_U
        Task.DisplayMode.POPUP);
    return action;
}

protected Status doActivity(BusinessTransaction businessTransaction,
    Record activityPayload) throws WorkflowActivityExecutionException {
```

```

        Task action = activityPayload.getTask();
        if (action != null) {
            if (action.getState() == Task.State.SUCCESS) {
                return Status.SUCCESS;
            } else if (action.getState() == Task.State.FAILURE) {
                return Status.FAILURE;
            }
        }
        return Status.WAITING;
    }
}

```

3. Add the newly created custom activity bean in `applicationContext-workflow-customizations.xml` file which is in `src/main/resources/META-INF/spring/` of `citrix.cpbm.custom.common` bundle.

For the above example, following bean is added:

```

<bean id="addSecondaryEmailActivity" class="com.citrix.cpbm.workflow.custom.activity.AddSecondaryEmailActivity" >
    <property name="authorityService" ref="authorityService"/>
    <property name="userService" ref="userService"/>
    <property name="userAlertPreferencesService" ref="userAlertPreferencesService"/>
</bean>

```

4. Add the activity reference in `osgi-context.xml` file which is in `src/main/resources/META-INF/spring/` of `citrix.cpbm.custom` bundle.

For the above example, following reference is added:

```

<osgi:service interface="com.citrix.cpbm.workflow.activity.Activity" ref="addSecondaryEmailActivity" />

```

Note: Also publish new service dependencies (if any).

5. Specify a user friendly name for the task label in `CustomApplicationResources.properties` file in `src/main/resources/OSGI-INF/l10n/resources` of `citrix.cpbm.custom` bundle for it to be displayed in the UI task list.

For the above example, following labels are specified to display when secondary email task is created

```

ui.widget.task.type.ADD_SEC_EMAIL.name= Add Secondary Email for {0}

```

```

ui.task.type.ADD_SEC_EMAIL.name= Add Secondary Email for {0}

```

You can add new workflow definitions also which can either replace existing definition or can be started for a new rule by customizing the file `transactionWorkflowMap.xml`. To add a new workflow for a transaction, perform the following steps:

1. Add a new workflow definition to the `workflows.xml` file in `src/main/resources` of `citrix.cpbm.custom`. For example -


```
<Workflow name="credit-card-account-activation">
  <Bucket name="identity-verification" order="1">
    <Activity name="verifyMasterUserEmailActivity" ref="verifyMasterUserEmailActivity" />
    <Activity name="addSecondaryEmailActivity" ref="addSecondaryEmailActivity" />
  </Bucket>
  <Bucket name="billing-verification-and-accounts-creation" order="2">
    <Activity name="collectCreditCardInfoActivity" ref="collectCreditCardInfoActivity" />
    <Activity name="createOSSAccountActivity" ref="createOSSAccountActivity" />
  </Bucket>
</Workflow>
```

2. Add a new rule in transactionWorkflowMap.xml.

To write rules in transaction workflow map (transactionWorkflowMap.xml):

Whenever a business transaction is triggered, the system will apply the applicable rules available in transactionWorkflowMap.xml on the business transaction and if the rule matches, then it will return the corresponding workflow. If multiple rules gets matched, then the system will return the workflow associated with a match that comes first.

For example -

```
<Match type="tenantStateChange" workflow="credit-card-account-activation">
  <Rule>{#transaction.tenantInitialState.name == 'NEW' and #transaction.tenantTargetState.name == 'ACTIVE' and #transaction.tenant.accountType.paymentModes == 2}</Rule>
</Match>
```

In above snippet, when a business transaction of type **tenantStateChange** is triggered for initial state "NEW" and target state "ACTIVE" for a tenant having payment mode as 2 (which internally means credit card), then system returns the **credit-card-account-activation** workflow.

Writing a rule

For writing a rule, any direct or transitive property available in business transaction object can be used.

Following are the properties available directly or transitively in different business transactions:

TenantStateChangeTransaction

Name	Type
tenant	com.vmops.model. Tenant
tenantInitialState	com.vmops.model.Tenant.State
tenantTargetState	com.vmops.model.Tenant.State

TenantAccountTypeConversionTransaction

Name	Type
tenant	com.vmops.model. Tenant

accountTypeInitial	com.vmops.model.AccountType
accountType	com.vmops.model.AccountType.State
paymentInfo	String

PaymentInfoChangeTransaction

Name	Type
tenant	com.vmops.model.Tenant
accountTypeInitial	com.vmops.model.AccountType
accountType	com.vmops.model.AccountType.State
paymentInfo	java.lang.String

SubscriptionActivationTransaction

Name	Type
tenant	com.vmops.model.Tenant
subscription	com.vmops.model.Subscription
subscriptionInitialState	com.vmops.model.Subscription.State
subscriptionTargetState	com.vmops.model.Subscription.State
isProvisioned	java.lang.String

com.vmops.model.Tenant

Name	Type
sourceChannel	com.vmops.model.Channel
accountType	com.vmops.model.AccountType
spendLimit	java.math.BigDecimal
initialDeposit	java.math.BigDecimal
address	com.vmops.model.Address
currency	com.vmops.model.CurrencyValue
state	com.vmops.model.Tenant.State
createdAt	java.util.Date

Subscription

Name	Type
state	com.vmops.model.Subscription.State
tenant	com.vmops.model.Tenant
productBundle	com.vmops.model.ProductBundle
createdAt	java.util.Date
derivedFrom	com.vmops.model.Subscription

AccountType

Name	Type
------	------

name	java.lang.String
paymentModes	java.lang.Long
trial	boolean
paymentInfoRequired	boolean
autoPayRequired	boolean
instantpayRequired	boolean
vextendCredit	boolean
selfRegistrationAllowed	boolean
manualRegistrationAllowed	boolean
notionalBilling	boolean
manualActivation	boolean

Channel

Name	Type
name	java.lang. String
code	java.lang. String

ProductBundle

Name	Type
name	java.lang. String
code	java.lang. String

com.vmops.model.CurrencyValue

Name	Type
currencyCode	java.lang. String
currencyName	java.lang. String

com.vmops.model.Address

Name	Type
city	java.lang. String
state	java.lang. String
postalCode	java.lang. String
country	java.lang. String

Creating Custom Action URL Resolver

If the Task created Type is NAVIGATE, then you have to specify the action Resolver URL. Following are the steps to create custom action URL resolver:

1. Create a CustomActionUrlResolver.java and populate the parameters list.

For Example : Below is the sample code of creating the customActionUrlResolver for above created adding secondary email activity.

```
/* Copyright (C) 2012 Citrix, Inc. All rights reserved. */
package com.citrix.cpbm.workflow.custom.resolver;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Set;

import com.citrix.cpbm.core.workflow.model.Task;
import com.citrix.cpbm.workflow.pendingaction.resolver.impl.AbstractActionUrlResolver;

public class CustomActionUrlResolver extends AbstractActionUrlResolver {

    private Map<String, String> actionUrls = new HashMap<String, String>();

    @Override
    public Set<String> getSupportedTypes() {
        return actionUrls.keySet();
    }

    public void setActionUrls(Map<String, String> actionUrls) {
        this.actionUrls = actionUrls;
    }

    @Override
    public List<String> getParams(Task action) {
        List<String> params = new ArrayList<String>();
        if ("ADD_SEC_EMAIL".equalsIgnoreCase(action.getType())) {
            params.add(action.getTenant().getOwner().getUuid());
        }
        return params;
    }

    @Override
    public String getActionUrl(Task action) {
        return actionUrls.get(action.getType());
    }
}
```

2. Add the bean for the newly created CustomActionUrlResolver in *applicationContext-workflow-customizations.xml* file which is under *src/main/resources/META-INF/spring/* of *citrix.cpbm.custom* bundle. Provide the target URL in the value field of <entry>

For the above example, following bean is added for newly created sample CustomActionUrlResolver.

```
<bean id="customActionUrlResolver" class="com.citrix.cpbm.workflow.custom.resolver.CustomActionUrlResolver">
    <property name="actionUrls">
        <map>
            <entry key="ADD_SEC_EMAIL" value="users/%1$s/myprofile"></entry>
        </map>
    </property>
</bean>
```

```
    </property>  
</bean>
```

3. Add the reference of the newly created CustomActionUrlResolver in osgi-context.xml file which is under src/main/resources/META-INF/spring/ of *citrix.cpbm.custom* bundle.

For the above example, following reference is added:

```
<osgi:service interface="com.citrix.cpbm.workflow.pendingaction.resolver.ActionUrlResolver" ref="custo
```

Adding an existing activity to a workflow

All the activities that can be used in a workflow are defined in the applicationContext-workflow-customizations.xml file. You can add a pre-existing activity and customize a workflow.

Adding an existing activity into an existing workflow

You can customize the account activation workflow to include the activity `verifyDepositActivity` for that account type. The `verifyDepositActivity` is an available pre-existing activity.

Workflow definition for credit-card-account-activation workflow before adding `verifyDepositActivity`:

```
<Workflow name="credit-card-account-activation">
  <Bucket name="identity-verification" order="1">
    <Activity name="verifyMasterUserEmailActivity" ref="verifyMasterUserEmailActivity"/>
  </Bucket>
  <Bucket name="billing-verification-and-accounts-creation" order="2">
    <Activity name="collectCreditCardInfoActivity" ref="collectCreditCardInfoActivity"/>
    <Activity name="createOSSAccountActivity" ref="createOSSAccountActivity"/>
  </Bucket>
</Workflow>
```

Workflow definition for credit-card-account-activation workflow after adding `verifyDepositActivity`:

```
<Workflow name="credit-card-account-activation">
  <Bucket name="identity-verification" order="1">
    <Activity name="verifyMasterUserEmailActivity" ref="verifyMasterUserEmailActivity"/>
  </Bucket>
  <Bucket name="billing-verification-and-accounts-creation" order="2">
    <Activity name="collectCreditCardInfoActivity" ref="collectCreditCardInfoActivity"/>
    <Activity name="createOSSAccountActivity" ref="createOSSAccountActivity"/>
    <Activity name="verifyDepositActivity" ref="verifyDepositActivity"/>
  </Bucket>
</Workflow>
```

The activity is added to Bucket(Phase) 2 of the workflow.

If required, the activity can be added either to any existing phase or to a new phase in the workflow.

Note: If the activity creates an approval task then make sure that there is a template of type "approval" available in the email_templates table. The template name must be of structure as shown below:

TASK_<UPPERCASE OF BUSINESS_TRANSACTION type which triggers the workflow>_<UPPERCASE of TASK type
For example: TASK_TENANTSTATECHANGE_FINANCE_APPROVAL.

To add a new template, refer to [Customizing Task Templates](#).

Creating approval workflow for subscription creation with specific bundles

To create an approval workflow for subscription creation with specific bundles, you have to create a new activity, a new workflow definition (that has the new activity), and a new rule associated to the new workflow.

Assumption: Product bundles that require an approval workflow will have a code which follows a standard format. For example, xxx_APP_REQ.

1. Create a new activity which creates an approval task with desired role (role who can approve/reject the workflow). Steps to create new activity are available at: [Customizing workflows](#).
2. Create a new workflow definition. Steps to create a new workflow definition are available at: [Customizing workflows](#).
3. Add the activity (created in Step 1) into the workflow definition created in Step 2.
4. Create a new rule associated to the new workflow definition created in Step 2. This rule will check for the bundle code having specific format. Steps to create new rule and what fields are available in rule expression, are available at: [Customizing workflows](#).

Example

```
<Rule>{#transaction.subscription.productBundle.code matches '^.*APP_REQ$'}</Rule>
```

To perform above steps you have to use the [Customization SDK Guide](#).

Customizing Task Templates

You can customize a template when a new task type is defined or an activity which creates the task is associated to another business transaction.

Defining a new template

You can insert new custom template into `cloud_portal.email_templates` table directly from the back end.

```
INSERT INTO email_templates ("template_name", template_text, category, locale, template_name_locale) V
```

TEMPLATE_NAME: This must have the structure as shown below.

TASK_<UPPERCASE OF BUSINESS_TRANSACTION type>_<UPPERCASE of TASK type>

For example: TASK_TENANTSTATECHANGE_FINANCE_APPROVAL

TEMPLATE_TEXT: Its Apache Velocity template and can have HTML with CSS Styles.

Check existing template text as an example.

CATEGORY: It must be 4 for approval tasks.

LOCALE: It is java locale having naming conventions such as en_US and so on.

TEMPLATE_NAME_LOCALE: It is same as TEMPLATE_NAME but having the locale appended to it. For example, TASK_TENANTSTATECHANGE_FINANCE_APPROVAL_EN

Editing an existing template

- Login as root.
- Navigate to Administration -> content templates -> Emails.
- Select the required template to edit.
- Click edit, then you will see a popup with the text to be edited. Click Save on edit.

The structure of the template name that CloudPortal™ Business Manager allows to show on the POPUP UI will be:

TASK_<UPPERCASE OF BUSINESS_TRANSACTION type>_<UPPERCASE of TASK type> for example TASK_TENANTS

The available model objects to define/edit these content templates are:

- task
- task.businessTransaction

For other available model objects from businessTransaction, refer [Writing a rule](#).

Customizing Topic Subscriber

TopicSubscriber is a spring ApplicationContextListener, which listens to all events that are raised by CloudPortal Business Manager.

This listener listen to the following events:

- TenantActivation
- TenantLock
- TenantSuspension
- TenantTermination
- TenantReactivation
- SubscriptionCreation
- SubscriptionDeletion
- SubscriptionTermination
- UserCreation
- UserDeletion
- CloudServiceEvent
- SubscriptionChangeEvent
- UtilitySubscriptionEvent
- MissingProductEvent
- UserDeactivateEmail
- SubscriptionActivation
- AccountConverted
- CreditCardFraudCheckEvent
- TrialExpiryWarning
- TrialSuspended
- DeviceFraudDetectionEvent
- ProvisionResourceFailedEvent

- PasswordResetRequest
- DepositReceived VerifyEmailRequest
- EmailVerified

By default, a concrete class "CustomTopicSubscriber" implementing "TopicSubscriber" is available in the package com.citrix.cpbm.admin of custom bundle citrix.cpbm.custom.common. But this class is not published into OSGi registry.

A customer or a service provider can add the desired business logic in any of the receive methods which listen to CloudPortal Business Manager events and publish this into OSGi registry.

To publish follow below steps:

1. Create bean for the class:

Uncomment "`<!-- <bean class="com.citrix.cpbm.admin.CustomTopicSubscriber" id="customTopicSubscriber" /> -->`" in applicationContext-admin-customizations.xml available at src/main/resources/META-INF/spring in bundle citrix.cpbm.custom.common.

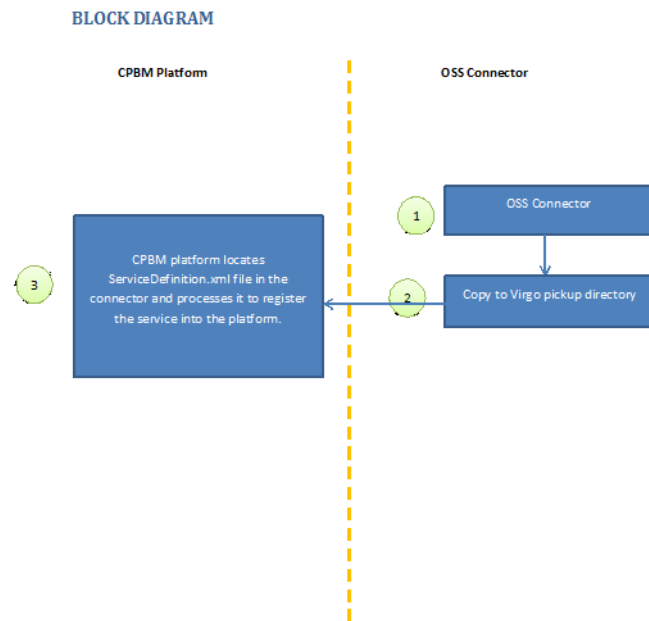
2. Publish the bean to OSGi registry:

Uncomment "`<!-- <osgi:service interface="com.citrix.cpbm.subscriber.TopicSubscriber" ref="customTopicSubscriber"/> -->`" in osgi-context.xml available at src/main/resources/META-INF/spring in bundle citrix.cpbm.custom.common.

After making the necessary changes, the customer or service provider has to rebuild the custom packages as documented in [Installing and building custom components](#).

Customizing Payment Gateway and Ticketing Service

This topic is for the developers/partners who would implement OSS Connector for CloudPortal Business Manager.



System requirements

The document assumes that the developer machine is installed with the following system requirements (and the developer is also aware of the same):

- Windows XP (32-Bit), Windows Vista or Windows 7 (32- or 64-Bit)
- JDK Version 1.6
 - Set **JAVA_HOME** environment variable to the root of the JDK install path
 - Add **%JAVA_HOME%\bin** to the **PATH** environment variable
- Spring Tool Suite (STS) Version 3.1.0 or higher
- Apache Maven Version 3
 - Add **<Maven Install Path>\bin** to the **PATH** environment variable
- Running installation of CloudPortal Business Manager

Pre-Requisites

Read the [Account Lifecycle](#), [User Lifecycle](#), and [Service Installation and Bootstrap](#) sections of the [Connector Development Kit](#) before proceeding.

Downloading and Installing the OSS Service SDK

- Download the CloudPortal Business Manager Cloud Service SDK.
- Unzip the content to a local directory in your system.
- Double-click `cssdk-standard.jar` or run the command `java -jar cssdk-standard.jar` to launch CloudPortal Business Manager Cloud Service SDK installer.

Note: For downloading and installing CloudPortal Business Manager, refer to the Installation and Administration Guide.

Creating the CS SDK Maven STS Project

- Start SpringSource Tool Suite.
- Go to File > New > Project.
- Select Maven > Maven Project.
- In the New Maven Project Wizard Screen, make sure Create a simple project is unchecked.
- Click Next.
- In the Catalog selection, select Default Local.
- From the table select item with:
 - Group Id: **com.citrix.cpbm**
 - Artifact Id:
 - For Payment gateway: **com-citrix-cpbm-payment-gateway-archetype**
 - For Ticketing Service: **com-citrix-cpbm-ticketing-service-archetype**
- Click Next.
- Enter the values for the following fields:
 - GroupId - A group under which all services are offered. For instance, **com.salesforce** in case of Ticketing Service.
 - ArtifactId - Name of the services, which the above group offers. Example, **com.salesforce.ticket** in case of Ticketing Service for Salesforce.
 - Version - Version of the service offered. For example, for Salesforce with API version 27, the version is 27.
 - Package - Base Java package for the maven project **.com.salesforce.ticket** in case of Salesforce ticketing service.
- Click **Finish**. This will create a OSS SDK Project.

Note: Updating Maven dependencies on the first project creation might take some time. This is a one-time update.

Various package structure and their responsibilities

Referring to the example mentioned above, we specified `com.salesforce.ticket` to be the base package for the service that this connector would offer. Taking `com.salesforce.ticket` as the base, the script will generate the below Java packages.

- `com.salesforce.ticket.connector`: Contains Connector Code
- `com.salesforce.ticket.service`: Contains Implementation of Salesforce ticketing Service
- `com.salesforce.ticket.lifecycle`: Contains various Life Cycle Handlers

Default classes generated by the project and their functions

The project will generate three packages and will have default classes included in these packages. You must rename these classes to suit the functionality provided by the class. And also as a note, it is a default project structure generated for the ease of use and so packages/classes can be reorganized.

Java package	Class Name	Functionality
<code>com.salesforce.ticket.connector</code>		
	XServiceConnectorImpl	Implementation class for com.citrix.cpbm.platform.spi.CloudServiceAdapter
<code>com.salesforce.ticket.service</code>		
	XServiceImpl	Implements respective OSS Service. In case of ticketing implements TicketService. Provides the actual service that is offered by this bundle.
<code>com.salesforce.ticket.lifecycle</code>		
	XAccountLifeCycleHandler	Implements com.citrix.cpbm.platform.spi.AccountLifeCycleHandler.
	XUserLifeCycleHandler	Implements com.citrix.cpbm.platform.spi.UserLifeCycleHandler

Note: You must rename the X prefix in the generated class to a suitable class name.

ServiceDefinition.xml

ServiceDefinition.xml file is located inside `/com.salesforce.ticket/src/main/resources/ServiceDefinition.xml`. When the mvn install is run, the config file is copied to the root folder of the package. By default, CloudPortal Business Manager platform searches for "ServiceDefinition.xml" file in the root folder of the deployed bundle. On locating the file, CloudPortal Business Manager parses the xml file and executes the steps to register the OSS Service.

OSS Service Connector

You may rename `com.salesforce.ticket.connector.XServiceConnectorImpl` to an appropriate Cloud Service Connector Implementation and provide its corresponding implementation.

CloudPortal Business Manager invokes this initialize method whenever required, to retrieve a OSS Service instance. On this instance, corresponding lifecycle handlers and collectors are invoked.

Configuring Connector and ServiceDefinition.xml

At this stage, we are ready with the connector and various class implementations required by the OSS Service connector.

Locate the sample ServiceDefinition.xml file and modify the parameters as per your needs. Most important parameters would be the Connector xml tag, ref attribute in the Connector xml tag and vendor version. Ref attribute should be the bean name of the implementation of "CloudConnectorFactory" interface, and version in the vendor should match with the version defined in the pom.xml file of the STS project.

Building OSGI Bundle from the project

Run `mvn clean install` at the root of the project path. An OSGI bundle jar will be generated in the target folder.

Publishing the OSGI Bundle to CloudPortal Business Manager

- Stop CloudPortal Business Manager Development Serve by running the command:
`service cloud-portal stop`
- Copy the new bundle into `/usr/local/vts3/connectors_ext` directory on the CPBM server.
- Add an entry for the new bundle into the plan
`/usr/local/vts3/connectors_ext/com.citrix.cpbm.connector.ext.plan`.
- Start CloudPortal Business Manager Development Server by running the command:
`service cloud-portal start`

Customizing Reports

CloudPortal Business Manager supports creation of custom reports. Custom reports can be scheduled for periodic generation and can be generated on demand from the CloudPortal Business Manager UI.

You can create the following two categories of custom reports:

- **Daily Custom Reports:** These reports are generated every day at the scheduled time that is defined in the cron trigger "dailyJobTrigger" in CloudPortal Business Manager. The default scheduled cron expression for this job is "0 15 10 * * ?"
- **Monthly Custom Reports:** These reports are generated once a month at the scheduled time that is defined in the cron trigger "monthlyJobTrigger" in CloudPortal Business Manager. The default scheduled cron expression for this job is "0 15 10 1 * ?"

Information about the custom reports is saved in the following tables under "cloud_reports" schema:

- **custom_reports:** Information about the new report that is customized is stored in this table. It has the following columns:
 - **name:** name of the custom report
 - **class:** implementation class name of the custom report including package name which is the fully qualified class name
- **sched_custom_reports:** Information about custom report that you define is stored in this table. It has the following columns:
 - **customreportid:** ID of the report in custom_reports table
 - **frequency:** type of custom report, daily or monthly
 - **deliveryURL:** url to deliver the generated custom report

The custom reports are delivered to the "deliveryURL" mentioned in the sched_custom_reports table in the following two ways:

- FTP
- Local File System

The format of the delivery URL is as follows <TransferType>://<FilePath Based on the TransferType>.

FTP

ftp://abc(ftpuserid):abc(password)@localhost(ftphost)/pub(ftpdirectorypath)/

Local File System

file:///E:/Work/reports_sequence(File system Path platform specific)/

Adding a new Custom Report Class to CloudPortal Business Manager

There are two ways in which you can create a custom report, by implementing the "GenericReport" or by extending the "AbstractReport."

Generic Report Interface

CloudPortal Business Manager defines the following interface to support creation of custom reports:

```
package com.vmops.portal.reports;
public interface GenericReport {
    public Report getReport();
    public Report getFusionReport();
}
```

The above interface is directly implemented and the object `com.vmops.model.Report` is returned. For custom reports, you can add implementation for the first method declaration and leave for the second method declaration blank.

Abstract Report Class

CloudPortal Business Manager has an abstraction layer to implement custom reports. The abstract class, that implements the above interface, is as follows:

```
public abstract class AbstractReport
```

This allows the following class member variables to be accessed by the sub class:

- `reportDataSource`: Underlying data base connection source
- `report`: Instance of the `com.vmops.model.Report`
- `recordCount`: Number of records added to the Report File till now

If Custom Report class extends this abstract class, the following declarations from this abstract class have to be implemented:

- `protected abstract Statement getStatement(Connection connection) throws SQLException;`

The abstract class gets the statement that needs to be executed to fetch the records from the underlying database `cloud_portal` which are going to be included in the final report. Because only the sub class knows how to fetch the records from the underlying database, the sub class implements this method. For example,

```
StringBuilder query = new StringBuilder("SELECT * FROM users WHERE id > 10 LIMIT 10");  
Return connection.prepareStatement(query.toString());
```

- protected abstract ResultSet executeQuery(Statement statement) throws SQLException

Once the statement is got from the sub class, the statement is executed. Because the statement has been created by the sub class, only the sub class knows how to execute the statement. Therefore, the sub class defines this method also. Once it gets the required set of results from the above execution, the abstract class generates the report. As part of the report file, the first record is the header of that report and last record is the footer of that report if any. And all results fit in between these two records. Therefore, the abstract class gets the Header and Footer from the sub class because only the sub class knows what should be there in these two records. Also, only the sub class knows how to create a records from the results above.

In the above example, the implementation of this method looks like this:

```
PreparedStatement pstmt = (PreparedStatement) statement;  
return pstmt.executeQuery();
```

- protected abstract String getHeader(Statement pstmt) throws SQLException

As mentioned above, only sub class knows what the header for this report should be. So, the sub class defines this method and if there is no header you can return null. This will be called for each report file only once. For example, return "column1, column2, column3,...";

- protected abstract List<String> getRecord(Statement statement, ResultSet rs) throws SQLException

This is called by the abstract class for every next call of ResultSet. Therefore, the pointer in ResultSet always points to the current row in the result set. Generally, only one record is returned for one ResultSet. However, based on the ResultSet, at times more than one record can be returned. The underlying implementation adds all records from this list to the report that is going to be generated.

- protected abstract getFooter(Statement statement, ResultSet rs) throws SQLException

Because only the sub class knows what should be the footer for this report, the sub class should define this method and if there is no footer you can return null. This will be called for each report file only once. For example, return "End of the Report";

- protected abstract void setFileType();

Only the sub class knows what type of file is going to be generated at the end, as records are returned by the sub class in a specific format. For example xml, csv, and so on. For example, report.setFileType("csv");

- protected abstract void updateWithLastAddedRecrod();

Once the report is generated, the abstract class should return a set of attributes as mentioned in the table above. The only information that is put in the attributes is the last added record. And this is used by the report service to generate multiple report files instead of generating one big report file. The Report service passes back this attribute in params to tell the corresponding report that records are already included till that last check point. Therefore, the Custom report generates report for records

after that record.

If sub class does not use the key **limit** which gets passed as part of the **params** (the variable in `com.vmops.model.Report`) or not even getting passed then leave empty implementation for this method in sub class as all records are included in a single report file. For example, `this.report.getParams().put("last_record_id", lastAddedRecord); // Where lastAddedRecord is maintained by the sub class as part of each getRecord() call.`

- protected abstract Object `getUniqueld(ResultSet rs)` throws `SQLException`

Currently, as this is not being called, there is a blank implementation for this.

Report Class

Both the above mentioned methods return the `com.vmops.model.Report` object.

The following are the key variables declared in `com.vmops.model.Report`:

Variable	Definition
Format	Format the file content. Currently this is not set.
reportFile	File that is going to be generated. It is populated by the abstract layer if at all extending the abstract class to implement a custom report.
fileType	Format of the file which is generated, for example csv, xml, and this is the file extension used to name the file once it is generated.
params	<p>Initial parameters passed to generate a report file. Currently the following will be passed:</p> <p>type: type of the custom report (DAILY/MONTHLY)</p> <p>sequenceService: bean reference for sequenceService</p> <p>config: bean reference to Config Service</p> <p>messageSource: bean reference to messageSource service</p> <p>For Daily Type Custom Reports, date: date for which report need to be generated</p> <p>For Monthly Type Custom Reports, month: month for which report need to be generated, year: year for which report need to be generated</p> <p>limit: number of records need to be generated per file if at all passed by the caller</p>
attributes	<p>Attributes that are populated at the end of the report generation. Currently, only the following attribute:</p> <p>number_of_records: number of records generated as part of the custom report file</p>

Custom reports, created by either implementing `GenericReport` interface or by extending `AbstractReport` class, must be added into the `custom_reports` and `sched_custom_reports`

table as shown below:

Run the following queries:

- `INSERT INTO custom_reports (name, class) VALUES ('<Report Name>', '<Fully qualified name of Report Class>');`
- `INSERT INTO sched_custom_reports (customreportid, frequency, deliveryurl) VALUES (<ID of the above insert> '<MONTHLY/DAILY>', '<The FTP path to transfer as in <TransferType>://<FilePath Based on the TransferType>');`

Scheduling custom reports

By default, CloudPortal Business Manager schedules two types of cron jobs to enable custom report scheduling. You do not have to schedule each custom report individually in quartz. You can add Custom reports to these scheduled cron jobs.

The two types of cron jobs are as follows:

- `dailyJobTrigger`: The cron expression for this is "0 15 10 * * ?"
- `monthlyJobTrigger`: The cron expression for this is "0 15 10 1 * ?"

These expressions can be customized in the application context file `applicationContext-customJobs.xml` available in the `citrix.cpbm.custom.common` bundle.

The daily scheduler generates all daily reports together once every day and the monthly scheduler generates all monthly reports together once every month, by reading information from the above entries.

Generating custom reports from the UI

You can generate and view custom reports on an ad-hoc basis from CloudPortal Business Manager UI.

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Reports > Custom Reports.
3. Select the report that you want to view and click Generate.

You can also download these reports and forward them via email.

Customizing Jobs

To create a custom job follow the steps explained below:

1. Create a custom job file in citrix.cpbm.custom.common bundle.
2. In the newly created custom job file, extend AbstractPortalJob and provide an implementation for the abstract run method.

Below example shows the creation of custom job called "ResetLoginFailedAttempts" which will reset the login failed attempts for any user to 0 if the value exceeds 1, the logic for the same was implemented in run method.

```
package com.citrix.cpbm.custom.job;
import java.util.List;
import org.quartz.JobExecutionContext;
import org.quartz.JobExecutionException;
import com.vmops.admin.jobs.AbstractPortalJob;
import com.vmops.model.User;
import com.vmops.model.UserAlertPreferences;
import com.vmops.service.UserAlertPreferencesService;
import com.vmops.service.UserService;
public class ResetLoginFailedAttemptsJob extends AbstractPortalJob {
    @Override
    public void run(JobExecutionContext context) throws JobExecutionException {
        System.err.println("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ in custom job");
        List<User> usersList = userService.listUsers(0, 1000, null, null, true, null);
        for(User user : usersList){
            int loginAttempt = user.getFailedLoginAttempts();
            if(loginAttempt>1){
                user.setFailedLoginAttempts(0);
                userService.update(user);
            }
        }
    }
}
```

3. Create a bean for the newly created custom job implementation class in applicationContext-Jobs.xml file present in src/main/resources/META-INF/spring.

Below is the bean added for the above created custom job - "ResetLoginFailedAttempts".

```
<bean name="resetLoginFailedAttemptsJob" class="org.springframework.scheduling.quartz.JobDetailBean"
    <property name="jobClass" value="com.citrix.cpbm.custom.job.ResetLoginFailedAttemptsJob"/>
    <property name="requestsRecovery" value="false"/>
    <property name="volatility" value="false" />
</bean>
```


4. Create a trigger bean for the newly created custom job and specify job reference, time interval and other parameters.

Below is the trigger bean created for the above custom job.

```
<bean id="resetLoginFailedAttemptsJobTrigger" class="org.springframework.scheduling.quartz.SimpleTrigger" />
  <property name="jobDetail" ref="resetLoginFailedAttemptsJob" />
  <property name="startDelay" value="10000" />
  <property name="repeatInterval" value="1200000" />
  <property name="volatility" value="false" />
  <property name="misfireInstructionName" value="MISFIRE_INSTRUCTION_RESCHEDULE_NEXT_WITH_EXCESS_INTERVAL" />
</bean>
```

5. Add the newly created trigger reference in applicationContext-scheduler.xml file present in src/main/resources/META-INF/spring under "triggers" property list.

Below sample shows the adding of trigger reference:

```
<ref bean="resetLoginFailedAttemptsJobTrigger"/>
```

Customizing Country List

Customers can customize the list of countries to be displayed in the address and the order in which they appear.

CloudPortal Business Manager reads the country list from table *countries* under *cloud_portal* schema. This table has a column *country_order* of type INTEGER which decides if the country be displayed and in what order in UI.

If the value of the column *country_order* is NULL, then the corresponding country will not be displayed in UI. If the value is not null, then it shows the order in which the corresponding country will be displayed in UI.

If you want to change the value of the column, then you can do so by writing a change set in liquibase xml file located in *citrix.cpbm.custom.common* bundle at *src/main/resources/citrix/cpbm/custom/db/portal/custom-db-config-v0.1.xml*.

Note: Liquibase is a tool which CloudPortal Business Manager uses to populate base data or to perform any upgrade/migration during CloudPortal Business Manager upgrade. More information about liquibase can be found at: <http://www.liquibase.org/>

Sample changeset shown below is to change order of a country with code 'CA':

```
<changeSet id="1373880932" author="Smith">
  <update tableName="countries">
    <column name="country_order" value="13"/>
    <where>country_code_2 = 'CA'</where>
  </update>
</changeSet>
```

Customizing Models, DAOs, and Services

You can add custom tables into CloudPortal Business Manager and extend this to use it in CloudPortal Business Manager UI by following these steps:

1. Create schema and insert default values using liquibase.
2. Create model objects.
3. Write DAOs to query the model objects (using hibernate).
4. Write services to perform business operations.
5. Use these services in web controllers to create a view.

Step 1. Create schema and insert default values using liquibase

To do any sql operation into CloudPortal Business Manager DB customization, you can write your liquibase change set into `citrix.cpbm.custom.common/src/main/resources/citrix/cpbm/custom/db/portal/custom-db-config-v0.1.xml`.

A sample liquibase change set looks as follows:

```
<changeSet id="1366711278999-1" author="authorname">
  <createTable tableName="user_login_audit">
    <column name="id" autoIncrement="true" type="BIGINT">
      <constraints nullable="false" primaryKey="true"/>
    </column>
    <column name="userid" type="BIGINT">
      <constraints nullable="false"/>
    </column>
    <column name="remote_ip" type="VARCHAR(100)">
      <constraints nullable="true"/>
    </column>
    <column name="created_at" type="DATETIME" remarks="The time at which user logged in.">
      <constraints nullable="false"/>
    </column>
    <column name="version" type="INT" defaultValue="0"/>
  </createTable>
  <modifySql dbms="mysql">
    <append value=" ENGINE=InnoDB DEFAULT CHARSET=utf8"/>
  </modifySql>
</changeSet>
```

Note: For more information on writing liquibase change set, please refer to <http://www.liquibase.org/documentation/changes/index.html>.

Step 2. Create model objects

To write the model object for this newly created table, add your java file under `citrix.cpbm.custom.model/src/main/java/com/vmops/custom/model` package. All new custom model should implement `com.vmops.model.EntityBase`.

An example model class looks as follows:

```
/* Copyright (C) 2013 Citrix, Inc. All rights reserved. */

package com.citrix.cpbm.custom.model;
import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Version;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;

import com.vmops.model.EntityBase;
import com.vmops.model.User;

/**
 * UserLoginAudit model to capture user login details like
 * <ul>
 * <li>ip address from which the user has logged in.</li>
 * <li>time at which user has logged in.</li>
 * </ul>
 *
 * @author authername
 */
@Entity
@Table(name = "user_login_audit")
public class UserLoginAudit implements EntityBase {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /**
     * Primary key.
     */
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id = 0L;

    /** The user. */
    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "userid", nullable = false, updatable = false)
    @NotNull
```

```
private User user;

/** The remote ip. */
@Column(name = "remote_ip", length = 255)
@Size(max = 100)
private String remotelp;

/** The created at. */
@Column(name = "created_at", nullable = false, updatable = false)
@Temporal(TemporalType.TIMESTAMP)
@NotNull
private Date createdAt = new Date();

/**
 * Version number. used for Optimistic Concurrency Control.
 */
@Column(name = "version")
@Version
private long version = 0;

/**
 * Instantiates a new user login audit.
 */
public UserLoginAudit() {
    super();
}

/**
 * Instantiates a new user login audit.
 *
 * @param user the user
 * @param remotelp the remote ip
 */
public UserLoginAudit(User user, String remotelp) {
    super();
    this.user = user;
    this.remotelp = remotelp;
}

/**
 * Gets the id.
 *
 * @return the id
 * @see com.vmops.model.EntityBase#getId()
 */
@Override
public Long getId() {
    return null;
}

/**
 * Sets the id.
 *
 * @param id the new id
 */
public void setId(Long id) {
```

```
    this.id = id;
}

/**
 * Gets the user.
 *
 * @return the user
 */
public User getUser() {
    return this.user;
}

/**
 * Sets the user.
 *
 * @param user the new user
 */
public void setUser(User user) {
    this.user = user;
}

/**
 * Gets the remote ip.
 *
 * @return the remote ip
 */
public String getRemotelp() {
    return this.remotelp;
}

/**
 * Sets the remote ip.
 *
 * @param remotelp the new remote ip
 */
public void setRemotelp(String remotelp) {
    this.remotelp = remotelp;
}

/**
 * Gets the created at.
 *
 * @return the created at
 */
public Date getCreatedAt() {
    return this.createdAt;
}

/**
 * Sets the created at.
 *
 * @param createdAt the new created at
 */
public void setCreatedAt(Date createdAt) {
    this.createdAt = createdAt;
}
```

```
/**
 * Gets the version.
 *
 * @return the version
 */
public long getVersion() {
    return this.version;
}

/**
 * Sets the version.
 *
 * @param version the new version
 */
public void setVersion(long version) {
    this.version = version;
}

/**
 * Hash code.
 *
 * @return the int
 * @see java.lang.Object#hashCode()
 */
@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((this.createdAt == null) ? 0 : this.createdAt.hashCode());
    result = prime * result + ((this.remotelp == null) ? 0 : this.remotelp.hashCode());
    result = prime * result + ((this.user == null) ? 0 : this.user.hashCode());
    return result;
}

/**
 * Equals.
 *
 * @param obj the obj
 * @return true, if successful
 * @see java.lang.Object#equals(java.lang.Object)
 */
@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    UserLoginAudit other = (UserLoginAudit) obj;
    if (this.createdAt == null) {
        if (other.createdAt != null) {
```

```
        return false;
    }
} else if (!this.createdAt.equals(other.createdAt)) {
    return false;
}
if (this.remotelp == null) {
    if (other.remotelp != null) {
        return false;
    }
} else if (!this.remotelp.equals(other.remotelp)) {
    return false;
}
if (this.user == null) {
    if (other.user != null) {
        return false;
    }
} else if (!this.user.equals(other.user)) {
    return false;
}
return true;
}

/**
 * To string.
 *
 * @return the string
 * @see java.lang.Object#toString()
 */
@Override
public String toString() {
    StringBuilder builder = new StringBuilder();
    builder.append("UserLoginAudit [");
    if (this.id != null) {
        builder.append("id=");
        builder.append(this.id);
        builder.append(", ");
    }
    if (this.user != null) {
        builder.append("user=");
        builder.append(this.user);
        builder.append(", ");
    }
    if (this.remotelp != null) {
        builder.append("remotelp=");
        builder.append(this.remotelp);
        builder.append(", ");
    }
    if (this.createdAt != null) {
        builder.append("createdAt=");
        builder.append(this.createdAt);
        builder.append(", ");
    }
    builder.append("version=");
    builder.append(this.version);
    builder.append("]");
    return builder.toString();
}
```



```
}  
  
}
```

Step 3. Write DAOs to query the model objects (using hibernate)

To write custom DAO interface, create your DAO interface under `citrix.cpbm.custom.common/src/main/java/com/citrix/cpbm/custom/persistence` package. You have to extend the `GenericDAO` interface.

`GenericDAO` represents a DAO interface to interact with model objects persisted in the database. Typically, DAOs are created only for so-called root objects: model objects that are directly accessible. Dependent objects are traversed from these root objects.

An example of the DAO interface and its implementation is shown below:

```
/* Copyright (C) 2013 Citrix, Inc. All rights reserved. */  
package com.citrix.cpbm.custom.persistence;  
  
import java.util.List;  
  
import com.citrix.cpbm.custom.model.UserLoginAudit;  
import com.vmops.model.User;  
import com.vmops.persistence.GenericDAO;  
  
/**  
 * The Interface UserLoginAuditDAO.  
 *  
 * @author authorname  
 */  
public interface UserLoginAuditDAO extends GenericDAO<UserLoginAudit> {  
  
    /**  
     * Find login audit records.  
     *  
     * @param user the user  
     * @return the list  
     */  
    List<UserLoginAudit> findLoginAuditRecords(User user);  
  
}
```



```
/* Copyright (C) 2013 Citrix, Inc. All rights reserved. */  
package com.citrix.cpbm.custom.persistence.hibernate;  
  
import java.util.List;  
import java.util.Map;  
  
import org.springframework.stereotype.Repository;  
  
import com.citrix.cpbm.custom.persistence.UserLoginAuditDAO;
```

```
import com.citrix.cpbm.custom.model.UserLoginAudit;
import com.vmops.model.User;
import com.vmops.persistence.hibernate.GenericHibernateDAO;

/**
 * The Class UserLoginAuditDAOImpl.
 *
 * @author authername
 */
@Repository("userLoginAuditDAO")
public class UserLoginAuditDAOImpl extends GenericHibernateDAO<UserLoginAudit> implements UserLoginA

/**
 * Find login audit records.
 *
 * @param user the user
 * @return the list
 * @see com.citrix.cpbm.custom.persistence.UserLoginAuditDAO#findLoginAuditRecords(com.vmops.model.
 */
@Override
public List<UserLoginAudit> findLoginAuditRecords(User user) {
    Map<String, Object> conditions = getConditionsTemplate();
    conditions.put("user", user);
    return findByCriteria(null, conditions);
}
}
```

Step 4. Write services to perform business operations

To create custom service layer API, you have to write your service interface under `citrix.cpbm.custom.common/src/main/java/com/custom/service`.

An example test service and its implementation is as shown below:

```
/* Copyright (C) 2013 Citrix, Inc. All rights reserved. */
package com.citrix.cpbm.custom.service;

import java.util.Date;
import java.util.List;

import com.citrix.cpbm.custom.model.UserLoginAudit;
import com.vmops.model.User;

/**
 * The Interface UserLoginAuditService.
 *
 * @author authername
 */
public interface UserLoginAuditService {

/**
 * Find user login audit records.
```

```
*
* @param date the date
* @return the list
*/
public List<UserLoginAudit> findUserLoginAuditRecords(Date date);

/**
 * Find user login audit records.
 *
 * @param user the user
 * @return the list
 */
public List<UserLoginAudit> findUserLoginAuditRecords(User user);

public void saveAudit(UserLoginAudit userLoginAudit);
}

/* Copyright (C) 2013 Citrix, Inc. All rights reserved. */
package com.citrix.cpbm.custom.service.impl;

import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.citrix.cpbm.custom.persistence.UserLoginAuditDAO;
import com.citrix.cpbm.custom.service.UserLoginAuditService;
import com.citrix.cpbm.custom.model.UserLoginAudit;
import com.vmops.model.User;

/**
 * The Class UserLoginAuditServiceImpl.
 *
 * @author authername
 */
@Service("userLoginAuditService")
public class UserLoginAuditServiceImpl implements UserLoginAuditService {

    /** The user login audit dao. */
    @Autowired
    private UserLoginAuditDAO userLoginAuditDAO;

    /**
     * Find user login audit records.
     *
     * @param date the date
     * @return the list
     * @see com.citrix.cpbm.custom.service.UserLoginAuditService#findUserLoginAuditRecords(java.util.Date)
     */
}
```

```

@Override
public List<UserLoginAudit> findUserLoginAuditRecords(Date date) {
    Map<String, Object> conditions = new HashMap<String, Object>();
    conditions.put("createdAt", date);
    return userLoginAuditDAO.findByCriteria(null, conditions);
}

/**
 * Find user login audit records.
 *
 * @param user the user
 * @return the list
 * @see com.citrix.cpbm.custom.service.UserLoginAuditService#findUserLoginAuditRecords(com.vmops.mo
 */
@Override
public List<UserLoginAudit> findUserLoginAuditRecords(User user) {
    return userLoginAuditDAO.findLoginAuditRecords(user);
}

/**
 * Save audit.
 *
 * @param userLoginAudit the user login audit
 * @see com.citrix.cpbm.custom.service.UserLoginAuditService#saveAudit(com.citrix.cpbm.custom.model.L
 */
@Override
public void saveAudit(UserLoginAudit userLoginAudit) {
    userLoginAuditDAO.save(userLoginAudit);
}
}
}

```

Once you add a service, you have to export this service from `citrix.cpbm.custom.common` bundle, so that it can be used in `citrix.cpbm.custom.portal` bundle, where we have web controllers and views. To do this, you need to make an entry into `citrix.cpbm.custom.common/src/main/resources/META-INF/spring/osgi-context.xml`.

A sample entry looks as follows:

```
<osgi:service interface="com.citrix.cpbm.custom.service.UserLoginAuditService" ref="userLoginAuditService"
```

Step 5. Use these services in web controllers to create a view

To use the services exported from `citrix.cpbm.custom.common` bundle, you have to first import them into `citrix.cpbm.custom.portal` bundle. To do this you have to make an entry into `citrix.cpbm.custom.portal/src/main/resources/WEB-INF/portal-osgi-context.xml`.

A sample entry looks as follows:

```
<osgi:reference interface="com.citrix.cpbm.custom.service.UserLoginAuditService" id="userLoginAuditService"
```

To use the imported service in web controller and pass it to the view, access it by any other spring bean.

For example,

```
@Autowired
@Autowired private UserLoginAuditService userLoginAuditService;
.
.
.
/**
 * creates a login audit record for current logged in user.
 * @param user
 * @param request
 */
private void createLoginAuditRecord(User user, HttpServletRequest request) {
    UserLoginAudit userLoginAudit = new UserLoginAudit(user, getRemoteUserIp(request));
    userLoginAuditService.saveAudit(userLoginAudit);
}
```

Miscellaneous

- The custom liquibase setting is available at `citrix.cpbm.custom.common/src/main/resources/META-INF/spring/applicationContext-custom-liquibase.xml`.

The bean which runs the custom liquibase file looks as follows:

```
<bean id="liquibaseportalcustomization" class="liquibase.integration.spring.SpringLiquibase">
  <property name="dataSource" ref="dataSource" />
  <property name="contexts" value="${liquibase.custom.context:basedata}"/>
  <property name="changeLog" value="classpath:citrix/cpbm/custom/db/portal/cloud_portal-custom.x
  <property name="changeLogParameters">
    <map>
      <entry key="key-needed" value="to-avoid-npe-bug"></entry>
      <entry key="portal_db" value="${profile.jdbc.database.schemaname}"/>
      <entry key="field_encryption_key" value="${field.encryption.key}"/>
    </map>
  </property>
  <property name="resourceLoader" ref="resourceLoader" />
</bean>
```

The properties used in this bean are defined in `cloud.properties` located in your `virgo-home>/repository/prop/cloud.properties`.

- Custom DAOs and Services bean are created in `citrix.cpbm.custom.common/src/main/resources/META-INF/spring/applicationContext-custom-service.xml`.

```
<context:annotation-config />
<context:component-scan base-package="com.citrix.cpbm.custom.persistence" />
<context:component-scan base-package="com.citrix.cpbm.custom.service" />
```

If you choose to change the custom service package name, you need to make the package entry in the exported package list in MANIFEST.MF file of citrix.cpbm.custom.common bundle as well imported package list in MANIFEST.MF file of citrix.cpbm.custom.portal bundle. Also, as specified above in Step 4 and Step 5, you need to make the necessary changes in osgi-context.xml of both these bundles.

Customizing Email Templates

Customers can modify existing email templates or add new email templates that can be used to send emails from CloudPortal Business Manager.

Follow these steps to modify existing email templates:

1. Log in to CloudPortal Business Manager UI a ROOT user.
2. Navigate to Administration > Content Templates.
3. Select the category of content templates that you want to modify.
4. Select any template from the list of templates (in the left pane).
5. Move the cursor over the manage icon (that looks like a gear wheel) and click Edit.

A dialog box opens that contains the code for the template.

6. Make the changes as required and click OK.

This modified template can now be used to send emails. This template gets persisted into the DB so it does not get reverted when CloudPortal Business Manager is re-started.

Adding new templates:

To add new templates, you must have the development setup as described in [Installing and building custom components](#).

1. Add new templates in liquibase xml file located in citrix.cpbm.custom.common bundle located at
src/main/resources/citrix/cpbm/custom/db/portal/custom-db-config-v0.1.xml.

A sample liquibase change set for adding a custom email template will look as follows:

```
<changeSet id="1373375974" author="John">
  <insert tableName="email_templates">
    <column name="template_name" value="SAMPLE_TEMPLATE"/>
    <column name="template_text">
      <![CDATA[
        #parse("EMAIL_HEADER_EN")

        <h2 style="${style_h2}">Hello #if($user.firstName)${user.firstName}#{else}${user.username}#e

        <p style="${style_p}">
          This is a sample email.
        </p>
        <br>

        #parse("SIGNATURE_EN")
        #parse("EMAIL_FOOTER_EN")
      ]]>
    
```

```
    ]]>
  </column>
  <column name="category" valueNumeric="0"/>
  <column name="locale" value="en_US"/>
  <column name="template_name_locale" value="SAMPLE_TEMPLATE_EN"/>
</insert>
</changeSet>
```

2. In code, an email template can be sent using emailService as shown below:

```
/**
 * @param template - the template location
 * @param model - model object required for the template
 * @param subjectKey - key code in messageSource for the subject
 * @param user - user to whom the email has to be sent
 */
public void sendEmail(final String template, final Map<String, Object> model, final String subjectKey, final String user) {
    // ...
}

eg:
Map<String, Object> model = new HashMap<String, Object>();
model.put("locale", LocaleUtils.toLocale(config.getDefaultLocale()));
model.put("user", user);
emailService.sendEmail("SAMPLE_TEMPLATE", model, "email.template.SAMPLE_TEMPLATE.subject", user);
```

Note: The above code uses the same value of template_name as provided in Step 1.

Stop emails sent with an email template

Service provider can choose not to send email associated with any email template by simply making the content of the email template as NULL or empty. When content of any email template is nullified, then the email is not sent and is ignored.

Note: Liquibase is a tool which CloudPortal Business Manager uses to populate base data or perform any upgrade or migrate operations during a CloudPortal Business Manager upgrade. More information about liquibase can be found at: <http://www.liquibase.org/>.

Objects and Properties for Email Templates

The following table provides the list of Email Templates with the available objects and properties in each template. The properties within the template can be directly used. For example:

`${customEmailText}` : for the property `customEmailText` in `VERIFY_EMAIL` template.

The objects in the templates are used to get the property values within them for the template. For example:

`($user.firstName)` : to get the first name of the user, use the object `user` from the template.

Email Template Name	Objects	Properties
WELCOME_EMAIL	User :: user	
VERIFY_EMAIL	User :: user	verifyEmailLink customSubject customEmailText subject
VERIFY_ALERT_EMAIL	User :: user	verifyEmailLink locale
RESET_PASSWORD	User :: user	resetPasswordLink
TRIAL_SERVICE_EXPIRY_WARNING	TrialAccount :: trial User :: user Date :: days	
TRIAL_SERVICE_EXPIRED	TrialAccount :: trial User :: user Locale :: locale	termination
TRIAL_SERVICE_EXPIRED_NOTIFICATION	TrialAccount :: trial User :: user	termination
CREDIT_LIMIT_CROSSED_SERVICE_SUSPENDED_NOTIFICATION	User :: user Tenant :: tenant	

Objects and Properties for Email Templates

SPEND_BUDGET_WARNING	User :: user SpendCheckCriteria :: criteria	threshold percentConsumption
CREDIT_LIMIT_CREDIT_NOTIFY	User :: user Tenant :: tenant	threshold percentConsumption
CREDIT_LIMIT_BREACH_ALERT	User :: user Tenant :: tenant	
CREDIT_LIMIT_CROSSED_SERVICE_SUSPENDED	User :: user Tenant :: tenant	
PAYMENT_RECEIVED	User :: user	currency paymentAmount
PAYMENT_FAILED	User :: user Tenant :: tenant	updateBillingLink restricteddays suspendeddays
PAYMENT_FAILED_NOTIFY	User :: user Tenant :: tenant	body subject
DEPOSIT_RECEIVED	User :: user Locale :: locale	currency depositAmount
SUPPORT_TICKET_NOTIFY	Ticket :: ticket	name time
CORPORATE_ACCOUNT_REQUEST	Tenant :: tenant User : user	userAgent remoteAddress referer time
CREDIT_EXPOSURE_MARKER_LEVEL	Tenant :: tenant User : user	
SEND_PROMOTIONAL_TOKEN	PromotionSignup :: signup	registerlink
ACCOUNT_RESTRICT	User :: user Tenant :: tenant Locale :: locale	

Objects and Properties for Email Templates

ACCOUNT_ACTIVATE	User :: user Tenant :: tenant Locale :: locale	
ACCOUNT_SUSPEND	User :: user Tenant :: tenant Locale :: locale	
ACCOUNT_TERMINATE	User :: user Tenant :: tenant Locale :: locale	
ACCOUNT_REACTIVATE	User :: user Tenant :: tenant Locale :: locale	
ACCOUNT_CONVERTED	User :: user Local :: locale	accountTypeName
ACCOUNT_CONVERTED_ADMIN	User :: user	accountTypeName
PENDING_CHANGE_REJECTED	User :: user PendingChangeRejected :: pendingChangeRejected	
VM_CREATION	User :: user	password username vmDisplayName
REPORT	URL :: reportImageURL	subject
JOB_FAILED	Date :: startTime Date :: endTime	jobName desc hostname
DESTROY_VM_FAILED	User :: user	message
ORPHANED_VM_CREATED		vmid accountName domainId accountId

Objects and Properties for Email Templates

CREDITCARD_FRAUD_CHECK_EVENT	FraudCheckInfo :: fraudCheckInfo Date :: date	details
ACCOUNT_ACTIVATION_REQUEST	User :: user Tenant :: tenant	
DEVICE_FRAUD_DETECTION_EVENT	DeviceFraudDetectionAudit :: fraudDetectionAudit	login email phone firstName lastName
CREDIT_LIMIT_WARNING	User :: user SpendCheckCriteria :: criteria	threshold percentConsumption
VM_PASSWORD_RESET		password firstName username vmDisplayName
INCOMPATIBLE_RESOURCE_TYPE	Locale :: locale	bundleName utilitySubscriptionUrl oldSubscriptionUrl resourceType
INTERNAL_APPROVAL	User :: user Tenant :: tenant	noOfRetries
INVOICE_EMAIL	Locale :: locale	billing_start_date billing_start_time billing_end_date billing_end_time subject

Objects and Properties for Email Templates

JOB_WARNING	Date :: startTime Date :: endTime	desc hostname subject jobName
MISSING_PRODUCT	Set<Long> :: usageDatas	
NEW_RESOURCE_TYPE	User :: user Locale :: locale	utilitySubscriptionUrl resourceType resourceHandle
PENDING_CHANGE_REJECTED	PendingChangeRejected :: pendingChangeRejected User :: user	
PROVISION_RESOURCE_FAILED	Subscription :: subscription	message updateResource
SUBSCRIPTION_ACTIVATION_SERVICE_PROVIDER_NOTIFICATION	List<Entitlements> :: entitlements ServiceInstance :: serviceInstance Subscription :: subscription ProductBundle :: productBundle Tenant :: tenant User :: user ServiceInstance :: service	userRecurringChargeName spRecurringChargeName recurringPrice userOneTimeChargeName spOneTimeChargeName oneTimePrice

Objects and Properties for Email Templates

SUBSCRIPTION_ACTIVATION_USER_NOTIFICATION	List<Entitlements> :: entitlements ServiceInstance :: serviceInstance Subscription :: subscription ProductBundle :: productBundle Tenant :: tenant User :: user ServiceInstance :: service	userRecurringChargeName spRecurringChargeName recurringPrice userOneTimeChargeName spOneTimeChargeName oneTimePrice cloudProvider
SUBSCRIPTION_TERMINATION_SERVICE_PROVIDER_NOTIFICATION	Subscription :: subscription ProductBundle :: productBundle Tenant :: tenant User :: user ServiceInstance :: service	
SUBSCRIPTION_TERMINATION_USER_NOTIFICATION	Subscription :: subscription ProductBundle :: productBundle Tenant :: tenant User :: user ServiceInstance :: service	cloudProvider
TENANT_ADMIN_APPROVAL	User :: user Tenant :: tenant	noOfRetries
USER_ACTIVATE_EMAIL	User :: user User :: masterUser Locale :: locale	
USER_DEACTIVATE_EMAIL	User :: user User :: masterUser Locale :: locale	

Objects and Properties for Email Templates

For each object in the template, the following table provides details of the properties/child objects for the specified templates.

Object	Child Objects	Properties
SpendCheckCriteria		displayName
fraudCheckInfo	User :: user	ipaddress city country postalCode riskScore
DeviceFraudDetectionAudit	Date :: callDate	policyScore reasonCode riskRating reviewStatus deviceId fuzzyDeviceId accountName
Address		street1 street2 city state postalCode country

Objects and Properties for Email Templates

CreditCard	Address :: creditCardAddress	creditCardType creditCardNumber creditCardExpirationMonth creditCardExpirationYear creditCardMaskedExpDate creditCardCVV nameOnCard firstNameOnCard lastNameOnCard
pendingChangeRejected	User :: user	changedStateTo changeType memo
URL		host port path
PromotionSignup	PromotionToken :: promotionToken Currency :: currency CampaignPromotion :: campaignPromotion	name company email phone

Objects and Properties for Email Templates

<p>Tenant</p>	<p>Channel :: sourceChannel</p> <p>AccountType :: accountType</p> <p>Date :: lastUsageUpdate</p> <p>User :: owner</p> <p>Address :: address</p> <p>Address :: secondaryAddress</p> <p>CurrencyValue :: currency</p> <p>TrialAccount :: trialAccount</p> <p>DepositRecord :: deposit</p>	<p>name</p> <p>spendLimit</p> <p>prepaidBalance</p> <p>initialDeposit</p> <p>spendBudget</p> <p>currentUsage</p> <p>oldUsage</p> <p>syncBillingAddress</p> <p>state</p> <p>memo</p> <p>paymentAlertsCounter</p> <p>accountSuspensionCounter</p> <p>usageQuery</p> <p>remoteAddress</p> <p>ownerProvisionalId</p> <p>ownerSpendLimit</p> <p>ownerCurrentUsage</p> <p>ownerOldUsage</p> <p>isManualActivation</p>
<p>Ticket</p>	<p>Tenant :: tenant</p> <p>User :: owner</p> <p>Date :: createdAt</p> <p>Date :: lastUpdateAt</p>	<p>caseNumber</p> <p>caseId</p> <p>creatorName</p> <p>companyName</p> <p>email</p> <p>description</p> <p>subject</p> <p>status</p> <p>caseOrigin</p>

Objects and Properties for Email Templates

TrialAccount	Tenant :: tenant Date :: expiryDate	state
User	Channel :: sourceChannel Address :: address Tenant :: tenant Profile :: profile	firstName lastName email username phone timeZone enabled locked trial spendBudget currentUsage oldUsage spendLimit countryCode failedLoginAttempts locale
PromotionToken	User :: createBy Date :: createDate CampaignPromotion :: campaignPromotion	code
Currency		currencyCode

Objects and Properties for Email Templates

CampaignPromotion	Date :: createDate Set<CampaignPromotionsInChannels> :: campaignPromotionsInChannels Date :: startDate Date :: endDate	code title durationDays discountType percentOff maxAccounts promoCode customerType enabled trial
Channel		uuid name domainId description code removed removedBy

AccountType		name displayName description paymentModes trial paymentInfoRequired autoPayRequired instantInvoicing private boolean notionalBilling manualActivation defaultSelfRegistered defaultRegistered maxUsers perAccountVmCount perAccountPublicIPCount perAccountVolumeCount perAccountSnapshotCount perAccountTemplateCount perUserVmCount perUserPublicIPCount perUserVolumeCount perUserSnapshotCount perUserTemplateCount accountRestrictionGracePeriod preAuthRequired depositRequired
-------------	--	--

Objects and Properties for Email Templates

CurrencyValue		currencyCode currencyName rank longCurrencyName sign altSign active conversionFactor
TrialAccount	Tenant :: tenant Date :: expiryDate Date :: lastChange	state
DepositRecord	Date :: receivedOn Tenant :: tenant User :: recordedBy	amount
Profile		name description operationsProfile userCRUD accountAdmin
CampaignPromotionsInChannels	CampaignPromotion :: campaignPromotion Channel :: channel	

Objects and Properties for Email Templates

CampaignPromotion	Set<CampaignPromotionsInChannels> :: campaignPromotionsInChannels Date :: startDate Date :: endDate Date :: createDate	code title durationDays discountType percentOff maxAccounts promoCode customerType enabled trial
Subscription	Tenant :: tenant User :: user Date :: terminationDate ProductBundle :: productBundle Date :: createdAt User :: createdBy List<Entitlement> :: effectiveEntitlements Date :: activationDate Date :: nextRenewalDate ServiceInstance :: serviceInstance ServiceResourceType :: resourceType	uuid state vmlId preAuthTransId preAuthAmount configurationData
Entitlements	ProductBundle :: productBundle Product :: product	includedUnits allowedUnlimitedEntitlement
ServiceInstance	Service :: service	uuid name description code

Objects and Properties for Email Templates

ProductBundle	ServiceInstance :: serviceInstanceId ServiceResourceType :: resourceType User :: createdBy User :: updatedBy Date :: createdAt Date :: updatedAt Date :: removed User :: removedBy	name description code trialEligibility publish notificationEnabled
ServiceResourceType	Service :: service	resourceTypeName constraint
Product	User :: createdBy User :: updatedBy Date :: createdAt Date :: updatedAt Date :: removed User :: removedBy ServiceInstance :: serviceInstance	name code price isReplaced description allowedUnlimitedEntitlement
Service	Date :: createdAt Date :: updatedAt Set<ServiceInstance> :: serviceInstances List<ServiceResourceType> :: serviceResourceTypes	uuid category type vendor locales singleton adapterRef serviceName vendorVersion enabled

Customizing Connector UI

From CloudPortal Business Manager 2.1 onwards, the connector UI can be customized using an OSGI fragment bundle. This bundle will contain assets (jsp, js, css, tiles, MVC controllers) for connector UI customization.

Creation of connector fragment from Archetypes

1. Run the *cssdk-standard.jar* installer and follow on-screen instructions.
2. Create a new connector fragment maven project by following the steps:
 - a. Goto File->New->Project.
 - b. Select Maven->Maven Project.
 - c. In the New Maven Project Wizard Screen, make sure Create a simple project is unchecked.
 - d. Click **Next**.
 - e. In the Catalog selection, select Newly added Archetype in Catalog dropdown.
 - f. From the table select item with
 - i. Group Id: *com.citrix.cpbm*
 - ii. Artifact Id: *com-citrix-cpbm-connector-fragment-archetype*
 - g. Click **Next** .
 - h. Enter the values for the following fields:
 - i. GroupId - A group under which all services are offered. For instance, *org.apache* in case of CloudStack.
 - ii. ArtifactId - Name of the services which the above group offers *.cloudstack.fragment* in case of Apache CloudStack.
 - iii. Version - Version of the service offered. For example, for *cloudstack.fragment*, the version is 1.0.0.
 - iv. Package - Base Java package for the maven project *.org.apache.cloudstack* in case of Apache CloudStack
 - i. Click **Finish**. This will create a Connector Fragment Project.

Note: If project is not getting created try restarting the STS once.

Default Structure

META-INF
readme.txt
src


```
main
  java
    base package name
    ExampleController.java
  resources
    css
      ${artifactId}
      example.css
    js
      ${artifactId}
      example.js
    images
      ${artifactId}
      example.png
  META-INF
  MANIFEST.MF
  WEB-INF
  default
    ${artifactId}-tiles-defs.xml
  jsp
    ${artifactId}
    example.jsp
  applicationContext-${artifactId}-fragment.xml
build.properties
pom.xml
template.mf
```

File Description

- **ExampleController.java:** It is a Spring MVC controller which will contain the url mapping, which is returned from the connector's view resolver. It returns a tile view.
- **MANIFEST.MF:** It is the bundle's manifest specifying the bundle's information along with the fragment's host, it also contains package imports (if any).
- **applicationContext-\${artifactId}-fragment.xml:** Spring Context file whose beans will be loaded by the fragment's host. Any spring beans should go here.
- **\${artifactId}-tiles-defs.xml:** An xml configuration file which contains tiles definitions for custom views which is loaded by fragment-host by file name.
- **example.jsp:** An example jsp file which specifies which JSTL tags can be used in the JSP and also provides an example to include a css or a javascript file.
- **example.js:** An example js file.
- **example.css:** An example css file.
- **example.png:** An example image file.
- **pom.xml:** The maven project file for dependency and build management. (Add maven dependency for the connector project for which this fragment will provide views).

Customizing Billing Exports

CloudPortal Business Manager has a built-in functionality of exporting billing information in xml everyday such as account statements, invoices, payment and credit statements, and payment failures.

Billing export is driven by a xml configuration named billing-export-config.xml located in the citrix.cpbm.custom.common bundle. CloudPortal Business Manager allows you to customize various factors before exporting a file. Customization includes the following:

- File name pattern
- Sequencing of records
- sftp/ftp configuration
- batch size
- xslt transformation config file that needs to be applied on the generated xml

Billing export engine



Supported streams

- AccountStatement Stream

The AccountStatement Stream is capable of generating data with different types of sequencing. Sequence can be altered in the configuration settings file. Valid Sequences for AccountStatement Stream are:

- ACCOUNT_TYPE
- CHANNEL
- TENANT

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AccountStatements>
```

```
<AccountStatement>
  <uuid>22c1aede-614d-4be4-8cf3-4737853111a7</uuid>
  <Tenant>
    <Uuid>890077fe-6844-4732-8199-215566da083f</Uuid>
    <Name>nageshp</Name>
    <AccountType>Corporate</AccountType>
    <Channel>Channel-14-8-2013</Channel>
    <State>ACTIVE</State>
    <CreationDate>2013-09-03T08:22:33Z</CreationDate>
    <Currency>US Dollar</Currency>
    <Owner>
      <Uuid>d34c5ee3-e3e0-4772-a5fb-d0377ffd6e14</Uuid>
      <Name>nageshp</Name>
      <Username>nageshp</Username>
    </Owner>
    <Address>
      <City>bangalore</City>
      <Country>IN</Country>
      <PostalCode>560073</PostalCode>
      <State>KAR</State>
      <Street1>1stAvenue</Street1>
    </Address>
  </Tenant>
  <State>POSTED</State>
  <StartDate>2013-09-09T00:00:00Z</StartDate>
  <EndDate>2013-09-11T23:59:59Z</EndDate>
  <PostedAt>2013-09-12T01:06:15Z</PostedAt>
  <BalanceForward>-43709.4</BalanceForward>
  <NewCharges>6301.69</NewCharges>
  <Credits>0.0</Credits>
  <Discounts>0.0</Discounts>
  <TotalTaxes>572.88</TotalTaxes>
  <FinalCharges>50011.09</FinalCharges>
  <SubscriptionCharges>137.99</SubscriptionCharges>
  <UtilityCharges>5590.82</UtilityCharges>
  <RenewalCharges>0.0</RenewalCharges>
  <Invoices>
    <Invoice>
      <uuid>750b8fc4-2ffd-474e-b779-c2aac284e4fd</uuid>
      <GenerationDate>2013-09-12T01:06:14Z</GenerationDate>
      <ServiceStartDate>2013-09-11T00:00:00Z</ServiceStartDate>
      <ServiceEndDate>2013-09-11T23:59:59Z</ServiceEndDate>
      <PostedAt>2013-09-12T01:06:14Z</PostedAt>
      <State>Posted</State>
      <RawAmount>2865.39</RawAmount>
      <Discount>0.0</Discount>
      <SubTotal>2865.39</SubTotal>
      <Tax>286.54</Tax>
      <FinalAmount>3151.93</FinalAmount>
      <Subscription/>
      <InvoiceItems>
        <InvoiceItem type="Charge" chargeableUsage="48.0" unitPrice="12.1234">
          <Amount>581.9232</Amount>
          <Description>Charge</Description>
        </InvoiceItem>
        <InvoiceItem type="Charge" chargeableUsage="0.020237457" unitPrice="6.0">
```

```

    <Amount>0.121424735</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="0.0" unitPrice="4.0">
    <Amount>0.0</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="48.0" unitPrice="3.0">
    <Amount>144.0</Amount>
    <Description>Prod-16-8-2013</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="0.0" unitPrice="5.0">
    <Amount>0.0</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="0.0" unitPrice="4.0">
    <Amount>0.0</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="15.026944" unitPrice="80.0">
    <Amount>1202.1555</Amount>
    <Description>Running-VM-CentOS</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="48.0" unitPrice="2.0">
    <Amount>96.0</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="0.032258064" unitPrice="2.0">
    <Amount>0.06451613</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="15.056945" unitPrice="6.0">
    <Amount>90.34167</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="15.026944" unitPrice="7.0">
    <Amount>105.18861</Amount>
    <Description>prod-26-8-2013</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="1.54227E-5" unitPrice="6.0">
    <Amount>9.25362E-5</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="15.026944" unitPrice="5.0">
    <Amount>75.13472</Amount>
    <Description>Prod-27-6-2013</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="0.40475658" unitPrice="0.0">
    <Amount>0.0</Amount>
    <Description>Charge</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="0.030000687" unitPrice="10.0">
    <Amount>0.30000687</Amount>
    <Description>stopped_VM</Description>
</Invoiceltem>
<Invoiceltem type="Charge" chargeableUsage="48.0" unitPrice="10.0">

```

```

        <Amount>480.0</Amount>
        <Description>Prod_RunningVM</Description>
    </InvoiceItem>
    <InvoiceItem type="Charge" chargeableUsage="15.026944" unitPrice="6.0">
        <Amount>90.16167</Amount>
        <Description>Charge</Description>
    </InvoiceItem>
    <InvoiceItem type="Charge" chargeableUsage="0.0" unitPrice="6.0">
        <Amount>0.0</Amount>
        <Description>Charge</Description>
    </InvoiceItem>
    <InvoiceItem type="Charge" chargeableUsage="1.54227E-5" unitPrice="100.0">
        <Amount>0.00154227</Amount>
        <Description>internetcharges</Description>
    </InvoiceItem>
    <InvoiceItem type="Subtotal" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>2865.39</Amount>
        <Description>Subtotal</Description>
    </InvoiceItem>
    <InvoiceItem type="Discounts" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>0.0</Amount>
        <Description>Discounts</Description>
    </InvoiceItem>
    <InvoiceItem type="Tax" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>286.539</Amount>
        <Description>Tax @ 10.00%</Description>
    </InvoiceItem>
    <InvoiceItem type="Taxes" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>286.54</Amount>
        <Description>Taxes</Description>
    </InvoiceItem>
    <InvoiceItem type="Net" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>3151.93</Amount>
        <Description>Net</Description>
    </InvoiceItem>
</InvoiceItems>
</Invoice>
</Invoices>
</AccountStatement>
</AccountStatements>

```

- All invoices for the day:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Invoices>
  <Invoice>
    <uuid>b36d0c37-1324-485b-8049-36b534754986</uuid>
    <GenerationDate>2013-09-12T01:04:11Z</GenerationDate>
    <ServiceStartDate>2013-09-11T00:00:00Z</ServiceStartDate>
    <ServiceEndDate>2013-09-11T23:59:59Z</ServiceEndDate>
    <PostedAt>2013-09-12T01:04:11Z</PostedAt>
    <State>Posted</State>
    <RawAmount>2327.99</RawAmount>
  </Invoice>
</Invoices>

```

```

<Discount>931.2</Discount>
<SubTotal>1396.79</SubTotal>
<Tax>139.68</Tax>
<FinalAmount>1536.47</FinalAmount>
<Subscription/>
<InvoiceItems>
  <InvoiceItem type="Charge" chargeableUsage="48.0" unitPrice="5.0">
    <Amount>240.0</Amount>
    <Description>Prod-14-8-2013</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="0.058015037" unitPrice="5.0">
    <Amount>0.29007518</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="80.36278" unitPrice="12.1234">
    <Amount>974.2701</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="96.0" unitPrice="2.0">
    <Amount>192.0</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="0.032258064" unitPrice="2.0">
    <Amount>0.06451613</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="32.362778" unitPrice="3.0">
    <Amount>97.08833</Amount>
    <Description>Prod-16-8-2013</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="0.0" unitPrice="4.0">
    <Amount>0.0</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="2.580645" unitPrice="5.0">
    <Amount>12.903226</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="1.2903225" unitPrice="6.0">
    <Amount>7.7419353</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="80.36278" unitPrice="10.0">
    <Amount>803.6278</Amount>
    <Description>Prod_RunningVM</Description>
  </InvoiceItem>
  <InvoiceItem type="Charge" chargeableUsage="3.57069E-5" unitPrice="4.0">
    <Amount>1.428276E-4</Amount>
    <Description>Charge</Description>
  </InvoiceItem>
  <InvoiceItem type="Subtotal" chargeableUsage="0.0" unitPrice="0.0">
    <Amount>2327.99</Amount>
    <Description>Subtotal</Description>
  </InvoiceItem>
  <InvoiceItem type="Discount" chargeableUsage="0.0" unitPrice="0.0">
    <Amount>931.196</Amount>

```

```

        <Description>Discount @ 40.0000%</Description>
    </InvoiceItem>
    <InvoiceItem type="Discounts" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>931.2</Amount>
        <Description>Discounts</Description>
    </InvoiceItem>
    <InvoiceItem type="Tax" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>139.679</Amount>
        <Description>Tax @ 10.00%</Description>
    </InvoiceItem>
    <InvoiceItem type="Taxes" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>139.68</Amount>
        <Description>Taxes</Description>
    </InvoiceItem>
    <InvoiceItem type="Net" chargeableUsage="0.0" unitPrice="0.0">
        <Amount>1536.47</Amount>
        <Description>Net</Description>
    </InvoiceItem>
</InvoiceItems>
</Invoice>
</Invoices>

```

- Payments and credits

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Payments>
  <Payment>
    <Uuid>8a80262e-3d52-4079-adda-1301362fa613</Uuid>
    <Tenant>
      <Uuid>8822c76b-0a98-4e37-864a-31508eccec72</Uuid>
      <Name>anusha</Name>
      <AccountType>RETAIL</AccountType>
      <Channel>Channel-14-8-2013</Channel>
      <State>SUSPENDED</State>
      <CreationDate>2013-08-28T09:31:55Z</CreationDate>
      <Currency>US Dollar</Currency>
      <Owner>
        <Uuid>a08a9bb6-6644-4971-8812-8211eb6b2d87</Uuid>
        <Name>anusha b</Name>
        <Username>anusha</Username>
      </Owner>
      <Address>
        <City>bangalore</City>
        <Country>IN</Country>
        <PostalCode>560078</PostalCode>
        <State>KAR</State>
        <Street1>MajesticAvenue</Street1>
      </Address>
    </Tenant>
    <type>AUTO</type>
    <TransactionAmount>2170.04</TransactionAmount>
    <memo>Making Auto Payment</memo>
    <AccountStatement uuid="69911443-bf90-408b-9331-443a0b23a822"
    <PaymentTransaction>

```

```

        <Uuid>257cc015-49a2-40aa-aed5-4a32b6bfe773</Uuid>
        <Type>CHARGE</Type>
        <TransactionId>3789481256760176056470</TransactionId>
        <TransactionTime>2013-09-12T01:05:16Z</TransactionTime>
        <TransactedAmount>2170.04</TransactedAmount>
        <Amount>2170.04</Amount>
        <State>COMPLETED</State>
        <StatusCode>0</StatusCode>
        <Memo>Making Auto Payment</Memo>
    </PaymentTransaction>
</Payment>
<Payments>

```

- Payment failure

```

<PaymentFailures>
  <PaymentFailure>
    <Uuid>f5892323-ad4c-4439-8b46-3687f1e0192d</Uuid>
    <Tenant>
      <Uuid>f132a5e3-f1ae-478b-999f-ddaf68e2b711</Uuid>
      <Name>peter</Name>
      <AccountType>RETAIL</AccountType>
      <Channel>Default</Channel>
      <State>ACTIVE</State>
      <CreationDate>2012-04-30T00:00:00Z</CreationDate>
      <Currency>Euro</Currency>
      <Owner>
        <Uuid>13c1d293-9dbc-4408-abd6-2190e72de091</Uuid>
        <Name>peter</Name>
        <Username>peter</Username>
      </Owner>
      <Address>
        <City>bangalore</City>
        <Country>IN</Country>
        <PostalCode>560002</PostalCode>
        <State>KAR</State>
        <Street1>FrazerTown</Street1>
      </Address>
    </Tenant>
    <type>CHARGE</type>
    <PaymentTransaction>
      <Uuid>f5892323-ad4c-4439-8b46-3687f1e0192d</Uuid>
      <Type>CHARGE</Type>
      <TransactionId>3678348733330176056442</TransactionId>
      <TransactionTime>2013-05-06T09:57:23Z</TransactionTime>
      <TransactedAmount>100.0</TransactedAmount>
      <Amount>100.0</Amount>
      <State>ERRORED</State>
      <StatusCode>0</StatusCode>
      <Memo></Memo>
    </PaymentTransaction>
  </PaymentFailure>
</PaymentFailures>

```


Global stream configuration

There are 2 configurations which can be applied at global level for all the streams:

- `com.citrix.cpbm.portal.billing.export.URL` - URL refers to the configuration to which generated stream file would be uploaded. It has to be a valid URL pattern supported by apache VFS. For example, `sftp://root:root1234@localhost/`.
- `com.citrix.cpbm.portal.billing.export.directory`: Directory (CloudPortal Business Manager installed system) on which all the streams will be generated locally.

Both these settings can be applied from the configuration screen.

Stream configuration

Stream configuration consists of the following:

- **Batch size**: Number of records that should be written to the batch file as output.

- **File:**

- **pattern**: File pattern is the name of the file to be generated during the batch process. For example, `cloudportal_subs_%d.log.%s`, where `%d` gets replaced by the current date and `%s` gets replaced by the last file sequence + 1.

CloudPortal Business Manager supports only `%d` and `%s` replacement of runtime variables in the file name pattern.

- **trasformxsl**: Path of the xslt configuration file, that will be used to generate the batch file in various formats like csv, xslt, and so on.

System has the capability of reading a file from classpath. Path should prefixed with appropriate config:

- **classpath**:

- **Sequence algorithms**: Sequence algorithm defines the logical grouping of billing export data that is by type-1 and then by type-2.

- **Export file**: Value of the element can be:

```
<Export url=""/>
```

- Full URL path, if global settings are not applied.

- Directory location in the remote system assigned to the stream.

This is implemented using Apache VFS. All file systems in Apache VFS that allow file upload and create in a remote system are supported.

- <http://commons.apache.org/proper/commons-vfs/filesystems.html>

- <http://commons.apache.org/proper/commons-vfs/>

- **Configuration:**

```
<BillingExport>
```

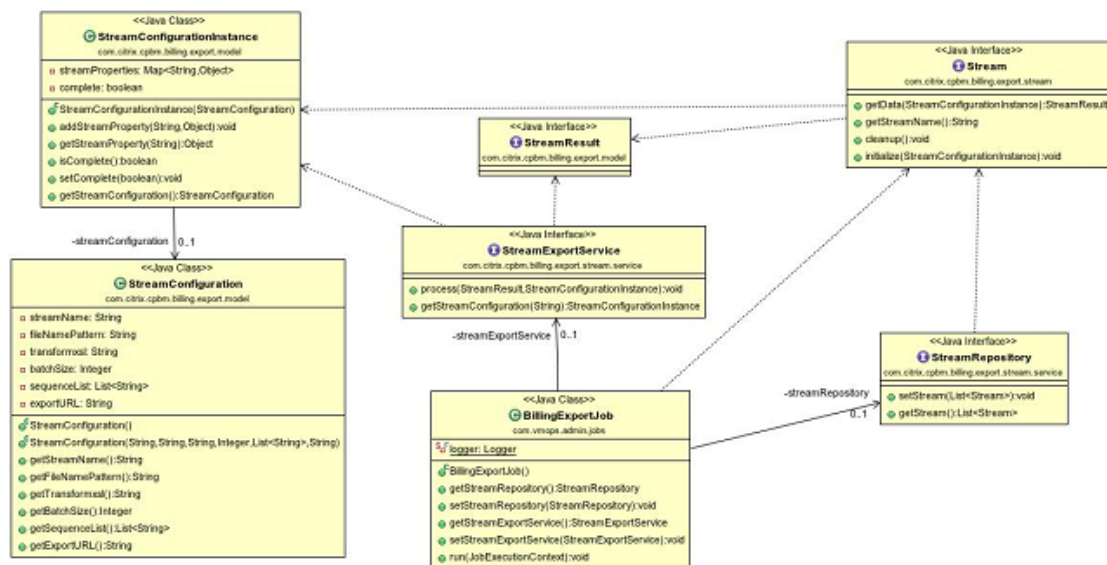
```
  <ExportStream name="AccountStatement">
```

```
    <file pattern="cpbm_account_stmt_%d_%s" transformxsl="classpath:/opt/billing/xsl/transform.xsl"/>
```

```

<batch size="10"/>
<sequence id="channel"/>
<sequence id="accounttype"/>
<Export url=""/>
</ExportStream>
<ExportStream name="invoice">
<file pattern="cpbm_invoice_%d_%s" transformxml="classpath:/com/citrix/cpbm/custom/transform."/>
<batch size="10"/>
<Export url=""/>
</ExportStream>
</BillingExport>
    
```

Class Diagram



Customizing Pricing Filters

Pricing filters can be customized. By customization, CloudPortal Business Manager allows the removal of filters.

Customizing pricing filters consists of the following:

Step 1

Open the applicationContext-admin-customizations.xml file in citrix.cpbm.custom.common bundle.

Step 2

There are 2 spring beans which refer to a collection of discounting engines and tax engines:

- discountEngineList
- taxEngineList

```
<bean name="adhocDiscountEngine" class="${discount.adhoc.engine.implementation:com.vmops.component.  
</bean>  
<bean name="percentDiscountEngine" class="${discount.percent.engine.implementation:com.vmops.comp  
</bean>  
<bean name="amountDiscountEngine" class="${discount.amount.engine.implementation:com.vmops.comp  
<bean name="taxEngine" class="${tax.engine.implementation}" >  
  <property name="flatTaxPercent" value="${tax.engine.flat.percent:10.0}" />  
</bean>
```

```
<bean id="discountEngineList" class="java.util.ArrayList">  
  <constructor-arg>  
    <list>  
      <ref bean="adhocDiscountEngine" />  
      <ref bean="percentDiscountEngine" />  
      <ref bean="amountDiscountEngine" />  
    </list>  
  </constructor-arg>  
</bean>
```

```
<bean id="taxEngineList" class="java.util.ArrayList">  
  <constructor-arg>  
    <list>  
      <ref bean="taxEngine" />  
    </list>  
  </constructor-arg>  
</bean>
```

Step 3

Remove the bean reference from the list which is not required or is not applicable. Any bean reference removed from the above mentioned list will also remove the bean declaration as this bean is not used any more.

Customizing Profiles

Customers can add more profiles to CloudPortal Business Manager, if required.

Below are the steps to create new profile.:

1. Create new entry for the profile in database. Add a liquibase change set to file custom-db-config-v0.1.xml located in bundle citrix.cpbm.custom.common at src/main/resources/citrix/cpbm/custom/db/portal/.

A sample liquibase change set is as follows:

```
<changeSet id="1373375974" author="John">
  <insert tableName="profiles">
    <column name="name" value="SuperUser"/>
    <column name="description" value="This is a super user"/>
    <column name="scope" value="<TENANT/TENANT_ADMIN/GLOBAL/GLOBAL_ADMIN"/>/>
  </insert>
</changeSet>
```

Note: In above example, "SuperUser" is the sample name of the profile and "This is a super user" is the sample description of the profile.

2. Add value of the profile to be displayed in UI. CloudPortal Business Manager looks for value of code "profileName.<profile name>" in application resource. Add a property "profileName.<profile name>" into the application resource. See [Customizing Resource Properties](#) for details.

Example: profileName.SuperUser=Super User

3. Restart the CPBM server:

```
service cloud-portal stop
service cloud-portal start
```

4. Login to CloudPortal Business Manager as Root user and navigate to **Administration > Profiles**.
5. If the scope provided by you in Step 1 is TENANT or TENANT_ADMIN, then your new profile will appear in the Customer tab, else it will appear in the Service Provider tab.
6. Select the new profile and check all the roles you want in the profile and click **Save**.

you can select this profile when creating new users.

To understand the scope in detail, refer to [Security Roles](#).

Note: Liquibase is a tool which CloudPortal Business Manager uses to populate base data or to perform any upgrade/migration during CloudPortal Business Manager upgrade. More information about liquibase can be found at: <http://www.liquibase.org/>

Customizing Account Types

Customer can add more account types to CloudPortal Business Manager, if required.

Below are the steps to create new account type:

1. Create a new entry for the new account type in the table `account_types` under schema `cloud_portal`. Add a liquibase change set to file `custom-db-config-v0.1.xml` located in bundle `citrix.cpbm.custom.common` at `src/main/resources/citrix/cpbm/custom/db/portal/`.

A sample liquibase change set is as follows:

```
<changeSet id="1373375974" author="John">
  <insert tableName="account_types">
<insert tableName="account_types">
  <column name="name" value="CUSTOM_AT"/>
  <column name="display_name" value="Custom Account Type"/>
  <column name="description" value="Custom account type for doc"/>
  <column name="trial" valueBoolean="false"/>
  <column name="payment_info_required" valueBoolean="true"/>
  <column name="instantpay_required" value="0"/>
  <column name="autopay_required" valueBoolean="true"/>
  <column name="extend_credit" valueBoolean="true"/>
  <column name="self_registration_allowed" valueBoolean="true"/>
  <column name="manual_registration_allowed" valueBoolean="true"/>
  <column name="notional_billing" valueBoolean="false"/>
  <column name="manual_activation" valueBoolean="false"/>
  <column name="credit_exposure_breach" valueNumeric="3"/>
  <column name="default_self_registered" valueBoolean="true"/>
  <column name="default_registered" valueBoolean="true"/>
  <column name="payment_modes" valueNumeric="2"/>
  <column name="max_users" valueNumeric="10"/>
  <column name="account_restriction_grace_period" valueNumeric="1"/>
  <column name="pre_auth_required" valueBoolean="false"/>
  <column name="deposit_required" valueBoolean="false"/>
  <column name="enable_secondary_address" valueBoolean="false"/>
  <column name="version" valueNumeric="0"/>
  </insert>
</insert>
</changeSet>
```

Note: In above example "CUSTOM_AT" is sample name , "Custom account type" is sample display name and "Custom account type for doc" is sample description of the account type. You must select other values as per your requirement. For more information, refer to [Account Properties](#).

2. Add required resources into custom application properties. Add the below mentioned properties:

```
registration.accounttype.custom_at=Custom Account Type
registration.accounttype.description.custom_at=Custom account type with special privileges
page.level2.custom_at=Custom Account Type
```

Note: custom_at is shown on above properties as an example only. You should add the name used for the account type and corresponding description.

For details, refer to [Customizing Resource Properties](#).

3. Restart the CPBM server:

```
service cloud-portal stop
service cloud-portal start
```

Note: Liquibase is a tool which CloudPortal Business Manager uses to populate base data or to perform any upgrade/migration during CloudPortal Business Manager upgrade. More information about liquibase can be found at: <http://www.liquibase.org/>

Customizing Discount Adjusters

Customers can change the way discounts are applied to a CPBM invoice.

1. Change the engine to customized implementation:

a. Customizing the way percent discount engine works:

- i. Implement a new DiscountEngine by extending the abstract class DiscountEngine (com.vmops.component.discount.policy.DiscountEngine) and put it inside the citrix.cpbm.custom bundle.
- ii. Edit the cloud.properties at /usr/share/vts3/repository/prop to point to the new entry by editing the property discount.percent.engine.implementation. Point it to the new implementation.

b. Customizing the way amount discount engine works:

- i. Implement a new DiscountEngine by extending the abstract class DiscountEngine (com.vmops.component.discount.policy.DiscountEngine) and put it inside the citrix.cpbm.custom bundle.
- ii. Edit the cloud.properties at /usr/share/vts3/repository/prop to point to the new entry by editing the property discount.amount.engine.implementation. Point it to the new implementation.

c. Customizing the way non promotional discount engine works:

- i. Implement a new DiscountEngine by extending the abstract class DiscountEngine (com.vmops.component.discount.policy.DiscountEngine) and put it inside the citrix.cpbm.custom bundle.
- ii. Edit the cloud.properties at /usr/share/vts3/repository/prop to point to the new entry by editing the property discount.adhoc.engine.implementation. Point it to the new implementation.

Restart the CPBM service.

2. Adding new engines over and above the existing engine:

- a. Implement a new DiscountEngine by extending the abstract class DiscountEngine (com.vmops.component.discount.policy.DiscountEngine) and put it inside the citrix.cpbm.custom bundle.
- b. Add one or more such implementations.
- c. Open the application context file applicationContext-admin-customizations.xml inside the citrix.cpbm.custom bundle.

Define a bean for each implementation:

```
<bean name="discountEngineCustom" class="<Newly added class>" />
```

Add the above created bean to the list 'taxEngineList':


```
<bean id="discountEngineList" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <ref bean="adhocDiscountEngine" />
      <ref bean="percentDiscountEngine" />
      <ref bean="amountDiscountEngine" />
      <ref bean="discountEngineCustom" />
    </list>
  </constructor-arg>
</bean>
```

or

```
<bean id="taxEngineList" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <ref bean="discountEngineCustom" />
    </list>
  </constructor-arg>
</bean>
```

Note: You can remove the default implementation and/or add multiple such beans in the list. Beans will be called in the same order as defined in the list.

d. Restart the CPBM service.

3. Example:

```
public interface RateAdjuster {

  /**
   * @param invoice A read only (not enforced) invoice for the rate adjuster to get info related to this invoice
   *   tenant, billing period etc.
   * @param newInvoiceItems Every RateAdjuster should process the newInvoiceItems and replace them with
   *   for the next adjusters for further processing. The first adjuster sees the invoiceItems of
   *   ChargeType.CHARGE
   * @param serviceStart is the billing period start for the moment, it can be any day in the billing period
   *   decided to support number of days for a promotion (as oppose to number of billing period)
   * @param serviceEnd is the billing period end for the moment, it can be any day in the billing period if
   *   to support number of days for a promotion (as oppose to number of billing period)
   * @param previewMode if true, nothing will be written to db or the billing activity file. Plan to remove
   * @return
   */
}

public abstract BigDecimal compute(Invoice invoice, List<InvoiceItem> invoiceItemsToBeProcessed, Date
  Date serviceEnd, boolean previewMode);

}

public class ExampleDiscountEngine extends DiscountEngine {

  @Override
```

```
public BigDecimal compute(Invoice invoice, List<InvoiceItem> invoiceItemsToBeProcessed, Date service
    Date serviceEnd, boolean previewMode) {
    ..... [Put your implementation here ]
}
}
```

Customizing Tax Adjusters

Customers can change the way tax rates are applied to a CPBM invoice.

1. Change the engine to customized implementation:

- a. Implement a new TaxEngine by implementing the interface TaxEngine (com.vmops.service.TaxEngine) and put it inside the citrix.cpbm.custom bundle.
- b. Edit the cloud.properties at /usr/share/vts3/repository/prop to point to the new entry by editing the property tax.engine.implementation. Point it to the new implementation.
- c. Restart the CPBM service.

2. Adding new engines over and above the existing engine:

- a. Implement a new TaxEngine by implementing the interface TaxEngine (com.vmops.service.TaxEngine) and put it inside the citrix.cpbm.custom bundle.
- b. Add one or more such implementations.
- c. Open the application context file applicationContext-admin-customizations.xml inside the citrix.cpbm.custom bundle.

Define a bean for each implementation:

```
<bean name="taxEngineCustom" class="<New Implementation path>" >
    <property name="flatTaxPercent"
value="{tax.engine.flat.percent:10.0}" />
</bean>
```

Use the tax percent either as defined in cloud.properties or put a value for each implementation.

d. Add the above created bean to the list 'taxEngineList':

```
<bean id="taxEngineList" class="java.util.ArrayList">
    <constructor-arg>
        <list>
            <ref bean="taxEngine" />
            <ref bean="taxEngineCustom" />
        </list>
    </constructor-arg>
</bean>
```

or

```
<bean id="taxEngineList" class="java.util.ArrayList">
    <constructor-arg>
```

```

        <list>
          <ref bean="taxEngineCustom" />
        </list>
      </constructor-arg>
    </bean>

```

e. Restart the CPBM service.

Note: You can remove the default implementation and/or add multiple such beans in the list. Beans will be called in the same order as defined in the list.

3. Example:

```

public interface RateAdjuster {

    /**
     * @param invoice A read only (not enforced) invoice for the rate adjuster to get info related to this invoice
     * tenant, billing period etc.
     * @param newInvoiceItems Every RateAdjuster should process the newInvoiceItems and replaced them
     * for the next adjusters for further processing. The first adjuster sees the invoiceItems of
     * ChargeType.CHARGE
     * @param serviceStart is the billing period start for the moment, it can be any day in the billing period
     * decided to support number of days for a promotion (as oppose to number of billing period)
     * @param serviceEnd is the billing period end for the moment, it can be any day in the billing period if
     * to support number of days for a promotion (as oppose to number of billing period)
     * @param previewMode if true, nothing will be written to db or the billing activity file. Plan to remove
     * @return
     */
    public abstract BigDecimal compute(Invoice invoice, List<InvoiceItem> invoiceItemsToBeProcessed, Date
        serviceEnd, boolean previewMode);

}

```

```

public interface TaxEngine extends RateAdjuster {

    /**
     * Get taxable amount based on tax engine.
     *
     * @param amount
     * @return
     */
    public BigDecimal compute(BigDecimal amount);

}

```

```

public class ExampleTaxEngine extends TaxEngine {

    @Override
    public BigDecimal compute(BigDecimal amount) {
        ..... [Put your implementation here ]
    }
}

```

Migration Guide

This guide describes the upgrade mechanism for CloudPortal™ Business Manager (CPBM) 2.1.x release from CPBM versions 1.4.3 through 1.4.7. It provides the insight into supported upgrade paths, migration strategy, assumptions in existing system before upgrade and impact on customization.

Supported upgrade paths:

1. CloudPortal™ Business Manager (CPBM) versions 1.4.3 through 1.4.7 to CPBM 2.1.
2. Customers who have CPBM version lower than 1.4.3 have to upgrade first to CPBM 1.4.3 in order to upgrade to CPBM 2.1.

Upgrade Mechanism

In a multinode environment, code upgrade (execute the command `./install.sh` with option U) must be performed on all the servers, but the Python script for database (DB) upgrade has to be run on one of the nodes only.

It is assumed that the earlier setup of CloudPortal Business Manager (CPBM) v1.4.3 through 1.4.7 is upgraded to CPBM v2.1.x on the same setup where it was running earlier. No changes to the setup are made during the process of upgrade.

There are lot of changes in database with this upgrade, therefore a forced database dump file will be created under the folder `/usr/share/cloud/portal/upgrade-dump/` at DB server automatically. However, after successful upgrade, this file can be moved to some other location.

As there are lot of changes in database, upgrade can take much longer time than what it used to take earlier. Actual time taken to upgrade depends on the data to be processed.

Rollback mechanism

On upgrade failure, none of the automatic restoration is supported. However, CloudPortal Business Manager database (DB) dump is automatically backed up and stored at `/usr/share/cloud/portal/upgrade-dump/`.

Note: It is strongly recommended to save these backup file(s) for longer time in case of any data issue discovered after migration.

List of Pre-checks

Following list describes some of the pre-conditions required for CloudPortal™ Business Manager upgrade.

- All Products should have some price set in reference catalog.
- Product Bundle can have unlimited entitlement only for the following products - RUNNING, ALLOCATED_VM
- Upgrade should happen on same machine where CloudPortal Business Manager 1.4.x was installed.
- **Authorize.Net** and **CyberSource** are the only supported payment gateways for upgrade. One of them has to be configured in the system for upgrade.

Upgrade steps

A shell script with following commands will be present in the following location:

```
./<CPBM_RPM_UNTAR_LOCATION>/migration/from/2.0/from/1.4/scripts.
```

Note:

- CCP management server should be up and running while performing the CloudPortal Business Manager upgrade.
- Make sure that you take a backup of CloudPortal Business Manager DB.
- In case of upgrade failure, automatic system rollback will not be supported. It is therefore strongly recommended to take the complete VM backup or snapshot of the CloudPortal Business Manager server and DB.

Pre-requisites:

In case of clustered environment, code upgrade must be performed on all the servers, but the DB upgrade (Pythonscript) has to be run only on one of the nodes.

Ensure that the cloud-portal service is stopped before the upgrade is performed. In a clustered setup, the cloud-portal service must be stopped on all nodes before upgrade.

The value of the following MySQL variables must be equal to 1:

- auto_increment_increment
- auto_increment_offset

Note: Before upgrade, if the cloud.properties file contains encrypted values of any property, they should be replaced by the clear-text (or decrypted) values. These properties can be replaced by encrypted values post upgrade.

Requirements

Requirement	Version
CloudPortal Business Manager (old)	1.4.3 to 1.4.7 (any)
CloudPortal Business Manager (new)	2.1.x
Python	Default with CentOS (Python 2.4 for CentOS 5.6 and Python 2.6 for CentOS 6.3)

Following are the commands to install the required packages if they do not exist & to update the packages with their respective latest version:

```
yum -y install MySQL-python  
yum -y install unzip
```

Installation and Upgrade

Following are the steps for CloudPortal Business Manager code upgrade.

Note: This must be done on every node of CloudPortal Business Manager.

- Extract the RPM in any location (CloudPortal-<untar_location>)
 - Example:tar zxvf CloudPortal-2.1.x-centos.tar.gz
- cd CloudPortal-<untar_location>
- Run installer using ./install.sh
- **Note:** Make sure you take a backup of CloudPortal Business Manager & DB (preferably complete CloudPortal Business Manager VM).
- Read & accept **Citrix End User License Agreement**.
- Select Option **U** in the Install options available in menu to upgrade.
- Follow the instructions that appear in the Upgrade wizard.
- Select option **Q** to quit.
- Following are the commands that must be run on each node in a clustered environment:

```
cd /root/CloudPortal-<version>/migration/2.0/from/1.4/scripts
# ./copy-connector-resources.sh
```

- Following are the commands for the pre-database upgrade of CloudPortal Business Manager, that must be run only once (on any one node in the clustered environment):

```
cd /root/CloudPortal-<untar_location>/migration/2.0/from/1.4/scripts
./pre-upgrade.sh
```

- Following are the commands for the database upgrade of CloudPortal Business Manager, that need to be run only once (on any one node in the clustered environment):

```
cd /root/CloudPortal-<version>/migration/2.0/from/1.4/scripts
# ./dbupgrader.sh
```

- If the proxy server is configured with port 20410. then make the changes given below in the tomcat-server.xml file:

```
<Connector port="20410" protocol="AJP/1.3"
redirectPort="8443"
URIEncoding="UTF-8"/>
```

- Once the above steps are successful, start the CloudPortal Business Manager server: using the command service cloud-portal start. Service start on any node should be done only after DB upgrade and copy connectors is done.

Note: Ensure that the browser cache is cleaned after upgrade.

Steps to Upgrade LDAP push from CloudPortal Business Manager 1.4.x to CloudPortal Business Manager 2.x

This section is applicable for those customers using LDAP in CloudPortal Business Manager 1.4.x.

1. Log in as Root.
2. Go to **Administration-> Integration-> Directory Server**.
3. Change `directory.service.enabled` to `true`.
4. Open `cloud.properties` which has current configurations of directory server.
5. Copy value of following from the file to the following in configuration in administration:
 - `Ldap.url: directory.server.url`
 - `Ldap.base: server.base.dn`
 - `Ldap.userDn: manager.dn`
 - `Ldap.password: manager.password`
6. Change following values in configuration in administration to the following:
 - `Handle.to.match.indirectory.server: (uid={0})`
 - `User.first.name.mapping: givenName`
 - `User.first.name.mapping: givenName`
 - `User.ldap.password.mapping: userPassword`
 - `User.class.type: account;extensibleObject`
 - `User.dn.identifie: uid`
 - `User.username.mapping: uid`
7. Click OK.
8. Ensure all the passwords for all the existing users are in place in directory server.
9. All the passwords in db for all the users are set to null.

Impact on customization

Magnitude of impact on customization depends on the degree of customization. Some of the very obvious points are reported below.

1. Any dependency on service layer methods of CloudPortal™ Business Manager core code base is not guaranteed to be backward compatible.
2. Many new UI screens are built in CloudPortal Business Manager 2.x and some of the existing ones are changed. This means any customization done for UI requires a careful review.
3. Some of the features, which were not present in CloudPortal Business Manager product earlier and required customization in past to achieve the same, may be supported in product itself now. Refer CloudPortal Business Manager 2.x features for the same.
4. Customization written based on assumption of some columns in database tables may undergo huge impact. Refer to new schema structure for the same.
5. Configuration file to set log levels (log4j.xml) is no longer in use now. New file to configure log levels is located at /usr/share/vts3/config/serviceability.xml. New logging mechanism supports both Log4j and Logback. However, transformation from old configuration file (log4j.xml) to new configuration file (serviceability.xml) is not supported. You have to reconfigure log levels in serviceability.xml.
6. Authorize.Net and CyberSource are the only supported payment gateways. If different payment gateway is used, one needs to write CloudPortal Business Manager connector for the same. For more information, please refer CloudPortal Business Manager Connector Developer guide.
7. All resource strings are persisted in database now. The mechanism to create custom resource strings is described in customization section in the Administration guide.
8. CloudPortal Business Manager now supports multi-level approvals for custom actions (workflows) such as account provisioning, subscription provisioning and account conversions. Any customization done prior should be rewritten using concept of customization bundle, where you can define and customize workflow and activities in structured manner in xml files. CloudPortal Business Manager will process these files and will instrument the customized workflows and activities. For more information refer, creating customized bundle section.
9. If any phase of usage collection and mediation is customized in CloudPortal Business Manager 1.4.x, same needs to be re-implemented using the product features of CloudPortal Business Manager 2.1.
10. Impact on CAS: If additional services are added in deployerConfigContext.xml, they need to be re-written after upgrade.
11. Impact on SSL Certificate: After upgrade to CPBM 2.4, re-import the SSL certificate using the command:

```
/usr/java/latest/bin/keytool -import -alias alias -keystore  
/usr/java/latest/lib/security/cacerts --file cert1.cer
```

Migration to Cloud Service, OSS and associated Service Instances

Concept of IaaS, Payment Gateway and Ticketing, Fraud control etc. is abstracted to Cloud Services and OSS in CloudPortal™ Business Manager 2.x. To setup the system to use this multi-service model, upgrade script automatically reads properties from cloud.properties and relevant tables. With help of this configuration setup in past, it creates service instance automatically. It automatically creates service instances for IaaS, Ticketing, and Payment Gateway, Credit card fraud control, Device fraud control, IP locator and Phone verification. If any of them is not enabled in the system before upgrade, script will not create any instance of the same.

Migration of properties in cloud.properties file

Following list details the properties that are migrated in cloud.properties file:

- Names of many properties in cloud.properties file are changed and some of them are migrated to database from properties file. Upgrade takes care of them automatically.
- Location of cloud.properties is also changed. Upgrade takes the default location of cloud.properties file in use which is /etc/cloud/portal/cloud.properties.
- Some of the properties are now specific to service instances, and will be moved to database after upgrade. Upgrade takes care of them.

Migration of Workflows

Workflow is a new feature in CloudPortal™ Business Manager 2.x. All the pending changes such as account creation and account conversion etc. will be migrated in form of "workflow" and "tasks". Any customization done for the same should be revisited in customization bundle to configure relevant xml files.

Migration of Billing

Billing has undergone metamorphosis in CloudPortal™ Business Manager 2.x. Both functional and technical implementations are changed to a large extent. Some of the points are highlighted below -

- Creation of each subscription will create an invoice. As soon as the subscription is created, a new invoice is generated. This invoice is called 'Subscription' invoice, which is accompanied by 'Renewal' invoice in case of monthly subscription.
- Even though invoice is created instantly on creation of subscription, payment of any invoice is not forced instantly unless it is configured to do so before upgrade. i.e., In CloudPortal Business Manager 1.4.x, Autopay & Instant invoicing should be on for the

account types to make sure, instant pay is on in CloudPortal Business Manager 2.x.

- Depending on the state & renewal date of subscriptions, necessary "Subscription" and "Renewal" invoices will be created.

Note: Make sure all the billing related jobs are successful for the previous run and system is in stable state on the day of migration.

Migration of Products and Product Bundles

Functionalities for Products and Product Bundles are augmented. In order to support migration for the same, default constraints are assumed. New products and product bundles created after upgrade can use the new features of CloudPortal™ Business Manager 2.x.

List of behavioral changes

Following list describes the list of features where behavior is changed in CloudPortal™ Business Manager 1.4.x. This list is addition to all the design changes stated in the earlier sections in this document. For more information on the feature list, refer to CloudPortal Business Manager 2.x documentation.

- If any bundle has more than one revision in current billing cycle & has an active subscription corresponding to it, then the subscription will be charged based on the most recent subscription charge of that bundle in the channel.
- Subscription has been decoupled from resource provisioning in CloudPortal Business Manager 2.x onwards. For example, destroying the resource does not terminate the subscription to which that resource was tied to. The subscription will continue to be charged, and user can re-provision a resource under that subscription.
- There is no restriction on launching a VM for a specific service offering and template group combination. On the contrary, in CloudPortal Business Manager 1.4.x, there were some constraints based on combination of service offering and template groups.
- Payment need not be done for a specific invoice. That means payments are now account wise and can be made even if there is no invoice due. Payments are decoupled from invoice. All payments are credited to account balance which is reflected in Account statement.
- In CloudPortal Business Manager 1.4.x, the entitlements are shared across all the users of the account whereas in CloudPortal Business Manager 2.x. onwards, the entitlements are user specific.
- Product Bundles cannot be removed from CloudPortal Business Manager 2.x. onwards.
- CloudPortal Business Manager 2.x will track the daily usage and daily spend per user, per day by default. It is not so in the older versions (unless configured). However, even if CloudPortal Business Manager 1.4.x is configured to use user daily usage data, the user usage data will not be migrated on daily basis. Adjustment invoice will be created for each user for the earlier usage while migrating.
- Logs of CloudPortal Business Manager from versions 2.x onwards are present in `/usr/share/cloud/portal/logs/cpbm-server.log`.
- Service Provider will no longer be able to set up Service Health status and Maintenance specific to a CCP zone. These will have to be set for specific service instance

Known issues

Following list describes some of the conceptual changes and observations along with workaround:

- In CloudPortal™ Business Manager (CPBM) 1.4.x, when an account is created, the domain is also created simultaneously in Cloud Stack. After upgrade, even if the account has any pending workflow such as email not verified or credit card details not entered, user still sees that the service instance to the cloud portal is enabled.
- For all inactive catalogs, prices will be null. You need the Catalog to synchronize with reference Catalog.
- Non Billing Admin Tenant users will no longer be able to see charges incurred to them for past Billing Periods.
- Non Billing Admin Tenant users will be able to see discounts and taxes on charges incurred by them for the current period.
- If there are any terminated accounts in the system, then billing job after migration might give partial error for terminated account.
- For products, types do not exist in CPBM 2.x. Instead, categories are introduced. Right now all the categories are shown as uncategorized after upgrade.
- If autopay is disabled & instant invoicing is enabled, then the autopay flag gets enabled. However, this is an unsupported configuration.
- After upgrade, the details page, of existing invoices, displays empty fields for description, quantity, and chargeable units.
- On upgraded setup (from 1.4.X), BSS APIs will fail with "authentication error". As a workaround, run the following SQL query on cloud_portal database:

```
update users set secret_key = HEX(AES_ENCRYPT(secret_key ,'Key_to_Decrypt'));
```


Replace "Key_to_Decrypt" with the "field.encrypted.key" property from your cloud.properties file.
- If CCP's non-Admin-Server list contains more than one IP, the upgrade from CPBM 2.1 might fail.
Workaround: Ensure that CCP's non-Admin-Server list contains only one IP.
- If the /usr/share/vts3/patches directory is not cleaned up, the upgrade to CPBM 2.2 might fail.
- After upgrade from CPBM 1.4.x to CPBM 2.x, network offering is displayed twice in the Provisioning Constraints page of the Add Product Bundle wizard.

- After upgrade from CPBM 1.4.x to CPBM 2.x, the "hours.in.usage.units.of.month" billing parameter gets reset to 744.

Workaround: In case you have set the value for the configuration "hours.in.usage.units.of.month" in CPBM 1.4.x, then after upgrading to 2.x, you have to reset it back to its original set value from **Administration > Configuration > Portal > Billing > hoursIn.month**.

Warning: CPBM upgrade is not supported on the physical server and is only supported on the VMs.

Connector Development Kit

Cloud Service Connector implementers should use this specification to realize integration between the Cloud Service and CPBM.

- [Terminology](#)
- [Target Audience](#)
- [Overview](#)
- [Service Installation and Bootstrap](#)
- [Product Management](#)
- [Development Using SDK](#)

Terminology

Connector Implementer: Any third party that is using this SDK to implement a connector for integration between a cloud service and CPBM

CPBM: CloudPortal Business Manager

Cloud Service: A capability delivered as a cloud-enabled service

Service Provider/Operator: The entity managing and operating a cloud, which is installing CPBM along with other Cloud services for self-service delivery.

SPI: Service Provider Interface. A technically correct name for a Java interface that the connector will have to *implement* and will be *called by CPBM* at appropriate points as specified in this specification. These implementations will have to conform to the semantics specified in this specification.

Target Audience

Cloud Service Connector implementers should use this specification to realize integration between the Cloud Service and CPBM. The intended audience is developers implementing the integration. Developers attempting to integrate a Cloud Service using this SDK are required to be proficient in:

- The Java programming language and libraries
- Spring Framework
- Basic understanding of OSGi and the Virgo platform
- The API and capabilities of the target cloud service

Overview

CloudPortal Business Manager (CPBM) is a multi-lingual; multi-currency self-service portal product that enables cloud-enabled and delivered services to be vended to end consumers. It provides a comprehensive business and operational support system for operators to use to deliver these cloud services to their audience.

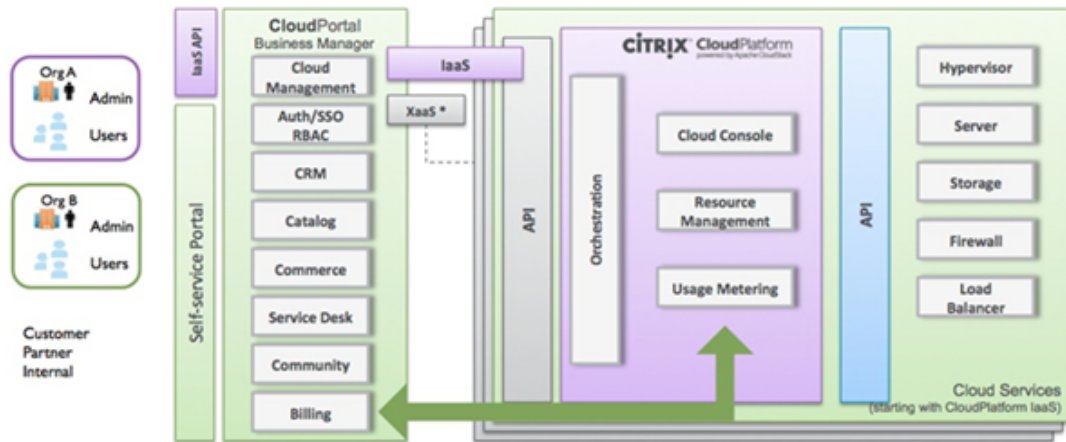


Figure 1. CloudPortal Business Manager

The above figure shows CPBM with the connector for an IaaS cloud service built for the Citrix CloudPlatform product. Other Cloud services may integrate with CPBM by developing a custom built connector for their own service that conforms to the specifications described in this document. To facilitate the development of this connector, CloudPortal Business Manager provides a Software Development Kit (SDK) that can be used to ease the development of this connector.

To implement a connector, an implementer will have to consider multiple aspects of the integration. The following figure describes the steps needed to integrate a Cloud Service with CPBM.



Figure 2. Steps for Cloud Service Integration

The specification includes descriptions of the metadata that needs to be provided, the SPI interfaces that the Cloud Service connector will have to implement as well as the CPBM facilities available for the connector to use in order to realize the integration. We will discuss each step in the following sections.

At a very high level, a Cloud Service Connector will have to provide the following:

- **Metadata:** This is vital information describing the capabilities of the underlying Cloud Service along with information needed to successfully connect and orchestrate actions against an instance of a Cloud Service. This will typically be represented as an XML file that is bundled as part of the package.
- **SPI Implementation:** A set of implementations to the various SPIs defined in CPBM. Some of these SPIs are mandatory while others will be able to leverage CPBM capabilities to deliver additional functionality, if implemented.
- **Resource Files:** A set of resource files that provide language specific translations for any UI visible elements of this Cloud Service.

The connector will be packaged as one or more OSGi bundles for deployment into the CPBM footprint.

Multiple Instances

It is important to note that while a connector implements the mechanisms needed to communicate with a cloud service, the expectation should be that CPBM might initialize multiple instances of the connector to communicate with multiple instances of the cloud service. Typically, this would mean that the information provided in the connector is merely metadata that will be replicated for each instantiation of this connector. Services that are delivered as SaaS applications, as opposed to installed applications, will of course be, by definition, limited to one instance.

Onboarding

The CPBM platform is a Java based application infrastructure that heavily uses the Spring Framework and is built on the Virgo OSGi container to enable modular deployment of its capabilities. A Cloud Service connector is required to conform to this model to enable seamless integration. A cloud service connector is packaged as a set of Virgo OSGi bundles. These bundles will then be deployed as part of a CPBM installation to deliver this cloud service as a capability that can be vended and billed for using CPBM.

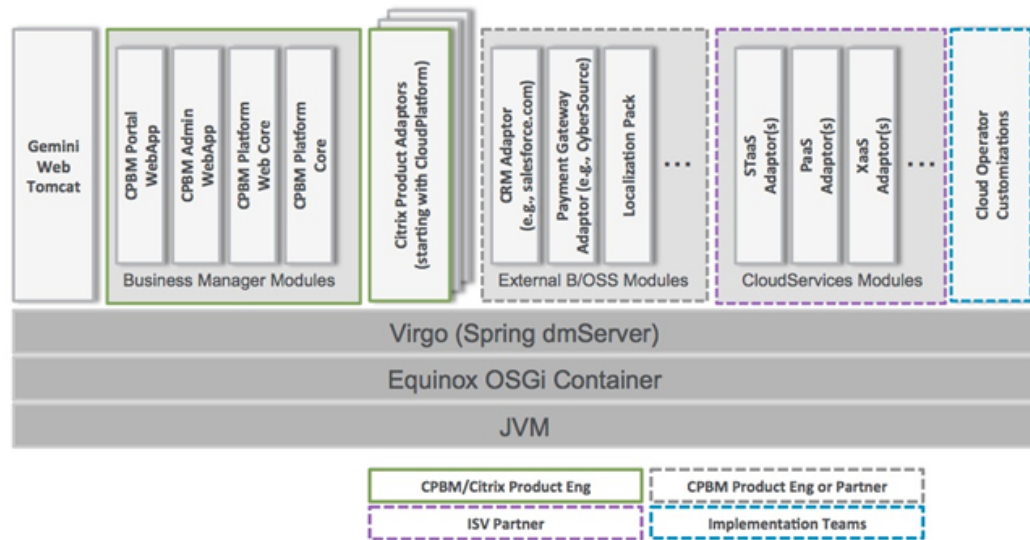


Figure 1. CloudPortal Business Manager Runtime Architecture

The above figure depicts the runtime architecture of the CPBM platform with multiple cloud service connectors installed. The following figure describes what a connector contains.

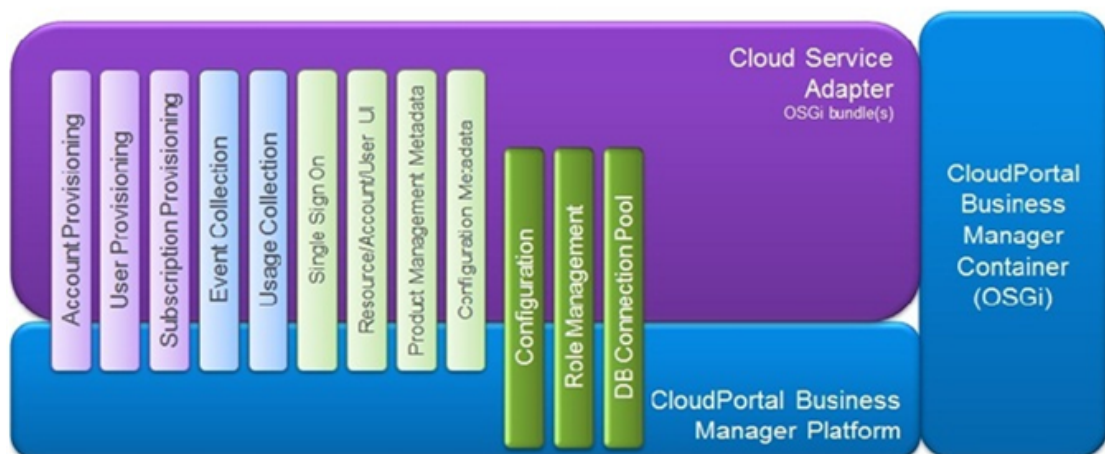


Figure 2. Cloud Service Connector

Metadata

A cloud service connector implementer provides the configuration details of the cloud service to CPBM in order for the cloud service to be correctly integrated with CPBM. The metadata required is comprised of many parts. These include:

- Metadata about the configuration properties needed for the connector to effectively communicate with the target cloud service. This will minimally include properties that will identify the end point for the specified Cloud Service.
- Metadata about the resource types, usage types (usage metrics collected by the target cloud service) and filters (used to scope resource and usage types) of the target cloud service.
- Metadata about the resource UI contribution requirements for the target cloud service. This is essentially a list of UI elements that can be embedded in various sections of the CloudPortal UI.
- Metadata about custom security roles that control access to the target cloud service.

SPI

A Cloud Service Connector implementer needs to implement certain SPIs to hook into CPBM to participate in the customer onboarding and service provisioning aspects, in addition to providing information to CPBM to generate appropriate usage metrics and bills to the end customer. These are:

- **Bootstrap:** Set of interfaces required by CPBM to bootstrap and interact with the service. These interfaces allow CPBM to configure cloud service instances and get access to the implementations of the SPIs described below.
- **Account Lifecycle Handler:** Called by CPBM at appropriate times, this SPI allows the connector to synchronize its state with that of CPBM. For example, CPBM will call this adapter to synchronize state when a CPBM account (tenant) is created, restricted, suspended, activated, terminated, etc.
- **User Lifecycle Handler:** Called by CPBM at appropriate times during a user lifecycle, this SPI allows the connector to synchronize its user state with that of CPBM. For example, when user is locked, disabled etc.
- **Subscription Provisioning Handler (Mandatory):** Called by CPBM to provision and de-provision resources that are subscribed to (or cancelled).
- **Usage Collection (Mandatory):** Called by CPBM periodically to collect usage related information for billing.
- **Event Collection:** Called by CPBM at periodic intervals to collect salient events from the underlying service. While optional, when implemented this will allow end customers to subscribe to notifications when certain things occur either via email or SMS etc.

Note: This capability will be implemented in CPBM in a future release.

- **ViewResolver (Mandatory):** Allows CPBM to drop the end user into the cloud service UI with the appropriate context, for example, account settings, user settings, resource management etc.
- **SSO Handler (Mandatory):** Allows CPBM to initiate and maintain single-sign on so that users don't have to login again when being redirected to the cloud service UI.
- **APIProxyHandler:** Used if the service has an API and the API needs to be fronted by CPBM. This allows CPBM to allow business rules and other technical constraints (such as blacklisting APIs that should not be called when integrated with CPBM).

Resource Bundles

In order to support localization, a cloud service implementer should provide resource bundles in supported languages to allow the display of the service entities in appropriate languages to the service operator as well as the end customer.

Example: `CloudServiceApplicationResources.properties`

`#About the Cloud Service`

`<UniqueServiceName>.service.name=#Name of the Serice`

`<UniqueServiceName>.service.description=#Description about the service`

`#Property section`

`<UniqueServiceName>.property.id.name=#Name of the Property`

`<UniqueServiceName>.property.id.tooltip =#Tool tip for the property`

`<UniqueServiceName>.property.id.errortext=#In case of validation error, the error message to be used`

Service Installation and Bootstrap

A connector package includes metadata and OSGi bundles that include SPI implementations for the above-mentioned SPIs, in addition to any resource bundles.

The OSGi bundle includes the following components:

- `/serviceDefinition.xml`: - contains the service metadata information, which includes BootStrap, Products, Views sections.
- `/images/`: contains the images, icons, banner files.
- `/OSGI-INF/l10n/resources/CloudServiceApplicationResource[_lang].properties`: contains the resource properties per language supported.
- `/TermsAndConditions.txt`: contains the terms and conditions text for your connector. Currently locale specific terms and conditions cannot be specified.

Installing a Cloud Service Connector

As a service operator, perform the following steps to install a cloud service connector:

1. Drop the connector OSGi service bundle in the Virgo's /connectors_ext directory.
2. Add the line `<artifact type="bundle" name="<bundle.artifact.id>" version="["<x.x.x>,<x.x.x>"]"/>` within the plan tag in the file `com.citrix.cpbm.connector.ext.plan` located at `/usr/share/vts3/connectors_ext`.
3. You can now instantiate the service in the CloudPortal Business Manager Configuration UI.

Note: For installing multiple connectors, you must repeat the steps mentioned above.

Service Bootstrap

In order to install and bootstrap a service into CPBM, CPBM requires some metadata and SPI implementations. This information is used by CPBM to display the available services in the configuration UI and allow the instantiation of the connector implementation to interface with one or more instances of each service.

Service Metadata

The following information is required by CPBM to enable instantiation and configuration of service instances. This information is provided as XML sections in the serviceDefinition.xml file.

Element (items in italics are attributes)	Description	
Id		A unique identifier for the service. Similar to java package conventions. For example, com.citrix.ccp
	<i>type</i>	(CLOUD OSS*) The type of service (For cloud services, this should always be set to CLOUD)
	<i>Singleton</i>	Boolean(default: false) Indicates if this cloud service is a singleton or not.
	<i>category</i>	(IaaS STaaS PaaS ...) Predefined values that determine the service category
CloudConnectorFactory		The service id of the connector factory implementation. This is the primary SPI for the bootstrap process.
Image		An image reference URL for use in the UI. Depending on the type it will be used under different circumstances
	<i>type</i>	Banner: Displayed when showing the service in the configuration UI Logo: Displayed when listing service instances
Locales		List of languages supported by the cloud service. For example: <Locales>en_US, ja_JP, ko_KR</Locales>
Vendor		The name of the connector implementer
	<i>version</i>	The connector version

Configuration Metadata

A cloud service connector must specify the set of properties needed for the connector to successfully communicate with an instance of the cloud service. These set of properties will be configured in the administration UI for each cloud service instance that is instantiated. For a singleton instance, there will be only one instance of the cloud service that can be configured. NOTE: It is important to note the point that when instances are configured there will be one set of properties that will be instantiated per instance of the cloud service that is configured. The configuration properties are specified as a collection of `Property` elements enclosed in a `Properties` element. The definition of each `Property` element is described below:

Element (items in italics are attributes)	Description
Property	A property that needs to be configured
<i>Id</i>	Dot separated identifier of property (for example, endpoint.url)
<i>Type</i>	(Integer/ String /url/Boolean/Number/Date/Password)
<i>Encrypted</i>	(true/false)
Validation	JSON validation string. Syntax same as for jquery.validation plugin, formal specification. Validations described here will be applied both on the Client and Server side. List of supported validations: required, min-value, max-value, regex, maxlength, minlength alphanumeric email

The property element under configuration now has new attributes, namely default and unique. The default attribute contains the default value for a given property, and if a property has unique = true, then no two service instances for the given service can have the same value for the given property.

CloudConnectorFactory Interface

The CloudConnectorFactory interface is the primary class that's used to bootstrap a cloud service.

The CloudConnectorFactory implementation for a service acts as a factory for CloudConnector objects. Each CloudConnector object represents a connection to one instance of the CloudConnector. The CloudConnectorFactory interface is defined as follows:

```
package com.citrix.cpbm.platform.spi;

public interface CloudConnectorFactory<T extends BaseConnector> {
    /* Initialize a CloudConnector instance with instance specific data */
    public T initialize(ServiceInstanceConfiguration config);
    public enum ConnectorType {
        CLOUD,
        ...
    }
    public ConnectorType getConnectorType();
}
```

CPBM will call the initialize method on the CloudConnectorFactory implementation providing a ServiceInstanceConfiguration object. This will be done for each instance configured in CPBM for this cloud service. The ServiceInstanceConfiguration object will contain values for each property described in the previous section. Note that the values will be specific to this instance of the cloud service.

Cloud Service CloudConnectorFactory objects should always return CLOUD for their ConnectorType.

ServiceInstanceConfiguration

The ServiceInstanceConfiguration object is the holder of all instance specific configuration values. The connector is initialized with a ServiceInstanceConfiguration instance that contains all the properties configured for this instance. The list of properties available is dependent on the property types declared in the service metadata.

ServiceInstanceConfiguration has methods that allow the connector to gain access the property values to connect to a specific instance of the cloud service. It is defined as follows:

```
public class ServiceInstanceConfiguration {  
  
    /* Returns the Service UUID */  
    public String getServiceUUID();  
  
    /* Returns the Service Instance UUID */  
    public String getInstanceUUID();  
  
    /* Returns the map of property name id and Instance Property */  
    public Map<String, ServiceInstanceProperty> getInstanceProperties();  
}  
  
public class ServiceInstanceProperty {  
  
    /* Returns the type of property */  
    public String getType();  
  
    /* Returns whether the property is encrypted or not */  
    public Boolean isEncrypted();  
  
    /* Returns the value of property */  
    public Object getValue();  
  
    /* Returns the validation string for the property */  
    public String getValidation();  
}
```

When the operator configures a service instance, CPBM will request the operator to configure the set of properties required for that service instance. These properties can be managed in the CPBM configuration UI. Once initialized, any changes to these configurations are NOT reflected in the running instance. Therefore, CPBM will have to be restarted when a service instance is reconfigured. Further, once configured to point to a particular service instance, changes to configuration cannot modify the target instance. This is due to the fact that CPBM handles and references to resources in that service instance and will fail if those handles and references are no longer valid.

CloudConnector Interface

The CloudConnector interface represents a connection to a specific instance of a cloud service. Implementations of this interface will provide CPBM handles to the various SPI implementations that interface with CPBM as described in later sections. Since these implementations are instance specific, ensure that they use the instance specific configuration values to connect to the appropriate instance of the cloud service. In addition, make sure that they are both thread-safe and have no shared state (across instances). How instance specific information is shared between the various SPI implementations is out of scope of this document. There are various ways this can be achieved and is left to the implementer to decide how they wish to architect this. The CloudConnector interface extends the Connector interface. For the purposes of this conversation, Connector itself is not pertinent except to say that the methods in it are applicable to CloudConnector. The CloudConnector interface is defined as follows:

```
/**
 * BaseConnector is the base interface for all connectors. Cloud Service Connectors
 * should always implement CloudConnector interface
 */
public interface BaseConnector {
    /**
     * Sets the ServiceInstanceConfiguration corresponding to the service instance this
     * is configured for
     */
    public void setServiceInstanceConfiguration(ServiceInstanceConfiguration config);
    /**
     * Gets the instance configuration
     *
     * @return the current service instance's configuration
     */
    public ServiceInstanceConfiguration getServiceInstanceConfiguration();
    /**
     * Returns the service UUID for the configured service instance
     */
    public String getServiceUUID();
    /**
     * Returns the service instance UUID for the configured service instance
     */
    public String getServiceInstanceUUID();
    /**
     * Returns the service status (and verifies connectivity) of this service instance
     */
    public boolean getStatus();
}

/**
 * The Connector expects basic account and user lifecycle
 * capabilities. Cloud Services are expected to implement the
 * extended CloudConnector interface instead.
 */
public interface Connector extends BaseConnector {
```

```
    /* Gets the account life cycle handler. */
    public AccountLifecycleHandler getAccountLifecycleHandler();

    /* Gets the user life cycle handler. */
    public UserLifecycleHandler getUserLifecycleHandler();
}

public interface CloudConnector extends Connector {

    /* Gets the metadata registry */
    public MetadataRegistry getMetadataRegistry();

    /* Gets the subscription lifecycle handler */
    public SubscriptionLifecycleHandler getSubscriptionLifecycleHandler();

    /* Gets the event collection service */
    public EventCollector getEventCollector();

    /* Gets the usage collection service */
    public UsageCollector getUsageCollector();

    /* Returns the view resolved configured for this service instance*/
    public ViewResolver getViewResolver();

    /* Get API handler for the api group */
    public APIHandler getAPIHandler(String apiGroupName);

    /*Get a map of API handles
    */
    public Map<String, String> getAPIHandles();

    /*Returns the Sso handler */
    public SsoHandler getSSOHandler();
}
```

All methods in the CloudConnector (and the Connector) interfaces are essentially factory methods that return service instance specific instantiations of their corresponding SPI interfaces.

Security Roles

In order to allow Cloud services to model their user privilege models in CPBM, a cloud service is able to introduce security roles that can be used by the connector to determine the level of privilege to be accorded to users being represented in the cloud service. These roles can then be added to appropriate profiles defined in CPBM that can then be granted to individual users in CPBM. In CPBM, roles are scoped by the level of visibility they have in the system. There are four defined scopes in CPBM. They are:

- **GLOBAL_ADMIN:** Represents the super user scope. There are two and only two users in the system that have profiles of this scope. 'root' and 'portal'. 'portal' represents the CloudPortal Business Manager portal itself, and any operations done by the system are done as the 'portal' user.
- **GLOBAL:** Has global visibility across multiple tenants. These roles require that the user be a service operator user (i.e, is a member of the SERVICE tenant), which will be tested by the security system before granting users access.
- **TENANT_ADMIN:** Tenant scoped role that represents a tenant administrator. This user, as a rule, should have visibility across users in this tenant.
- **TENANT:** Tenant scoped roles are granted to users who have visibility across all users in a tenant. Roles in this scope are used to manage resources within a given tenant.
- **USER:** User scoped roles are roles that are granted to users who should have visibility to only what they own and manage.

When users are created, they are associated with a profile. This profile has a list of roles associated with the profile. Connectors should use these roles to determine the level of privilege the user should be provided when the user is created in their system.

Security roles specific to a connector/cloud service are described in the metadata.

Semantics of adding/removing roles from profiles and how that is reflected in the underlying service.

Product Management

CPBM natively bills users for usage accrued by an account. Ideally this usage accrual is scoped by user (if the service supports that notion) or at the account level. In order to support this, a connector has to provide sufficient information to CPBM for it to be able to do the following actions:

- Appropriate selection of resource type components which make up the information that needs to be specified for provisioning a resource
- Appropriate input of optional provisioning time configuration information for each resource that's provisioned.
- Ability to identify and display prices for resources that are being provisioned
- Ability to filter resource components based on domain specific filters (in the case of resource component options being large or pricing being different)
- Ability to segregate usage information into appropriate pricing buckets
- Ability to describe technical constraints with respect to provisioning (such as once per account only or require X to be provisioned before Y can be provisioned etc)
- Ability to describe 'add-on' relationships where a resource is provisioned as an 'add-on' to an existing resource.

Before delving into the details of the metadata and SPI implementations that needs to be provided by a connector to achieve the above, it is useful to understand a little bit about how CPBM provisions and bills for resources consumed by a customer. We will discuss the CPBM product management model in the following section.

CPBM Product Management Primer

CPBM models resources in terms of Products and Product Bundles. While the CPBM billing engine contains other entities such as Catalogs, Channels, Campaigns etc, these have no bearing on what the cloud service connector and therefore be left out of this conversation. Please refer to the Service Operator's Guide for details on those entities.

Products

In CPBM, anything that can be measured and has a unique unit price is modeled as a Product. A Product is associated with a set of mediation rules that allows CPBM to segregate collected usage into appropriate product buckets. Service operators will create products guided by metadata provided by the underlying cloud service connector but based on their unique requirements for how they would like to charge for the service.

Note that Products represent anything that can be charged for. It is NOT REQUIRED that these be resources that are explicitly provisioned by CPBM. For example, in an IaaS service, one may provision one or more virtual machines, but may also get charged for data transfer that is not explicitly provisioned but gets accrued nevertheless. Products are defined specific to a given service instance (i.e., different instances of the service may have different products).

Mediation Rules

As discussed above, a given product can have one or more mediation rules.

All usage that matches *any* of the mediation rules will be attributed to that product. The mediation rule determines how usage record matches are done by the mediation strategy. A mediation rule contains among other things the usage type (a unique cloud service specific identifier that distinguishes usage of different types) and a list of discriminator attributes and corresponding values that need to match for usage to be attributed to this product. When CPBM collects usage from the cloud service, the collector will also provide values for all the discriminator attributes. CPBM will then match those values with those specified in the mediation rules to determine a match.

Mediation rules also allow CPBM to determine which products are likely to be used when provisioning resources. Based on the selections during provisioning, one or more products may be applicable. Using the available selections, CPBM will identify the set of products that are applicable and will be able to display appropriate pricing for the same.

Product Bundles

Product Bundles are essentially what is sold to an end customer. End customers always buy product bundles. A product bundle is composed of five parts.

Type

The bundle type essentially represents the resource that is being provisioned when purchasing this bundle. Type can be null indicating that this bundle does not provision anything but provides an entitlement for a price.

Provisioning Constraints

Provisioning constraints describe any constraints that restrict what can be provisioned. Typically, this would be used to constrain resources to a subset of possible choices for that resource. See Resource Components metadata for more details. Provisioning constraints are described in terms of resource component constraint rules.

Business rules

These are attributes that essentially reflect the operator's business rules. They include:

- **published:** Whether this bundle is published, ie available for purchase or not
- **constraint:** Any kind of purchasing constraint for this bundle. Typically inherited from the resource being provisioned, this constraint can also be manually applied when no resource is being provisioned. For example, whether this bundle can be purchased once per account, once per user, many times but at the account scope or many times at the user scope.

Pricing

These are attributes that reflect the operator's pricing for a given bundle. They include:

- **recurrence:** The frequency at which the prices will be charged (Monthly, Annually or none which implies utility (or pay for what you use) pricing)
- **one-time charges:** Charges applied at subscription time
- **recurring charges:** Charges applied at the recurrence frequency

Entitlements

Entitlements reflect what is 'bundled' with the purchase of a subscription to this bundle. Entitlements are essentially usage that comes free with the purchase of this bundle. Entitlements are described in terms of amount of units of a given product that is bundled. Entitlement can be unlimited in which case all usage of that product is free for this subscription. Some entitlements are scoped by subscription whereas user or account scopes other entitlements.

Product Metadata and Metadata Registry

In order to support the above functionality in CPBM, the connector implementer has to provide some product metadata as well as implement some SPIs.

There are three main elements that describe product metadata for CPBM to use. These are:

- **Resource Type:** Resource type represents resources that require an explicit provisioning action and optionally additional configuration for provisioning.
- **Usage Type:** Usage type represents distinct measurements made related to usage by a cloud service. Examples include number of mailboxes, number of hours of VM running time, amount of storage consumed, amount of data transfer, and so on.
- **Filter:** Filters are used to subset resource components or segregate usage based on domain specific distinctions. Examples include availability-zone, delivery location, operating system, and so on.

ccpb-cssdk-resource-type-20-con

Due to technical difficulties, we are unable to display this topic. Citrix is currently fixing this problem. In the meantime, you can view this topic online:

<http://support.citrix.com/proddocs/index.jsp?lang=en&topic=/cpbm-21-map/ccpb-cssdk-resource-type-20-con.html>

UsageType

UsageType represents distinct measured usage metrics generated by the cloud service. UsageTypes may be generated by provisioned resources such as hours of VM running time generated by the VM or implicitly, such as data transfer, which does not get explicitly provisioned but usage is incurred nevertheless. When generated by provisioned resources, usage can be attributed to that resource. Otherwise, usage is typically accrued at the ownership scope, which could be user or account.

To understand cloned usage types, consider monitoring which is would be modeled as an add-on resource type to a VM. The actual monitoring usage is derived from running VM hours. However, such usage is not generated unless monitoring is provisioned on a target VM. So, CPBM will automatically generate a cloned usage type identical to the target usage type but only if the resource type that generates it has been provisioned as an add on. As with resource types, cloned usage types may refer to usage types from another service using the syntax <service_id/name>.

Table 1. UsageType

Element (Items in italics are attributes)		Description
UsageType	<i>name</i>	The name of the usage type
	<i>[clone]</i>	Indicates the usage type is a clone of another usage type. If generated by a resource type, this kind of usage is generated only if the associated resource type has been provisioned
	<i>uom</i>	Unit of measure reported by usage collector
	<i>Discriminators</i>	List of discriminator attributes that can be used to mediate this resource type. Possible values can be acquired from <code>getDiscriminatorValues</code> method of the <code>MetadataRegistry</code> interface
	<i>Attribute</i>	Name of a discriminator attribute

It is useful to note that there are a few implicit discriminators for usage types. All filters and all resource component selections (based on their corresponding discriminator attributes) are valid discriminators for usage types. Cloned usage types inherit all discriminators of their target.

Unit of Measure (Uom)

Unit of Measure is the unit of a UsageType, which is specified along the UsageType definition. Each Uom can have multiple scales, and each scale has a conversion factor. Each Uom has a implicit scale which is the Uom itself, which has a conversion factor of 1, and it may have other scales which have other conversion factors. Each Uom has a default scale which is specified using the default attribute, and if no scale has a default attribute, then the implicit scale is considered as default. Each Scale has a monthly attribute, which if set as true would imply the conversion factor specified in the scale is a monthly unit and will be multiplied with number of hours in a month, which is specified in CPBM as a global configuration.

Element (Items in italics are attributes)		Description	
<u>Uom</u>	<i>raw</i>		Name of the Uom, Same as the <u>uom</u> attribute specified in an <u>UsageType</u> .
	<i>Scale</i>	<i>name</i>	Name of the scale.
		<i>conversionFactor</i>	The conversion factor, which is used to convert the usage into different scales.
		<i>monthly</i>	Attribute which specifies whether scale is monthly and whether conversion factor needs to be multiplied with number of hours in a month(Specified by CPBM).

Filter

A Filter can be considered a global discriminator. It can be used to classify all resource components as well all usage. Use filters only when it is applicable to all resources and usage types declared for a cloud service.

A filter can be used to filter resource components. This allows scoping of resource component options based on filter values. For example, in Citrix CloudPlatform, a Zone would be a filter allowing templates or service offerings to be filtered by Zone.

In addition, a filter can also be used to discriminate usage, and as such is considered an implicit discriminator for any usage type in a given service.

Possible filter values can be acquired dynamically from `getFilterValues` method of the `MetadataRegistry` interface.

Table 1. Filter

Element(Items in italics are attributes)		Description
Filter	<i>name</i>	(Attribute) the name of the filter
	<i>discriminator</i>	discriminator property

MetadataRegistry Interface

The MetadataRegistry interface is the runtime product metadata view into the target cloud service. Actual list of available options for resource components, filters, discriminators are dependent on multiple factors, such as who the calling user is and what options are visible to them or not. Due to this constraint, CPBM will use the metadata registry to query the target cloud service at runtime to determine exact set of options available for each case. This allows the service operator to make changes to available options without having to reflect that explicitly in CPBM.

ResourceComponent

A representation of ResourceComponent is used by the MetadataRegistry and is defined as follows:

```
public class ResourceComponent {  
  
    public ResourceComponent(String resourceType, String name, String value, ResourceComponent parent,  
        Map<String, String> attributes, Set<DisplayAttribute> displayAttributes);  
  
    public String getResourceType();  
    public String getName();  
    public String getValue();  
    public Map<String, String> getAttributes();  
    public ResourceComponent getParent();  
    public Set<DisplayAttribute> getDisplayAttributes();  
    public String getAttributesString();  
    public String getDisplayAttributesString();  
  
}
```

A ResourceComponent object is a representation of a resource component. It contains the following properties:

- **resourceType:** The name of the resource type that this component is a part of
- **name:** The service defined name of this component option
- **value:** The service defined value that needs to be passed back to the service to provision a resource with this component
- **attributes:** additional descriptors for that resource component, if any.
- **parent:** can usually be null, but under some advanced use cases, resource components may be derived from another resource component. This field defines the parent

A new field has been added in ResourceComponent, called displayAttributes. This is a set of name-value pairs (DisplayAttribute), used to show additional details for a given ResourceComponent in the following places:

- While adding or editing provisioning constraints for a Product Bundle.

- While end user provisions a resource using the default UI. (can also be used by connector in custom UI).

The `getDisplayAttributesString()` is a helper method which returns a comma separated string representation of `displayAttributes` field.

FilterComponent

```
public class FilterComponent{

    public FilterComponent(String name, String value,
        Map<String, String> attributes, Set<DisplayAttribute> displayAttributes);

    public String getName();
    public String getValue();
    public Map<String, String> getAttributes();
    public Set<DisplayAttribute> getDisplayAttributes();
    public String getAttributesString();
    public String getDisplayAttributesString();

}
```

The `getFilterValues()` method has been updated, it now returns a list of `filterComponent`.

MetadataRegistry

The `MetadataRegistry` as we described earlier is used by CPBM to query the underlying service at runtime to determine dynamically the value space for resource component options, filters, discriminators etc. The metadata registry itself will be used in three different contexts and it is important to understand these. They are:

- **At product management time:** In order to set pricing, it is important for CPBM to understand the various resource type component options, filter options, possible discriminator attribute values. These help CPBM build the products, define their mediation rules and define product bundles and any constraints they may want to specify.
- **At provisioning time:** When provisioning a resource type, the user will have to select specific resource component values (optionally filtered by the available filters) for each resource component. The actual list of options can vary depending on the user, account, filter selections or other resource component selections. Resource types may not have components or variations. Selection of product bundles that can be subscribed to will match these selections with bundle constraints to show the set of bundles that match the selections.
- **At mediation time:** At mediation time, usage can be received for a specific resource type with specific resource component values. At that time, CPBM will need to determine ancestry of resource components to correctly bucket usage into appropriate products. CPBM will query the `MetadataRegistry` to determine this ancestry.

```
public interface MetadataRegistry {

    public Map<String, String> getResourceTypeValues(String tenantHandle, String userHandle, String name,
```

```
    Map<String, String> context, Map<String, String> filters);

    public List<FilterComponent> getFilterValues(String tenantHandle, String userHandle, String name);

    public Map<String, String> getDiscriminatorValues(String name);

    public List<ResourceComponent> getResourceComponentValues(String resourceType, String name);

    public List<ResourceComponent> getResourceComponentValues(String resourceType, String name, String
        String userHandle, Map<String, String> context, Map<String, String> filters);

    public ResourceComponent getResourceComponent(String resourceType, String resourceComponent, Stri

}

```

getResourceComponentValues

This method is overloaded for two different contexts. One is used in product management while the other is used at provisioning time.

The first form is called when the operator is defining products to determine the list of system defined resource components. In this form, ancestry is not important (and will be ignored), as ancestry is typically relevant only during provisioning (for price display) and mediation. Also, in this form, the connector may return sentinel values (such as -1) to designate classes of resource components that are available only to a specific custom scope (since the actual resource component options are known only when the scope is defined. For example, you may want to use -1 to designate all user defined templates. The actual list of user defined templates, of course, can only be known given the user). This form is used to build product mediation rules that use resource component values as discriminators.

The second form is used during provisioning. At this time, the account, user, any other components that have already been selected and any filters applied are known to the system and will be passed to the call. Based on these parameters, the implementation should return a list of specific resource components that are available to be used for provisioning this resource type. In the case of IaaS, for example, this may include user defined templates, templates owned by users, but derived from system templates, zone or account specific templates etc.

getFilterValues

This method is called during product management to use filters as usage discriminators as well as during provisioning to filter out resource components. It should basically return a list of legal values for that filter that can be used to filter a resource component or categorize usage with. For example, you may declare availability zone as a filter and this method would return the current list of availability zones.

getDiscriminatorValues

`getDiscriminatorValues` works similar to `getFilterValues` and returns possible values for a given discriminator attribute. This is typically called only during product management. It may return null, which would indicate that the actual discriminator value space is not static or cannot be determined. In this case, CPBM will assume the value space is undefined and allow the operator to enter free form text. The actual value of course, will have to match the values that are configured by the operator for that usage type. For example, this may be used to discriminate volume usage based on storage pool tags

(the list of tags potentially being infinite and not formally defined).

Understanding How This Works

To understand how all this works, let's take a look at how they all fit together. The following figure shows the relationship between these entities and what each of the Consumer, the Service Operator and the Connector deal with.

To summarize at a high level, the Consumer can provision resources of a given resource type either directly on a pay-as-you go basis or by purchasing a subscription to a Product Bundle.

Service Operators define Products and Product Bundles. Products and Product bundles have prices. You can pay as a utility, Pay-as-you-go, or recurring respectively. Bundles may also provide free usage in terms of entitlements along with the purchase of a subscription. Bundles in turn will also provision resources but subject to the pricing of the bundle. Resources provisioned without a subscription will be charged on a utility basis. Products have mediation rules that allow CPBM to mediate usage reported by the cloud service (via the connector).

Connectors provide meta information, in terms of the Resource Types available, their components, UsageTypes with their discriminators and Filters which allow Resource Types to be classified and Usage to be mediated. Some usage is generated only when resources of the corresponding resource type are provisioned. This usage is directly attributable to that resource. Other usage can be generated based on other activity in the cloud service which will be accrued at the user (or account) level.

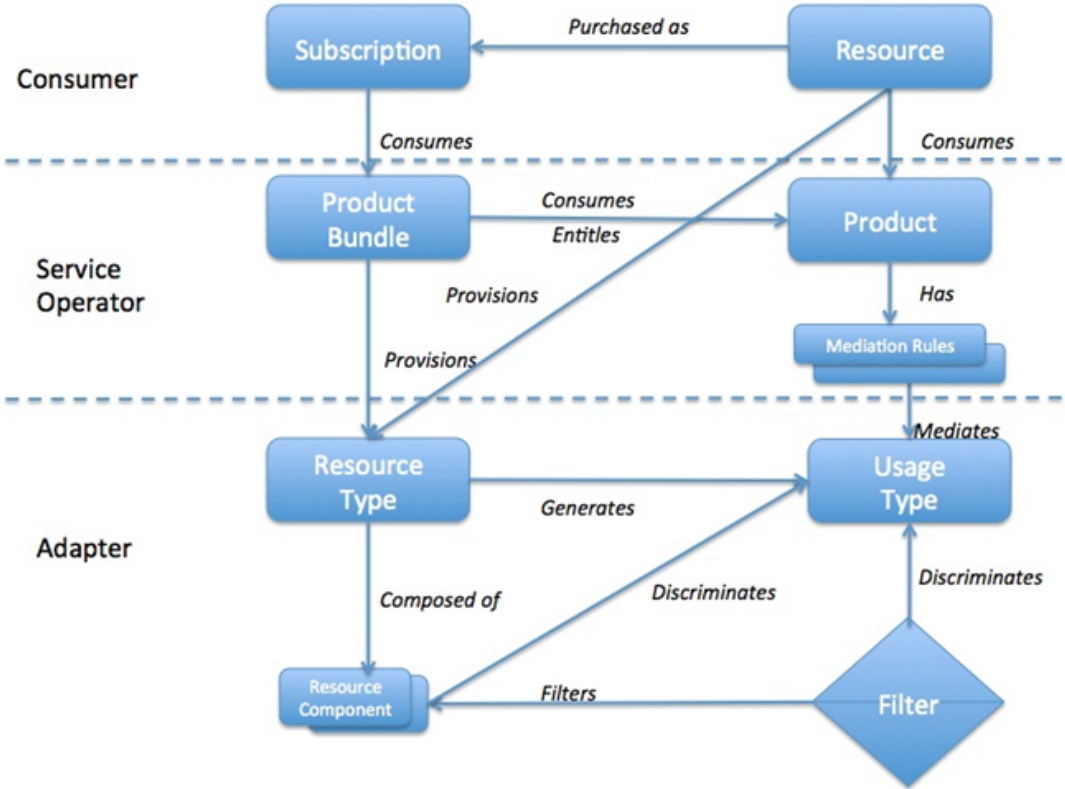


Figure 1. Product Management Inter-relationships

The following sections describe how all of the above come together under various scenarios and use cases.

Product Management

When creating products, CPBM will call `getResourceComponentValues`(first form), `getFilterValues` and `getDiscriminatorValues` to create mediation rules based on all these discriminators. Remember that resource component values as well as filter values are both implicit discriminator candidates.

The interactions between CPBM, the connector metadata and the MetadataRegistry during the creation of products is shown in the figures below.

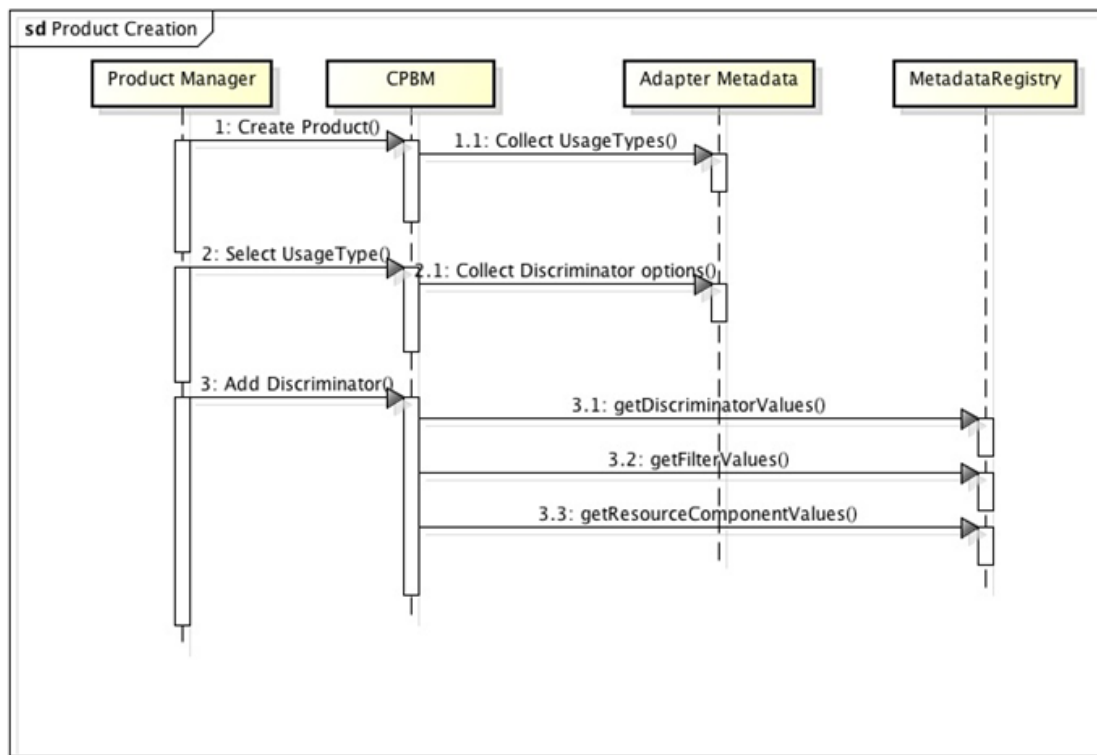


Figure 2. Product creation

When creating product bundles, CPBM will use metadata to identify resource types and call `getResourceComponentValues` to build provisioning constraints.

The interactions between CPBM, the connector metadata and the MetadataRegistry is during the creation of product bundles is shown in the figure below.

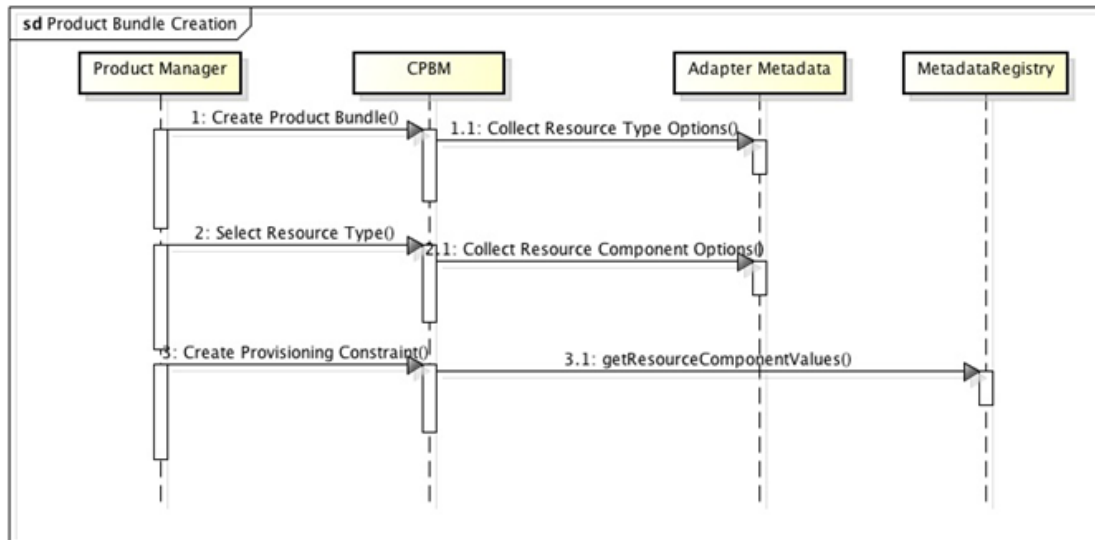


Figure 3. Product bundle creation

Subscription Process

When a customer buys a bundle, the system detects the provisioned resource type and uses that to determine the launch experience. The following steps are performed:

- For a given resource type, the system first identifies the set of filters to present to the user. This is extracted from the service metadata. The filter values are determined by calling `getFilterValues` from the `MetadataRegistry`.
- Once filter values are selected, the system presents the possible choices of resource components to select from. The components are presented in the order defined in service metadata. The possible value choices (with filters applied) are acquired from `getResourceComponentValues` in the `MetadataRegistry` (Note that the appropriate tenant and user handles are passed to these methods.). Selections of prior resource components may affect available choices for subsequent resource components. Bundle choices are further filtered based on bundle provisioning constraints. A choice will not be shown unless there's at least one corresponding bundle whose constraints allow the current selection of resource components.
- Prices are shown for all bundles/products (see below for a discussion on how pricing display is determined)
- Customer selects the bundle to subscribe to
- The subscription is created. If user requested immediate provisioning, then provisioning flow kicks in. Otherwise user can navigate to the subscription and provision at a later time.

Displaying prices

The following items are collected as discriminators:

1. Applied filters and their values
2. Selected Resource components

Based on this, all products that match those discriminators (The match is non-unique, ie it is possible that multiple products match for the above discriminators - this is because they may be differentiated based on other discriminators) and their corresponding prices are shown.

Provisioning Flow

If this resource type depends on another resource type as:

- **ADDON:** The system lists the resources of the target resource type for selection (the handle to that resource is passed to the provisioning call).
- **AVAILABLE:** The system checks to see if a resource of a given target resource type is already provisioned for this user/account (depending on the scope of the target resource type) before proceeding, failing if the target resource type has not been provisioned.

If custom fields are needed for provisioning, they are presented to the user for input. A mechanism to customize the UI for this is available. See [UI contributions](#) for more details. Any resource component selected may be modified at this time, but only within the constraints of the subscribed bundle. The system provides subscription data to the subscription provisioning handler of underlying service to provision the resource.

Sample Product Metadata

```
<?xml version="1.0" encoding="UTF-8"?>
<Service>
  <Bootstrap>
    <Id type="CLOUD" singleton="false" category="IAAS">com.citrix.cloudplatform</Id>
    <Connector ref="cloudPlatformConnector"/>
    <Image type="logo">logo.jpg</Image>
    <Image type="banner">logo.jpg</Image>
    <Vendor version="1.1.6">Citrix</Vendor>
    <Locales>en_US, ja_JP, ko_KR, fr, es, pt_BR, zh_CN</Locales>
    <Configuration>
      <!-- Any Properties that are added here must also be added to the ENUM in MetadataRegistryImpl -->
      <Property id="publicProtocol" type="String" encrypted="false" validation="{required:true, regex:'http|https'}"/>
      <Property id="publicHost" type="String" encrypted="false" validation="{required:true}"/>
      <Property id="publicPort" type="Integer" encrypted="false" validation="{required:true} default='8080' />
      <Property id="cloud.jdbc.host" type="String" encrypted="false" validation="{required:true}"/>
      <Property id="cloud.jdbc.username" type="String" encrypted="false" validation="{required:true} default='sa' />
      <Property id="cloud.jdbc.password" type="String" encrypted="true" validation="{required:true, confirm:'true'} />
      <Property id="cloud.jdbc.database.schemaname" type="String" encrypted="false" validation="{required:true} default='sa' />
      <Property id="cloud.usage.jdbc.host" type="String" encrypted="false" validation="{required:false} />
      <Property id="cloud.usage.jdbc.username" type="String" encrypted="false" validation="{required:false} default='sa' />
      <Property id="cloud.usage.jdbc.password" type="String" encrypted="true" validation="{required:false, confirm:'true'} />
      <Property id="cloud.usage.jdbc.database.schemaname" type="String" encrypted="false" validation="{required:false} default='sa' />
    </Configuration>
  </Bootstrap>
</Service>
```



```

<Property id="ssoKey" type="String" encrypted="true" validation="{\"required\":true}\" />
<Property id="apiKey" type="String" encrypted="false" validation="{\"required\":true}\" />
<Property id="secretKey" type="String" encrypted="true" validation="{\"required\":true}\" />
<Property id="parentDomainId" type="String" encrypted="false" validation="{\"required\":true}\" default='1' />
<Property id="adminServerList" type="String" encrypted="false" validation="{\"required\":false}\" />
<Property id="nonAdminServerList" type="String" encrypted="false" validation="{\"required\":false}\" />
<Property id="apiProxySuffix" type="String" encrypted="false" validation="{\"required\":true, \"regex\":\"^[a-zA-Z]\" />
<Property id="apiWhitelist" type="String" encrypted="false" validation="{\"required\":false}\" />
<Property id="apiBlacklist" type="String" encrypted="false" validation="{\"required\":false}\" />
<Property id="default.vm.locale" type="String" encrypted="false" validation="{\"required\":false, \"regex\":\"de\" />
<Property id="max.custom.disk.offering.size" type="Integer" encrypted="false" validation="{\"required\":false}\" />
</Configuration>
</Bootstrap>

<AccountControls>
  <Property id="sharedAccount" type="Boolean" validation="{\"required\":false}\" />
  <Property id="perAccountPublicIPCount" type="Integer" validation="{\"required\":true}\" unique="true" />
  <Property id="perAccountSnapshotCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perAccountTemplateCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perAccountVmCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perAccountVolumeCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perUserPublicIPCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perUserSnapshotCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perUserTemplateCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perUserVmCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="perUserVolumeCount" type="Integer" validation="{\"required\":true}\" />
  <Property id="defaultNetworkOffering" type="String" validation="{\"required\":false}\" />
</AccountControls>

<ResourceMetadata>
  <ResourceType name="VirtualMachine" constraint="NONE" componentSelector="apachecloudstack.fragment.virtual.machine">
    <Properties editor="apachecloudstack.fragment.resource.virtual.machine.properties.selection">
      <Property id="hostName" type="String" validation="{\"required\":false, \"regex\":\"^[a-zA-Z]{1}[a-zA-Z0-9\\-]{1}\" />
      <Property id="displayName" type="String" validation="{\"required\":false}\" />
      <Property id="group" type="String" validation="{\"required\":false}\" />
      <Property id="diskOfferingUuid" type="String" validation="{\"required\":false}\" />
      <Property id="keypair" type="String" validation="{\"required\":false}\" />
      <Property id="diskSize" type="Integer" validation="{\"required\":false}\" />
      <Property id="keyboard" type="String" validation="{\"required\":false, \"regex\":\"de|de-ch|es|fi|fr|fr-be|fr-ca\" />
      <Property id="networkIds" type="String" validation="{\"required\":false}\" />
      <Property id="securitygroupIds" type="String" validation="{\"required\":false}\" />
      <Property id="userData" type="String" validation="{\"required\":false}\" />
      <Property id="ipAddress" type="String" validation="{\"required\":false, \"regex\":\"^([0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3})\" />
      <Property id="ipToNetworkList" type="String" validation="{\"required\":false, \"regex\":\"^([0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3})\" />
    </Properties>
    <Group name="fromTemplate">
      <Component name="templateUuid" discriminator="templateUuid" />
      <Component name="serviceOfferingUuid" discriminator="serviceOfferingUuid" reconfigurable="true" />
    </Group>
    <Group name="fromISO">
      <Component name="isoUuid" discriminator="isoUuid" />
      <Component name="hypervisorType" discriminator="hypervisorType" />
      <Component name="serviceOfferingUuid" discriminator="serviceOfferingUuid" reconfigurable="true" />
    </Group>
  </ResourceType>
  <Generates>
    <Usage>RUNNING_VM</Usage>
  </Generates>

```

```

    <Usage>ALLOCATED_VM</Usage>
    <Usage>ROOT_VOLUME</Usage>
  </Generates>
</ResourceType>

<ResourceType name="Volume" constraint="NONE">
  <Properties>
    <Property id="name" type="String" validation="{required:true}"/>
    <Property id="size" type="Integer" validation="{required:false}"/>
  </Properties>
  <Group name="fromDiskOffering">
    <Component name="diskOfferingUuid" discriminator="diskOfferingUuid"/>
  </Group>
  <Generates>
    <Usage>VOLUME</Usage>
  </Generates>
</ResourceType>

<ResourceType name="Network" constraint="NONE">
  <Properties>
    <Property id="name" type="String" validation="{required:true}"/>
    <Property id="displayText" type="String" validation="{required:true}"/>
    <Property id="gateway" type="String" validation="{required:true, regex:"^([0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3})$"/>
    <Property id="netmask" type="String" validation="{required:true, regex:"^([0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3})$"/>
    <Property id="networkDomain" type="String" validation="{required:false}"/>
    <Property id="vlan" type="Integer" validation="{required:false}"/>
  </Properties>
  <Group name="fromVPCandNetworkOffering">
    <Component name="vpclId" discriminator="vpclId"/>
    <Component name="networkOfferingUuid" discriminator="networkOfferingUuid"/>
  </Group>
  <Group name="fromNetworkOffering">
    <Component name="nonVPCNetworkOfferingUuid" discriminator="networkOfferingUuid"/>
  </Group>
  <Generates>
    <Usage>NETWORK_BYTES_SENT</Usage>
    <Usage>NETWORK_BYTES_RECEIVED</Usage>
  </Generates>
</ResourceType>

<UsageType name="RUNNING_VM" uom="Compute-Hours">
  <Discriminator>isHaEnabled</Discriminator>
  <Discriminator>guestOsName</Discriminator>
</UsageType>

<UsageType name="ALLOCATED_VM" uom="Compute-Hours">
  <Discriminator>isHaEnabled</Discriminator>
  <Discriminator>guestOsName</Discriminator>
</UsageType>

<UsageType name="ROOT_VOLUME" uom="Byte-Hours">
  <Discriminator>poolType</Discriminator>
  <Discriminator>volumeTags</Discriminator>
</UsageType>

<UsageType name="VOLUME" uom="Byte-Hours">

```

```
<Discriminator>poolType</Discriminator>
<Discriminator>volumeTags</Discriminator>
</UsageType>

<UsageType name="TEMPLATE" uom="Byte-Hours" />

<UsageType name="ISO" uom="Byte-Hours"/>

<UsageType name="SNAPSHOT" uom="Byte-Hours">
  <Discriminator>diskOfferingUuid</Discriminator>
  <Discriminator>poolType</Discriminator>
</UsageType>

<UsageType name="IP_ADDRESS" uom="IP-Hours"/>

<UsageType name="NETWORK_BYTES_RECEIVED" uom="Bytes">
  <Discriminator>isRedundantRouterServiceEnabled</Discriminator>
  <Discriminator>isVlanEnabled</Discriminator>
  <Discriminator>networkType</Discriminator>
</UsageType>

<UsageType name="NETWORK_BYTES_SENT" uom="Bytes">
  <Discriminator>isRedundantRouterServiceEnabled</Discriminator>
  <Discriminator>isVlanEnabled</Discriminator>
  <Discriminator>networkType</Discriminator>
</UsageType>

<UsageType name="SECURITY_GROUP" uom="SecurityGroup-Hours"/>

<UsageType name="LOAD_BALANCER_POLICY" uom="Policy-Hours"/>

<UsageType name="PORT_FORWARDING_RULE" uom="Rule-Hours"/>

<UsageType name="VPN_USERS" uom="Vpn-Hours"/>

<Uom raw="Bytes">
  <Scale name="GB" conversionFactor="1073741824" default="true" />
  <Scale name="TB" conversionFactor="1099511627776" />
</Uom>

<Uom raw="Byte-Hours">
  <Scale name="GB-Months" conversionFactor="1073741824" default="true" monthly="true" />
  <Scale name="TB-Months" conversionFactor="1099511627776" monthly="true" />
  <Scale name="GB-Days" conversionFactor="25769803776" />
  <Scale name="TB-Days" conversionFactor="26388279066624" />
</Uom>

<Uom raw="Compute-Hours">
  <Scale name="Compute-Days" conversionFactor="24" />
</Uom>

<Uom raw="Vpn-Hours">
  <Scale name="Vpn-Days" conversionFactor="24" />
  <Scale name="Vpn-Months" conversionFactor="1" monthly="true" />
</Uom>
```

```
<Uom raw="IP-Hours">
  <Scale name="IP-Days" conversionFactor="24" />
  <Scale name="IP-Months" conversionFactor="1" default="true" monthly="true" />
</Uom>

<Uom raw="Policy-Hours">
  <Scale name="Policy-Days" conversionFactor="24" />
  <Scale name="Policy-Months" conversionFactor="1" default="true" monthly="true" />
</Uom>

<Uom raw="Rule-Hours">
  <Scale name="Rule-Days" conversionFactor="24" />
  <Scale name="Rule-Months" conversionFactor="1" default="true" monthly="true" />
</Uom>

<Uom raw="SecurityGroup-Hours">
  <Scale name="SecurityGroup-Days" conversionFactor="24" />
  <Scale name="SecurityGroup-Months" conversionFactor="1" monthly="true" />
</Uom>

  <Filter name="zoneUuid" />
</ResourceMetadata>

<Roles>
  <Role name="ROLE_CLOUD_MANAGEMENT" scope="GLOBAL" />
  <Role name="ROLE_ACCOUNT_CLOUD_MANAGEMENT" scope="TENANT" />
  <Role name="ROLE_USER_CLOUD_MANAGEMENT" scope="TENANT" />
</Roles>
</Service>
```

Lifecycle Management

A connector integrated with CPBM is required to reflect entities created in CPBM in the corresponding Cloud Service and is responsible for keeping these entities in sync with CPBM during their lifecycle in CPBM. There are 4 entities created in CPBM that need synchronization. They are:

- **Channel:** Channels represent customer segmentation in CPBM. They may refer to Geographies, selling channels, resellers, partners etc.
- **Account:** Accounts (sometimes referred to as Tenants) represent customer billable entities. This is the entity against which usage is aggregated and invoices are generated.
- **User:** Users represent actual people interacting with the system. Users can provision and consume resources in an account. While CPBM can track usage at a user level, billing is aggregated at an account level.
- **Subscription:** Subscriptions are specific commitments for resources that are purchased by users. Users may acquire resources without commitments. These will not be attached to subscriptions, and will be charged on a utility basis. Users own subscriptions and resources.

While subscriptions are tied to resources from specific services, the Channel, Account and User entities are cross service entities, and represent the core organizational structure in CPBM. The following figure shows the organization of these entities in CPBM.

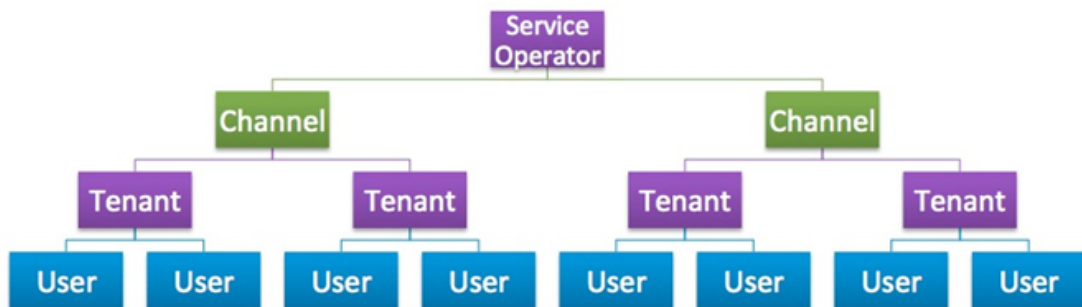


Figure 1. Organizational Structure of CPBM entities

Account Lifecycle

A Connector package can implement account provisioning lifecycle handler so that accounts in CPBM can be mapped to a corresponding entity in the underlying cloud service. CPBM has accounts and each account minimally has an owner, who is called a master user. Users in CPBM initiate all actions and if a service does not differentiate between an account and a user then it should return same handle for both account as well as a user. In such a case the service should not implement the User Provisioning Lifecycle handler. In this case, the service will not be able to expose user specific access.

CPBM creates these mapping lazily i.e. the mapping is not created at CPBM account creation time but rather at first subscription provisioning time. See Subscription Lifecycle handling for details on provisioning.

Connectors should use the account lifecycle handler to be in synchronization with CPBM account lifecycles. Connectors should implement all methods in AccountLifecycleHandler if they need to represent an account in their service. Connectors may choose to not distinguish between certain states (like restrict and suspend) or may choose to do nothing for certain states.

TenantHandle

Tenant handles are represented as Handle objects which an account can set/get from a Tenant object. A tenant handle is the representation of account information in the cloud service in CPBM. For the most part, other than its existence, TenantHandles are opaque to CPBM. A TenantHandle object is represented as follows.

```
public class TenantHandle {
    /* A unique opaque handle to identify the service specific entity corresponding to
       this account */
    public String getHandle();

    /* Can store any json/string data and CPBM will preserve this as it is. This field
       may be used to store any additional data the connector would like to associate with
       this account handle
       */
    public String getData();

    /* Connector can store apiKey/secret or username/passwords etc in this key and values
       in this map would be displayed in the CPBM credentials section.
       */
    public Map<String, String> getApiCredentials();

    /* Back reference to the tenant */
    public Tenant getTenant();
}

public class Tenant {
    ...
    /* Return the handle for the corresponding service instance if exists. Returns null
```

```
        otherwise */
    public TenantHandle getHandle(String serviceInstanceId);
    /* Add an tenant handle to the tenant */
    public void addHandle(String serviceInstanceId, TenantHandle handle);
    /* Remove an tenant handle from the tenant */
    public void removeHandle(String serviceInstanceId);
    ...
}
```

The handle string should be the same one that is used to identify the tenant in the usage record.

AccountLifecycleHandler

The AccountLifecycleHandler SPI is defined as follows:

```
/**
 * Handles account lifecycle.
 */
public interface AccountLifecycleHandler {

    /* Registers an account as well as account owner user in cloud service and stores the
       mapping of the account and account owner user as handles in tenant and
       tenant.owner()passed */
    public void register(Tenant t);

    /* Restrict the account. */
    public void restrict(Tenant t);

    /* Suspend the account */
    public void suspend(Tenant t);

    /* Reactivate the account */
    public void reactivate(Tenant t);

    /* Deletes the account. Connector should clean the handle for tenant by calling
       t.removeHandle(serviceInstanceObj) */
    public void terminate(Tenant t);

    /**
     * Sets service level controls at tenant level. Service level
     * controls are set under the following scenarios
     *
     * <ul>
     * <li> When a tenant is newly created, the controls are applied from the tenant's account type
     * <li> When a tenant is updated with new control values
     * <li> when a tenant moves account types
     * </ul>
     *
     * In all cases, the service is expected to do its best to apply the
     * updated values by triangulating between the tenant's current
     * settings, the new settings and any current limitations. For
     * example, a tenant's current quota, new quota and currently used
     * amount (against that quota).
     */
}
```

```
* @param tenant The tenant to update
* @param controls - Map of control names and their values; Not null when SP edits the service controls
*                  and when a tenant moves account types
*/

public void setControls(Tenant tenant, Map<String, String> controls);

/**
 * Retrieves service level controls defined at tenant level
 * @param tenant
 * @return
 */

public Map<String, String> getControls(Tenant tenant);

}
```

The SPI implementation should implement these methods as follows:

- **register:** Register service specific scope corresponding to an account in CPBM. It is required that the account creation process also handle the master user (owner) creation process as well. The `UserLifecycleHandler` will NOT be called for the master user. **Account handle MUST be added using `Tenant.addHandle`**. The tenant owner can be accessed using `Tenant.getOwner()` and user handle can be added using `User.addHandle()` on the owner object. If the connector does not support users the user handle need not be added.
- **restrict:** restrict all users from managing resources in the cloud service. The resources may still be operational and may be accessible by other means. In general, it is preferred that this state allow users to reduce their resource consumption. It may be sufficient to allow master users access to perform this.
- **suspend:** prevent users from accessing their resources. The resources should as much as possible cease to be operational. For example, VMs should be shut down but not destroyed. Note that this is not a terminal state so the resources should remain recoverable.
- **reactivate:** re-enable access to resources for all users in the account so that they can get back in operational state. It is not required that the service make the resources operational (Eg. VMs can be started but the service need not start the VMs).
- **terminate:** delete the account, all the users and remove all resources used by this account.

For `UserHandle` structure and processing, see `UserLifecycleHandler` section. The same applies to the master user as well.

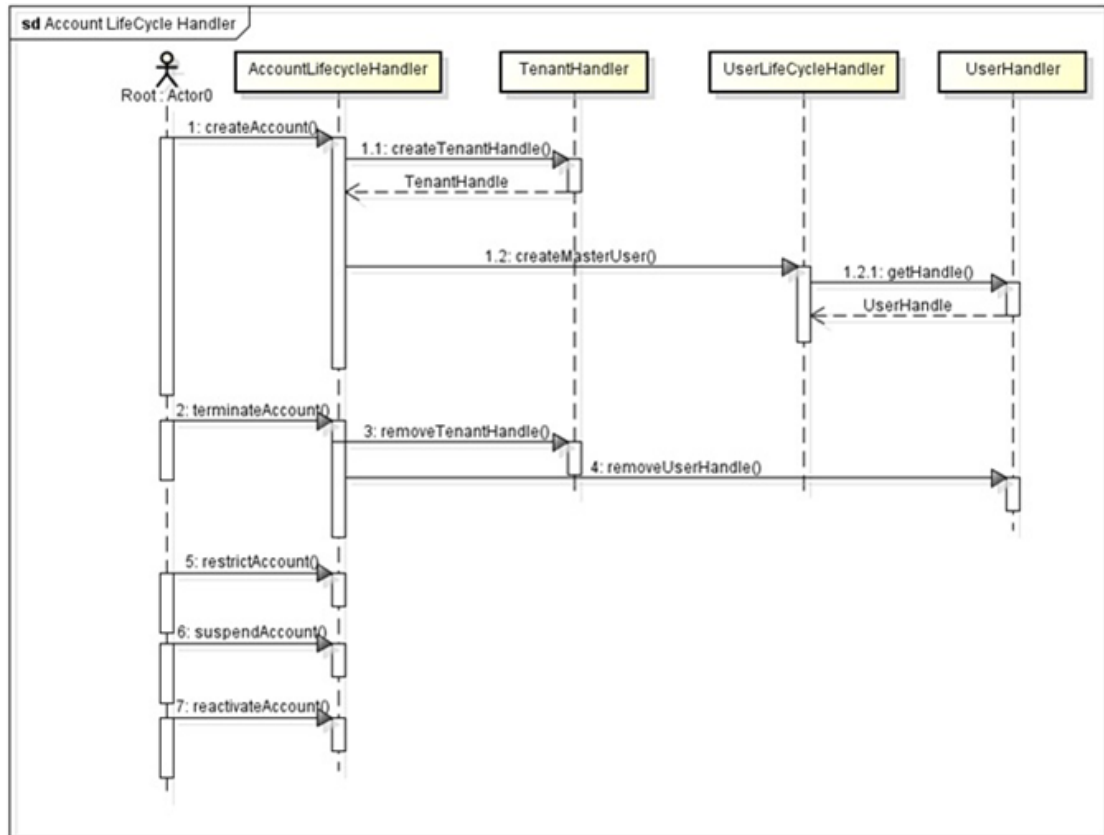


Figure 1. Account Lifecycle Handler

For information on service onboarding a customer, see [Onboarding and managing a service](#).

User Lifecycle

Connectors should implement the user lifecycle SPI if the cloud service supports users within accounts and resources owned by users. This SPI will be called by CPBM to keep users in services in sync with the user lifecycle in CPBM.

UserHandle

The UserHandle is an opaque handle that represents a given user in a service instance.

```
public class UserHandle {
    /* A unique opaque handle to identify the service specific entity corresponding to
       this user */
    public String getHandle();

    /* Can store any json/string data and CPBM will preserve this as it is. This field
       may be used to store any additional data the connector would like to associate with
       this user handle
       */
    public String getData();

    /* Connector can store apiKey/secret or username/passwords etc in this key and values
       in this map would be displayed in the CPBM credentials section.
       */
    public Map<String, String> getApiCredentials();

    /* Back reference to the user */
    public User getUser();
}

public class User {
    ...
    /* Return the user handle for the corresponding service instance if exists. Returns
       null otherwise */
    public UserHandle getHandle(String serviceInstanceId);
    /* Add an user handle to the user */
    public void addHandle(String serviceInstanceId, UserHandle handle);
    /* Remove a user handle from the user */
    public void removeHandle(String serviceInstanceId);
}
```

The handle string should be the same one that is used to identify the user in the usage record.

UserLifecycleHandler

The UserLifecycleHandler is defined as follows:

```
/**
 * Handles user lifecycle.
 */
public interface UserLifecycleHandler {

    /** Registers a user in the cloud service and stores the mapping of the user as
        handles in user object passed as parameter. User handle must be added as
        User.addHandle. */
    public String register(User user);

    /** Lock the user */
    public void lock(User user);

    /** Disable the user */
    public void disable(User user);

    /** Unlock the user */
    public void unlock(User user);

    /** Enable the user */
    public void enable(User user);

    /** Terminate the user. In this case user handle should be cleaned. This can be done
        as user.removeHandle */
    public void terminate(User user);

/**
 * Sets service level controls at user level. User level
 * controls are set under the following scenarios
 *
 * <ul>
 * <li> When a user is newly created, the controls are applied from the tenant's account type
 * <li> when a tenant moves account types
 * </ul>
 * In all cases, the service is expected to do its best to apply the
 * updated values by triangulating between the tenant's current
 * settings, the new settings and any current limitations. For
 * example, a tenant's current quota, new quota and currently used
 * amount (against that quota).
 *
 * @param user The user to update
 * @param controls - Map of control names and their values; Not null when SP edits the service controls
 * and when a tenant moves account types
 */

    public void setControls(User user, Map<String, String> controls);

/**
 * Retrieves service level controls defined at tenant level
 * @param user The user
 * @return
 */

    public Map<String, String> getControls(User user);
```

}

The UserLifecycleHandler SPI methods are as follows:

- **register:** Register a scope corresponding to this user in the cloud service
- **lock:** Prevent the user from accessing their resources from the service UI/API. The resources can remain operational.
- **disable:** Prevent user from accessing resources. Reduce all resource consumption while allowing reactivation.
- **unlock:** allow users access to their resources again
- **enable:** allow users to access their resources again. Ideally, enable will remember the lock state of the user. This allows blanket disable/enable without changing lock state.
- **terminate:** Remove the user from the service

A note on the difference between locking and disabling. Lock simply prevents access to resources. It does not reduce resource consumption. It may be used for minor infractions such as login failures etc. However, disable reduces resource consumption (say by shutting down VMs etc) and is expected to put the user in a state of minimal resource consumption, while still allowing the system to recover that user back to operational state. Disable is also expected to remember (ideally) the lock state so that when a user is locked and then disabled and re-enabled, they continue to be in the locked state. However, in practice, it is not expected that all cloud services support this nuance. So, it is acceptable for disable to behave exactly like lock.

User Lifecycle

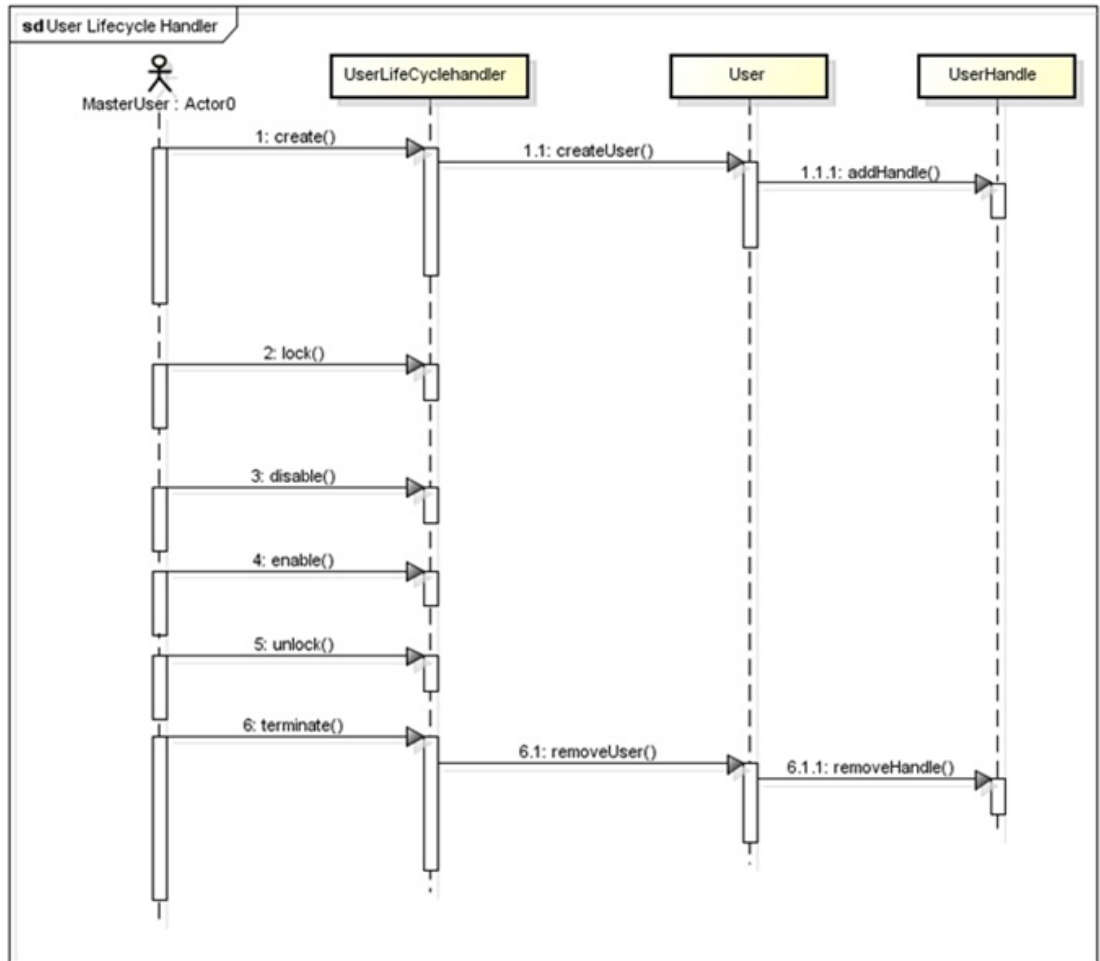


Figure 1. User Lifecycle Handler

Subscription Lifecycle

For the most part resource lifecycles are decoupled from subscription lifecycles. However, in order to enable CPBM to provision and de-provision a resource when subscriptions need to be provisioned and terminated respectively, an connector should implement the below interface to handle provisioning and termination of resources managed by them. As before, CPBM expects the connector to maintain a handle on CPBM to refer to a resource that was provisioned as part of a subscription.

SubscriptionHandle

A SubscriptionHandle for a subscription represents the resource in the service corresponding to a subscription. It is defined as follows:

```
public class SubscriptionHandle {
    /* A unique opaque handle to identify the service specific entity corresponding to
       this resource */
    public String getHandle();

    /* Can store any json/string data and CPBM will preserve this as it is. This field
       may be used to store any additional data the connector would like to associate with
       this resource handle
    */
    public String getData();

    /* Back reference to the subscription */
    public Subscription getSubscription();
}
public class Subscription {
    ...
    /* Return the subscription handle if exists. Returns null otherwise */
    public SubscriptionHandle getHandle();
    /* Add a subscription handle to the resource */
    public void addHandle(SubscriptionHandle handle);
    /* Remove an subscription handle from this subscription */
    public void removeHandle();
    /* Returns provisioning data as a Map */
    public Map<String, String> getConfigurationMap();
}
```

Provisioning configuration data for a given subscription can be retrieved from the subscription by calling `getConfiguration()`. This will contain all configuration information (resource component selections, filters, custom configuration data) used to provision a resource for this subscription.

The handle string should be the same one that is used to identify the resource in the usage record.

SubscriptionLifecycleHandler

SubscriptionLifecycleHandlers are used to synchronize subscription states with the underlying service. The SubscriptionLifecycleHandler is defined as follows:

```
/**
 * Handles subscription lifecycle.
 */
public interface SubscriptionLifecycleHandler {

    /* Create a resource based on configuration provided in subscription. Set the handle
    of the resource back to subscription object as s.addHandle(handle). Note
    that unlike tenants and users where the same CPBM entity has representation in
    multiple services, you do not need to identify the service instance for a
    subscription as a subscription is already tied to a single service instance. */
    public void provision(Subscription s);

    /* Validate if the resource tied to this subscription is still active. */
    public void validate(Subscription s);

    /* Destroy the resource and remove the subscription handle field by calling
    s.removeHandle. Subscription configuration is no longer valid. */
    public void destroy(Subscription s);
}
```

The SubscriptionLifecycleHandler methods should be implemented as follows:

- **provision:** Provision a resource corresponding to this subscription based on configuration data provide by Subscription.getConfiguration()
- **validate:** Validate whether a resource (if exists) continues to be active for a given subscription. This method will be called periodically or on demand to verify that the underlying resource tied to this subscription remains active. It is possible that resource is no longer active, if the resource failed to launch or if the resource was destroyed or removed directly on the service through other means.
- **destroy:** Destroy the resource corresponding to this subscription as the subscription is now terminated.

Subscription Lifecycle

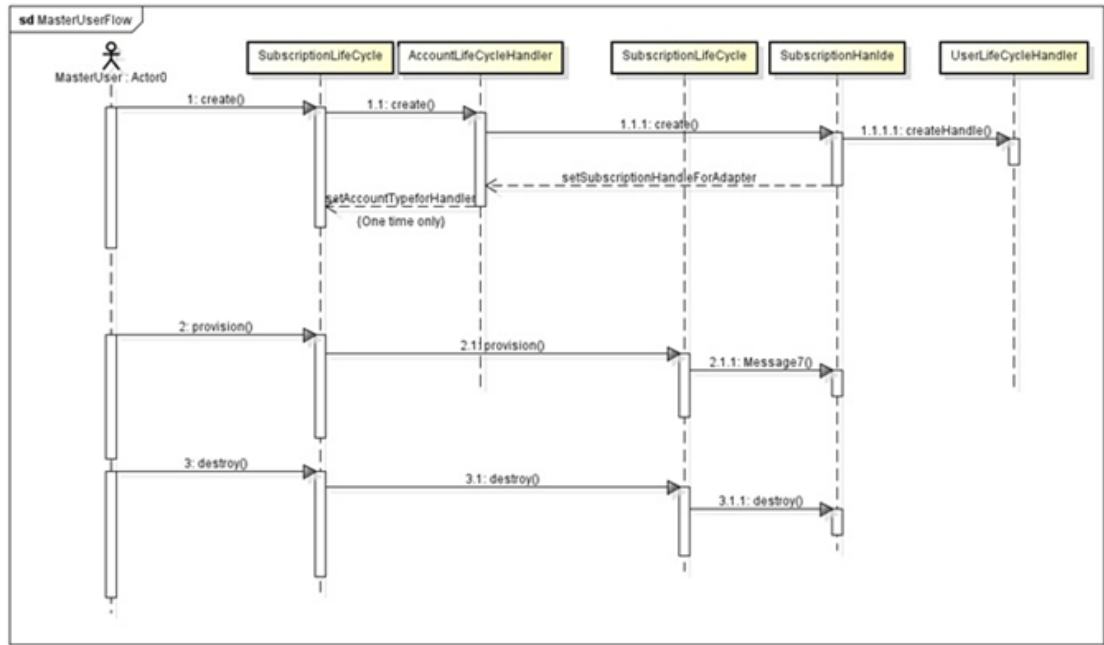


Figure 1. Subscription Lifecycle Handler

Usage Collection

In order for CPBM to accurately bill a customer for usage of resources from an underlying cloud service, CPBM needs to periodically collect usage from the cloud service and be able to understand and mediate it. Mediation is the process by which CPBM identifies the product corresponding to usage received from an underlying cloud service. CPBM does this based on a set of mediation rules attached to a product. A usage record is matched to a product based on its usage type and matching discriminator values between what is collected for the usage record and what is specified in a mediation rule for the product.

As described earlier, in order for this to happen, CPBM periodically collects usage in a well-defined format that is understood by CPBM's mediation engine. It is the responsibility of the Usage collector implementation to transform usage from the underlying resource to the standard format recognized by CPBM.

A connector must implement a Usage collector for its underlying cloud service. The connector (i.e., the Usage collector implementation) shall return usage in terms of UsageData objects. Each UsageData object describes a distinct usage record and should include values for all discriminators described in the product metadata (this includes resource type, filter and additional discriminator values).

UsageData

UsageData is defined as follows:

```
public class UsageData {
    /* the usage type for this resource. Should match the type specified in the metadata
    */
    String getUsageType();
    /* The tenant handle if any. Should match the handle for owning tenant associated
    in the account lifecycle handler */
    String getTenantHandle();
    /* The user handle if any. Should match the handle for the owning user associated
    in the user lifecycle handler */
    String getUserHandle();
    /* The resource handle, if any. Should match the handle for this tenant
    associated in the subscription lifecycle handler */
    String getResourceHandle();
    /* Start date/time for this usage record */
    Date getStartDate();
    /* End Date/Time for this usage record */
    Date getEndDate();
    /* service specific reference id for this usage record. This should be the same
    format used by the markers. */
    String getReferenceld();
    /* Discriminator values for *all* declared discriminators for this usage. This
    includes resouce type ids, filter values and other discriminators declared for
    this usage type in the metadata */
    Map<String, String> getDisriminators ();
    /* Raw usage. Actual usage consumed, in the unit of measure described in metadata */
}
```

```
    Double getUsage();  
}
```

UsageCollector

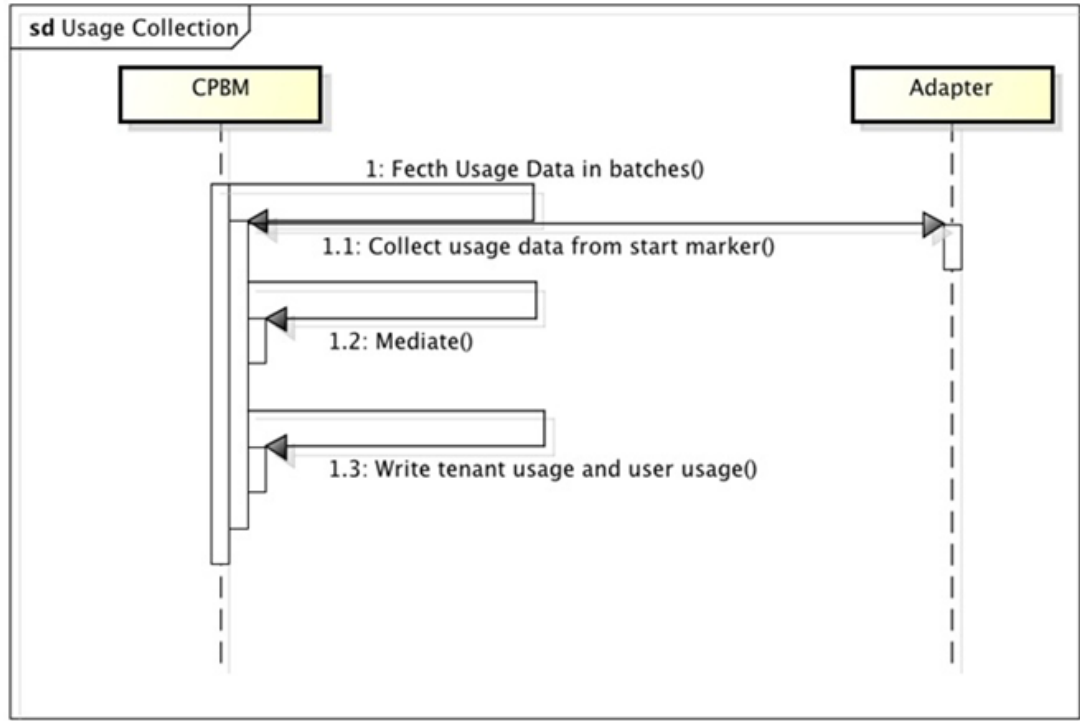
The UsageCollector SPI defines the protocol which CPBM uses to collect usage from the underlying service on a periodic basis. It is defined as follows:

```
public interface UsageCollector {  
    /* Returns a batch of usage data collected from the cloud service. */  
    public UsageResultSet collect(String startMarker, Date endDate);  
}
```

The UsageCollector is called periodically to collect data from the underlying cloud service. CPBM will schedule a job to collect this data at an operator defined frequency. In any given run CPBM will repeatedly call the UsageCollector to collect usage, until it exhausts available data. The startMarker and endDate determines the range of usage data to collect. When collect is called, the returned usage data:

- should start from after the *startMarker* and it must be generated on or before the passed *endDate*. For better performance, we recommend all implementations of the *UsageCollector* return usage data in small batches. The batch size should typically be in hundreds. Any subsequent call to *UsageCollector.collect()* should return the next batch of *usageData*. In case there is no *usageData* left, return *UsageResultSet* with the input startMarker as the marker and null *usageData* list. CPBM will continuously call *UsageCollector.collect()* for a given *endDate*, until all data is exhausted for this time range.
- In case of conflicting *usageData* (i.e., duplicate *referenceId*) CPBM will resolve the conflict by using the first record received.
- The *UsageResultSet* object contains *List of UsageData*. This list should be sorted by *UsageData.endDate* from oldest to newest. Also, for any two calls to *UsageCollector.collect*, all of *UsageData* collected in the later call should be newer (have later *UsageData.endDate*, than the data collected in the earlier call).

The interaction between CPBM and Cloud Service for collecting usage is shown the figure below



powered by Astah

Figure 1. Usage Collection Process

Event Collection

Similar to usage collection, CPBM periodically can collect events from the cloud service and expose it to the end customer and operators from CPBM. While event log may be displayed in the resource UI of the service, CPBM offers a rich set of notification capabilities allow the end user to customize what events they would like to receive via one or more notification channels including email and SMS.

The EventCollector model is very similar to that of the UsageCollector. CPBM will periodically call the EventCollector to collect events in a normalized form for CPBM to store and operate upon. It is the EventCollector's responsibility to map events from the underlying cloud service to the normalized form prescribed by CPBM.

An implementation of the EventCollector SPI will be frequently (typically every few minutes) to collect events from the cloud service. Because it is being called frequently, EventCollectors should be fast as well as prudent in the set of events they expose. Typically, since CloudServices will be SSO enabled, local login events in the underlying service should, for example, be excluded. Otherwise, the event stream will get too noisy. Each call of the EventCollector will return an EventResultSet containing one batch of events.

EventResultSet

The returned Event objects are enclosed in an EventResultSet to allow for batch operations. EventResultSet is defined as follows:

```
/**
 * A result set returned for each run of the collector
 */
public class EventResultSet {
    /* The marker for the last record fetched. This is stored and passed back by CPBM
    during the collection process to identify the last record fetched.
    public String getMarker();
    /* List of Event objects in this batch */
    List<Event> getEvents();
}
```

The EventResultSet has the following elements:

- **marker:** Marks the last record returned by the collector. This is an opaque handle that is passed back to the collector so it can continue collecting event data. This should never be null. When data is exhausted for a given run, the marker should return the input marker as the last marker as indication that there is no more data.
- **Event List:** The returned list of *Events*. Calls to EventCollector should return Event in reasonable batch sizes. CPBM does not impose batch sizes except to offer guidance that this should be small to prevent memory overruns.

Event

An Event is defined as follows:

```
public class Event {
    enum Severity { CRITICAL, ALERT, INFO };
    /* return the severity level */
    public Severity getSeverity();

    /* the time stamp of event generation */
    public Date getCreatedAt();

    /* Message code */
    public String getCode();

    /* handle of the user */
    public String getUser();

    /* Unique identifier of event in the Cloud service. Same format as the marker */
    public String getId();

    /* Event category as classified by the cloud service */
    public String getCategory();

    /* Dynamic attributes that represent the subject of this event */
    Map<String, String> getAttributes();
}
```

EventCollector

The EventCollector SPI defines the interface connectors have to implement to support event collection. Note that this SPI is optional. It is defined as follows:

```
public interface EventCollector {
    /* Returns a list of events collected from the Cloud Service. */
    public EventResultSet collect(String startMarker, Date endDate);
}
```

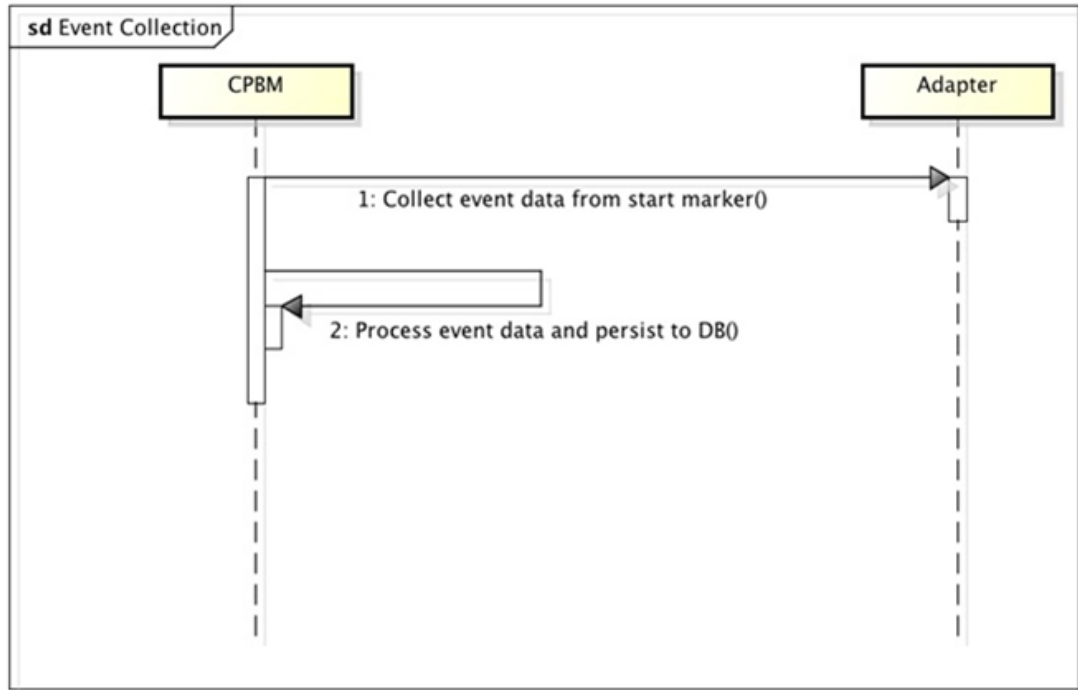
The EventCollector is called periodically to collect event data from the cloud service. CPBM will schedule a job to collect event data at an operator defined frequency. In a given run, CPBM will repeatedly call the EventCollector until it exhausts all available data for the range specified between startMarker and endDate. When collect is called, the returned events list:

- should start from after the *startMarker* and it must be generated on or before the passed *endDate*. For better performance, we recommend all implementations of *EventCollector* return usage data in small batches. The batch size should typically be under a hundred. Any subsequent call to *EventCollector.collect()* should return the next batch of *Events*. In case there is no *Events* left, return *EventResultSet* with the input startMarker as the marker and null *events* list. CPBM will continuously call *EventCollector.collect()* for a given *endDate*, until all data is exhausted for this time range.

Event Collection

- In case of conflicting *Event* (i.e., duplicate *id*) CPBM will resolve the conflict by using the first record received.

The interaction between CPBM and Cloud Service for collecting events is shown the figure below



powered by Astah

Figure 1. Event Collection Process

UI Contributions

CPBM can allow the user to have a seamless experience moving from its UI in the resource management UI of the cloud service. To support this capability, the connector needs to provide some metadata and implement a ViewResolver. The role of the ViewResolver is to resolve URLs in the target cloud service's UI to allow a user to navigate to a given context. Connector implementer can contribute to four types of UI views and they are:

1. **Resource View:** There can be one or more resource views. Within the context of a service instance, each resource view can be presented by CPBM as a sub navigational element. Alternatively, a cloud service can use just one view and handle sub navigation internally.
2. **Subscriptions Details View:** A view that provides information on a resource associated with a subscription.
3. **Account Settings View:** A view that provides account level settings for this cloud service.

User Settings View: A view that provides user level settings for this service.

ResourceView

A Connector can define multiple resource views, each of which will be presented as a sub-tab or window with the navigation provided by CPBM. This allows for UIs built with navigation menus that are incompatible to fit their leaf pages directly into CPBM. In the serviceDefinition.xml file, a connector can list codes that represent each resource view they wish to present to CPBM.

View

The View object represents a UI view contributed by the cloud service connector into CPBM. It is defined as follows:

```
/*
 * Used to represent a UI contribution to CPBM
 */
public class View{
    public enum ViewMode { WINDOW, IFRAME };
    /* Unique name for the View */
    String getName();

    /* Url of the View */
    public String getURL();

    /* The view mode (ie should it be embedded as an iFrame or shown as a window */
    public ViewMode getMode();
```

```
    /* Path to the image used to represent it in the UI */  
    private String getIcon();  
}
```

ViewResolver Interface

The ViewResolver interface allows CPBM to resolve URLs for specific contexts from the underlying cloud service. This allows the connector to produce URLs in any style (query string, REST style etc). Each of these methods essentially returns a View object that includes a URL and additional metadata. The ViewResolver is defined as follows:

```
public interface ViewResolver {  
  
    /** Returns list of resource view */  
    public List<View> listResourceViews(User user);  
  
    /* Returns the Subscription detail view */  
    public View resolveSubscriptionDetailView(Subscription subscription);  
  
    /* Returns the Account setting view */  
    public View resolveAccountSettingsView(Tenant tenant);  
  
    /* Returns the User setting view */  
    public View resolveUserSettingsView(User user);  
  
    /*Returns the Administrative view of the cloud service*/  
    public View getConsoleView(User user);  
  
}
```

The URLs to the Subscription Details, Account Settings and User Setting are resolved using `resolveSubscriptionDetailsView()`, `resolveAccountSettingsView()`, `resolveUserSettingsView()` respectively. Depending on the number of resources that the underlying cloud service offers, `listResourceViews()` should return the list of Views.

Note: If there is only one section defined, it is assumed that a tabbed UI structure is not required.

Single Sign-on Setup

In order to enable a seamless experience switching between cloud service UIs and CPBM UI, it is strongly encouraged that the connector implement or integrate with a single sign on scheme. CPBM comes out of the box with support for a couple of ways to SSO into a target cloud service system.

Using CAS (Central Authentication Service)

This is most preferred way to achieve SSO. CPBM also ships CAS, which runs on the same server as CPBM. Any Cloud Service, which is capable of authenticating using CAS, can be added to the CAS server to achieve SSO.

Steps for enabling SSO using CAS are as follows:

Add the service URL in the configuration file "deployerConfigContext.xml" as show below:

```
<bean id="serviceRegistryDao"
  class="org.jasig.cas.services.InMemoryServiceRegistryDaoImpl">
  <property name="registeredServices">
    <list>
      <bean class="org.jasig.cas.services.RegisteredServiceImpl">
        <property name="id" value="0" />
        <property name="name" value="CloudPortal" />
        <property name="description" value="CloudPortal Service" />
        <property name="serviceId"
          value="http://CPBMHost/portal/portal/j_spring_security_check"/>
      </bean>

      <bean class="org.jasig.cas.services.RegisteredServiceImpl">
        <property name="id" value="1" />
        <property name="name" value="Cloud Service" />
        <property name="description" value="Cloud Service XaaS" />
        <property name="serviceId" value="<xaas.service.url>" />
      </bean>
    </list>
  </property>
</bean>
```

Set the login and logout URL in the Cloud Service as ***https://<CPBM host>/cas*** , and ***https://<CPBM host>/cas/logout***, respectively.

Import the CAS server certificate into the system truststore of the Cloud Service server.

Implementing “SsoHandler” Interface

This method is used to achieve SSO by creating Cookies and/or by making direct login calls to Cloud Service. If the Cloud Service is running on the same domain or single level sub-domain then setting the cookies would be sufficient to achieve SSO. Cloud Service can extend the below interface to inject the cookies into the response object.

```
public interface SSOHandler {  
  
    /* This method returns the SSO Object which contains list of Cookies that will be  
    * added to the response, and a string which will be used by the js file (The js  
  
    * file is provided by CloudConnector as part of meta data and injected into  
    * the UI) */  
    public SsoObject handleLogin(User user);  
}  
  
public class SsoObject {  
    /* List of cookies to set */  
    List<Cookie> getCookies();  
  
    /* callback method to call (optional) */  
    String getCallback();  
  
    /* SSO JSON String to return (optional) */  
    String getSSOString();  
}
```

Sample Implementation of SSOHandler interface:

```
public class XaaSHandler implements SSOHandler {  
  
    @Override  
    public SsoObject handleLogin(User user) {  
        Cookie cookie = new Cookie("XAASUSER", "abcd-efgh");  
        String ssoData = "{ 'ssoData': { 'command': 'login', 'signature': 'qWErtYUiop/+=} }";  
        SsoObject ssoObject = new SsoObject();  
        ssoObject.getCookies().add(cookie);  
        ssoObject.setCallback('handleSSO');  
        ssoObject.setSSOString(ssoData);  
        return ssoObject;  
    }  
}
```

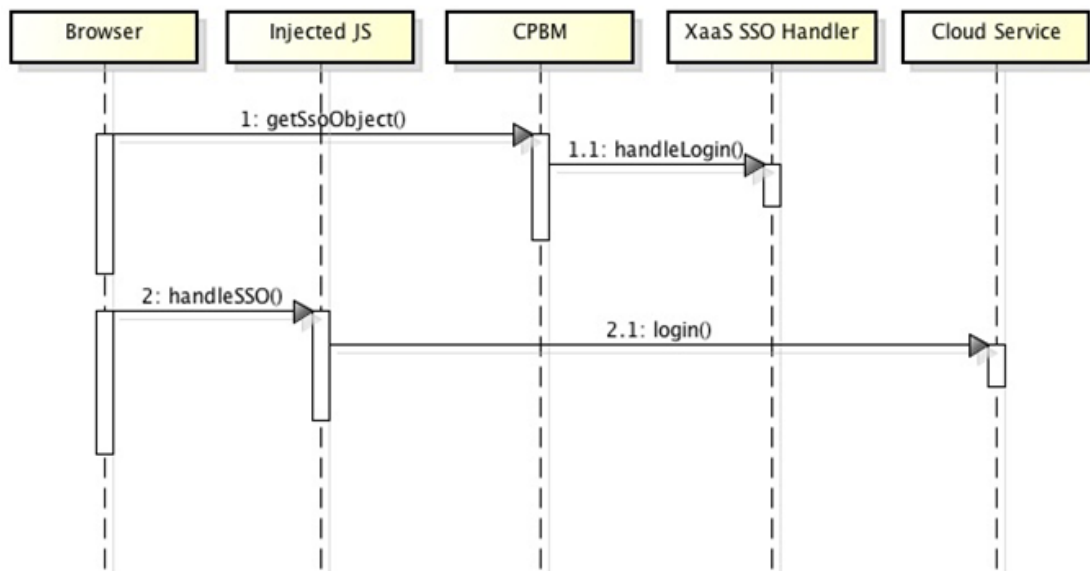
Along with implementing the "SSOHandler" interface, the Cloud Service can optionally provide a javascript file as part of meta data which CPBM will inject into the UI just before the iFrame of Cloud Service. This javascript file will have the implementation of the callback method specified in the SSO object returned by SSOHandler, and this method will take input parameters as the SSO data from SSO Object and the URL of the Cloud service view along with a callback function to call. The callback method may then make a call to the target service to single sign on. Typically, this should be a JSON-P call.

Steps to SSO into Cloud Service:

1. CPBM will inject the javascript, provided by the Cloud Service, into the UI.
2. CPBM will make an AJAX call to the SSOHandler implementation.
3. CPBM will set the cookies returned in the SSO Object into the response. If the cookies are intra domain and have the correct path for the target service to read, this may be sufficient to single-sign on. If not, the callback method mechanism (see below) offers additional options to achieve SSO.
4. If callback property exists, CPBM will call the callback method as returned in the SSO Object, it will pass the SSO data and view URL as input parameters to the callback method.
5. Callback method will make JSONP call to Cloud Service for login, and use the returned values to generate final view URL. It then calls the supplied callback function

A sample cloud service callback method is shown below:

```
function handleSSO(Var ssoData, Var iFrameURL, var callback){
    var finaliFrameURL;
    $.ajax({url: "http://xaas.service.com/api/login",
        dataType: 'jsonp',
        data: ssoData.ssoData,
        jsonp : "jsoncallback",
        success: function(data) {
            finaliFrameURL = iFrameURL + "&sessionKey=" + data.sessionKey;
            callback(finaliFrameURL);
        }
    });
}
```



The interaction between CPBM and Cloud Service for achieving SSO is shown in the figure below:Figure 1. Single Signon Process

Development Using SDK

This section is for the developers/partners who would implement XAAS (Anything as a service) Connector from CPBM. This document is divided in the following major categories.

1. System Requirements
2. Downloading and Installing the Cloud Service SDK
3. Downloading and Installing the CPBM Development Server
4. Creating the CS SDK Maven STS Project
5. Various package structures and their responsibilities
6. Default classes generated as part of the CPBM script and its function
7. Spring OSGI Service Registration Listeners
8. Location of serviceDefinition.xml
9. Writing Cloud Service Connector

System Requirements

The document assumes that the developer machine is installed with the following system requirements and the developer is also aware of the same.

1. Windows XP (32-Bit), Windows Vista or Windows 7 (32- or 64-Bit), or MacOSX, or Linux
2. JDK Version 1.6
 - a. Set `JAVA_HOME` environment variable to the root of the JDK install path.
 - b. Add `%JAVA_HOME%\bin` to the `PATH` environment variable.
3. Spring Tool Suite (STS) Version 3.1.0 or higher.
4. Apache Maven Version 3.
 - a. Add `<Maven Install Path>\bin` to the `PATH` environment variable.
5. CPBM Server

Downloading and Installing the CS-SDK

1. Download the CPBM Cloud Service SDK from [here](#).
2. Unzip the content to a local directory in your system.
3. Double-click `cssdk-standard.jar` or run the command `java -jar cssdk-standard.jar` to launch CPBM Cloud Service SDK installer.

Creating the CS-SDK Maven STS project

1. Start SpringSource Tool Suite.
2. Goto *File->New->Project....*
3. Select *Maven->Maven Project*.
4. In the New Maven Project Wizard Screen make sure *Create a simple project* is unchecked.
5. Click *Next*.
6. On the next Screen for the *Catalog* selection, select *Default Local*.
7. From the table select item with
 - a. Group Id: *com.citrix.cpbm*
 - b. Artifact Id: *com-citrix-cpbm-cssdk-archetype*
8. Click *Next*.
9. On the next screen enter your values for the following fields:
 - a. GroupId - A group under which all services are offered. For instance, com.citrix in case of Citrix.
 - b. ArtifactId - Name of the services which the above group offers. com.citrix.cpbm.cs in case of CloudPlatform service from Citrix.
 - c. Version - Version of the service offered. For example, for CloudPlatform 3.0.2, the version is 3.0.2.
 - d. Package - Base Java package for the maven project com.citrix.cpbm.cs.sdk in case of Citrix CloudPlatform Service.
10. Set **ClassNamePrefix** to a preferred prefix for Class Names for the project generated by the Maven archetype.
11. Click *Finish*. This will create a CPBM CS SDK Project.

Note: Updating Maven dependencies on the first project creation might take some time. This is a one-time update.

Package Structure

Referring to the example mentioned above, we specified **com.citrix.cpbm.cs.sdk** to be the base package for the service that this connector would offer. Taking **com.citrix.cpbm.cs.sdk** as the base, the script will generate the below Java packages.

1. com.citrix.cpbm.cs.sdk.connector - Contains Connector Code

2. com.citrix.cpbm.cs.sdk.collectors - Contains Collector Code
3. com.citrix.cpbm.cs.sdk.cloud.service - Contains Implementation of Cloud Service
4. com.citrix.cpbm.cs.sdk.lifecycle.handlers - Contains various Life Cycle Handlers
5. com.citrix.cpbm.cs.sdk.views - Contains the View resolver

Default Classes

The project will generate five packages and will have default classes included in these packages. You must rename these classes to suit the functionality provided by the class.

Note: It is a default project structure generated for the ease of use so that the packages/classes can be reorganized.

Java package	Class Name	Functionality
com.citrix.cpbm.cs.sdk.connector		
	<ClassNamePrefix>ServiceConnectorImpl	Implementation class for com.citrix.cpbm.platform.spi.CloudServiceConnector
	<ClassNamePrefix>MetadataRegistry	Implements com.citrix.cpbm.platform.spi.MetadataRegistry
com.citrix.cpbm.cs.sdk.service		
	<ClassNamePrefix>ServiceImpl	Implements CloudService. Provides the actual service that is offered by this bundle.
com.citrix.cpbm.cs.sdk.lifecycle.handlers		
	<ClassNamePrefix>AccountLifecycleHandler	Implements com.citrix.cpbm.platform.spi.AccountLifecycleHandler.
	<ClassNamePrefix>SubscriptionLifecycleHandler	Implements com.citrix.cpbm.platform.spi.SubscriptionLifecycleHandler
	<ClassNamePrefix>UserLifecycleHandler	Implements com.citrix.cpbm.platform.spi.UserLifecycleHandler
com.citrix.cpbm.cs.sdk.collectors		

	<ClassNamePrefix>UsageCollector	Implements com.citrix.cpbm.platform.spi.UsageCollector
	<ClassNamePrefix>EventCollector	Implements com.citrix.cpbm.platform.spi.EventCollector
com.citrix.cpbm.cs.sdk.views	<ClassNamePrefix>ViewResolver	Implements com.citrix.cpbm.platform.spi.ViewResolver

Location of Service Definition File

serviceDefinition.xml file is located inside "/com.citrix.cpbm.cs/src/main/resources/serviceDefinition.xml". When the mvn install is run, the config file is copied to the root folder of the package. By default, CPBM platform searches for "serviceDefinition.xml" file in the root folder of the deployed bundle. On locating the file, CPBM would parse the xml file and execute the steps to register the Cloud Service.

CloudService Connector

CPBM invokes the initialize method of com.citrix.cpbm.cs.sdk.connector.<ClassNamePrefix>ServiceConnectorImpl whenever required, to retrieve a CloudService instance. On this instance, corresponding lifecycle handlers and collectors are invoked.

Building OSGi Bundle

Run mvn clean install at the root of the project path. An OSGi bundle jar will be generated in the target folder.

Publishing the Connector to CPBM

1. Secure copy the created bundle jar to the CPBM Development Server at the location /usr/share/vts3/connectors_ext.
2. Add the line <artifact type="bundle" name="<bundle.artifact.id>" version="[<x.x.x>, <x.x.x>]"/> within the plan tag in the file com.citrix.cpbm.connector.plan located at /usr/share/vts3/connectors_ext within the section marked by the xml comment <!--Cloud Service Connectors-->
3. To stop CPBM Development Server, run command:

```
service cloud-portal stop
```
4. To start CPBM Development Server, run command:

```
service cloud-portal start
```

Connector Fragment

Creation of connector fragment from Archetypes:

1. Create a new connector fragment maven project by following the steps given below:
 - a. Go to File->New->Project...
 - b. Select Maven->Maven Project.
 - c. In the New Maven Project Wizard screen, make sure Create a simple project is unchecked.
 - d. Click Next.
 - e. In the next screen, for the Catalog selection, select Default Local in Catalog drop down.
 - f. From the table select item with:
 - i. Group Id: com.citrix.cpbm
 - ii. Artifact Id: com-citrix-cpbm-connector-fragment-archetype
 - g. Click Next .
 - h. In the next screen, enter your values for the following fields:
 - i. GroupId: A group under which all services are offered. For instance, org.apache in case of CloudStack.
 - ii. ArtifactId: Name of the services which the above group offers, cloudstack.fragment in case of Apache CloudStack.
 - iii. Version: Version of the service offered. For example, for cloudstack.fragment, the version is 1.0.0.
 - iv. Package: Base Java package for the maven project, org.apache.cloudstack in case of Apache CloudStack.
 - i. Click Finish. This will create a Connector Fragment Project.

Note: If project does not get created, try restarting the STS once.

Default Structure:

```
META-INF
  readme.txt
src
  main
    java
      base package name
```

```
    ExampleController.java
resources
  css
    ${artifactId}
    example.css
  js
    ${artifactId}
    example.js
  images
    ${artifactId}
    example.png
  META-INF
    MANIFEST.MF
  WEB-INF
    default
      ${artifactId}-tiles-defs.xml
    jsp
      ${artifactId}
      example.jsp
      applicationContext-${artifactId}-fragment.xml
build.properties
pom.xml
template.mf
```

File Description:

- ExampleController.java: It is a Spring MVC controller which will contain the url mapping, which is returned from the connector's view resolver. it returns a tiles view.
- MANIFEST.MF: It is the bundle manifest specifying the bundle information along with the fragment host. It also contains package imports (if any).
- applicationContext-\${artifactId}-fragment.xml: Spring Context file whose beans will be loaded by the fragment host, Any spring beans should go here.
- \${artifactId}-tiles-defs.xml: An xml configuration file which contains tiles definitions for custom views which is loaded by fragment-host by file name.
- example.jsp: An example jsp file which specifies which JSTL tags can be used in the JSP and also provides an example to include a css or a javascript file.
- example.js: An example js file.
- example.css: An example css file.
- example.png: An example image file.
- pom.xml: The maven project file for dependency and build management. (Please add a maven dependency for the connector project for which this fragment will provide views).

General Guidelines:

Connector Fragment

- Use unique filenames and tiles definitions to avoid conflicts with other connector fragments.
- Use simple version numbers for bundle version to avoid bundle installation problems in Virgo server. Use x.x.x style of versioning, for example 1.0.0.

Contributing Custom UI for Configuring a Subscription for Provisioning

When a resource type includes custom properties, CloudPortal Business Manager will automatically generate a UI to collect values for these custom fields. However, because of its generic nature, this UI cannot be anything more than simple text fields. In order to support richer configuration, CloudPortal Business Manager allows connectors to define custom editors for these custom fields.

These editors will be embedded into CloudPortal Business Manager (they are not external windows or IFrames).

Each property defined in `Service/ResourceMetadata/ResourceType/Properties` is collected from the user at the time of provisioning. This default UI can be overridden by specifying a `Service/ResourceMetadata/ResourceType Editor` attribute which will be a reference to a jsp which will replace the default editor. The relative path of jsp is given as a tile definition defined in the fragment of the connector and this tile name is given in the service definition of the connector. This editor can be as simple or as sophisticated, depending on how the implementer wants it to be.

It can also interact with the back-end cloud service to provide a more interactive experience. The only requirement from CPBM is that it updates the set of html input fields with their names matching those of the property names specified in the metadata (and trigger the change event afterwards which will indicate to CPBM to refresh bundles based on selection as well as the selection summary shown in CPBM generic UI). CPBM will also extract data from this set of input fields and add them to the subscription configuration for provisioning. .

```
<Service name="CloudStack">
<ResourceMetadata>

<ResourceType name="VirtualMachine" constraint="NONE" componentSelector="apachecloudstack.fragment.r
<Properties editor="apachecloudstack.fragment.resource.virtual.machine.properties.selection" />
<Property id="hostName" type="String" validation="{\"required\":false}"/>
<Property id="displayName" type="String" validation="{\"required\":false}"/>

</Properties>

</ResourceType>
...
<ResourceType name="Volume" constraint="NONE">

<Properties>

<Property id="size" type="Number" validation="{\"required\":true}"/>
<Property id="tag" type="String" validation="{\"required\":false}"/>

</Properties>
```

</ResourceType>

</ResourceMetaData>

</Service>

Introduction

- Provisioning Resource is a 2 step process (with both steps with their own views, namely 'component selector' and 'editor'):
 - First step allows the user to filter bundles based on selection of resource components and pick one.
 - In the second step (view) (appears after the user has picked a bundle from step 1), the user gets to configure properties of the resource.
- CloudPortal Business Manager provides a default UI to the user for both steps, based on the resource types, filters and resource components defined in the service definition.
- Custom UI can be provided by the connector for any or both steps. The UIs provided will override the respective default ones.
 - Resource Component Selection (Step 1):
 - width: 650px
 - height: auto (the CloudPortal Business Manager UI will adapt depending on the height of the custom UI)
 - Resource Properties Selection (Step 2):
 - width: 650px
 - height: auto (the CloudPortal Business Manager UI will adapt depending on the height of the custom UI)

Fields made available by CloudPortal Business Manager

- tenantParam : The uuid of the tenant to which current user belongs
- serviceInstanceUuid : The uuid of the currently selected service instance
- Following parameters are only available in case the user has opted to reconfigure or reprovision an existing subscription:
 - subscriptionId : This parameter is available
 - isReconfigure: If available, means that the user has chosen reconfigure and not reprovision
 - subscriptionConfJson : The json representation of the configuration data of the given subscription (Used for prefilling the selectors and properties for custom UI Functions expected by CloudPortal Business Manager for initialization)
- **For resource selection custom UI:**

```
// loadValuesForSelectedBundleOnly : It tell whether to load all resource components available or ask C
// currentPage : This tells the current page out of the 2 pages mentioned in the introduction for resourc
// selectedResourceComponentsForStep2 : In step1, when group is ambiguous based on the resource com
void initializeComponentSelector(boolean loadValuesForSelectedBundleOnly, int currentPage, String[] se
// initialize resource components
}
```

- **For editor custom UI:**

```
1void initializeEditor() {
  // initialize resource type properties editor
}
```

Functions provided by CloudPortal Business Manager

```
/**
 * If the user is on second page, that means, a bundle is selected. So, custom resource selection UI does not
 * This method should be used instead to get only allowed list of values for the given component name (as pr
 **/
List<ResourceComponent> getValuesForComponent(String componentName);
```

```
/**
 * enable tells whether to enable or disable the provision button on page 2. This can be disabled by custom U
 * 'msg' is the info or warning message to be shown when the provision button is disabled
 **/
void enableProvisionButton(boolean enable, String msg);
```

Working with hidden input fields

CloudPortal Business Manager needs a way to be aware of the selections made by user inside the custom UI (to generate selection summary, check bundle compatibility, and so on. If the user has opted to re-provision or reconfigure an existing subscription, CloudPortal Business Manager needs to provide the selections made while creating subscription to the custom UI (for custom UI to pre-fill fields based on previous selections).

For solving both the above purposes, CloudPortal Business Manager uses hidden input fields. The whole UI for resource provision (resource selection as well as properties editor) is driven by these hidden input fields in the background. For each of the following (defined in service definition), CloudPortal Business Manager auto generates a hidden input field:

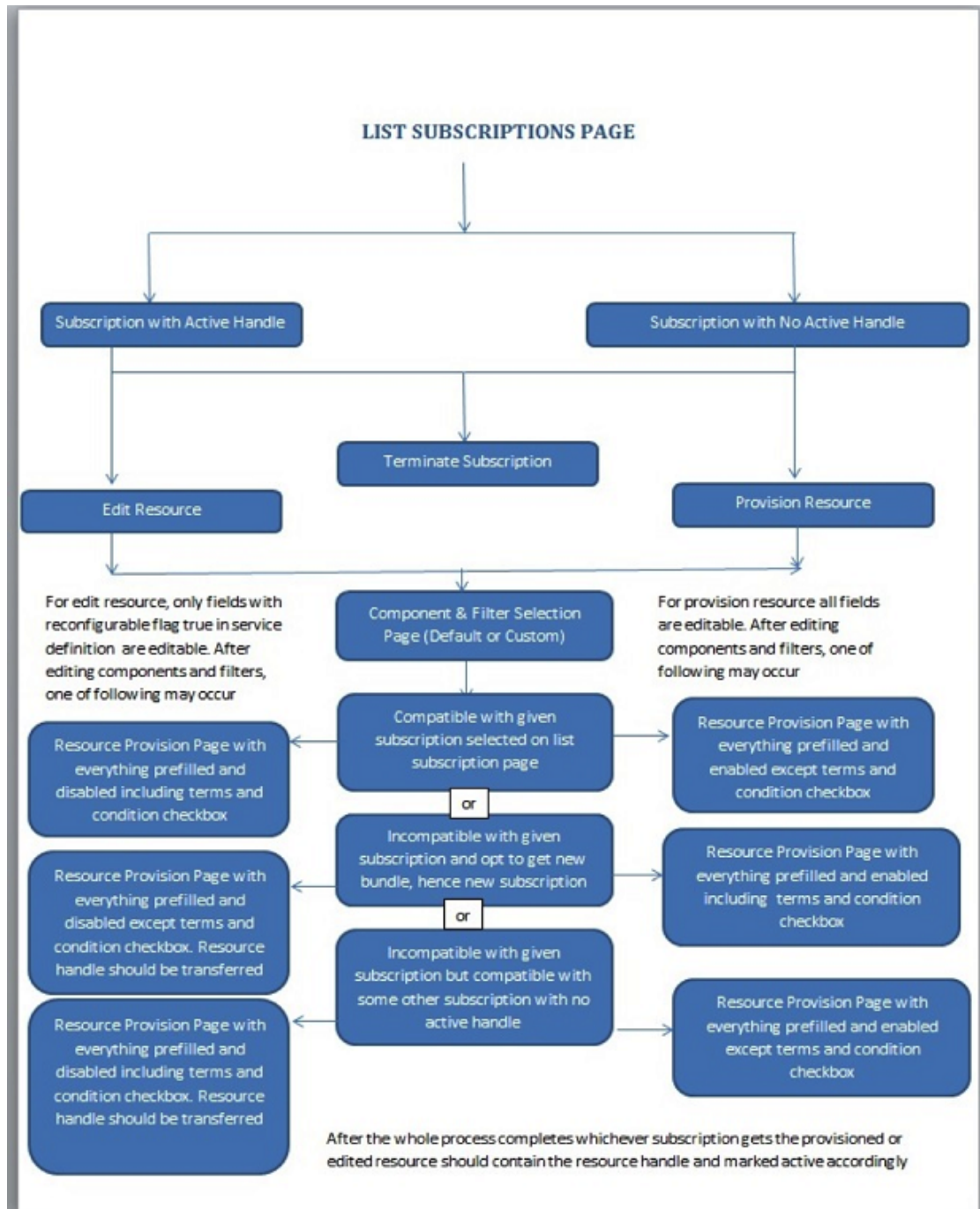
- Filters
- Resource Components
- Resource Properties

The values of the hidden fields are preset in case the user has opted for reconfiguring or reprovisioning an existing subscription. Other specific details about the hidden input DOM element are as follows:

Property	Description	Set by
attr("id")	The id of the hidden input is set by CloudPortal Business Manager as "conf_prop_" + name of the filters, components and properties as given in the service definition.	CloudPortal Business Manager
attr("name")	The name of the hidden input is set by CloudPortal Business Manager same as the name of the filters, components and properties as given in the service definition.	CloudPortal Business Manager
val()	The value of the filter, resource component or property that will finally be used for the resource provision.	CloudPortal Business Manager (for reprovision and reconfigure) and Custom UI
data("valueDisplayName")	The display name for the selected value for filter, component or property to be displayed in the selection summary.	Custom UI
data("fieldDisplayName")	The title for the selected filter, component or property to be displayed in the selection summary	Custom UI
data("effectiveValue")	Required for resource components only, this value signifies if charging of a component is based on a different value than the one used for resource provisioning. For all other resource components this value is to be set same as the value of the input field.	Custom UI

Irrespective of the fields and options in the custom UI, custom UI needs to call the `.change()` method on the hidden input field after updating the above mentioned properties of the input field. This is applicable for resource components as well as for the resource properties.

Use Cases



Upgrading custom UI written for CloudPortal Business Manager 2.0 release

1. Create a new connector fragment by following the steps given in the [Connector Fragment](#) section.
2. Move the jsp into fragment in src/main/resources/WEB-INF/jsp/\${artifactId}/ folder.
3. Spring messaging can be used after including appropriate JSTL tags which are specified in example.jsp in the fragment.
4. Script from inside the custom UI can be moved out to another file in the fragment in src/main/resources/js/\${artifactId}/ folder.

5. The jsp needs to be declared in `src/main/resources/WEB-INF/default/${artifactId}-tiles-defs.xml` of the fragment.
6. Mention the tile definition name in the service definition of connector instead the jsp path.
7. Custom UIs no more need to declare the input fields themselves as they are now auto generated by CloudPortal Business Manager.

Earlier for properties, CloudPortal Business Manager read input fields with "pp_" appended to property name in the 'id' of the field. Now it uses the name of the field and expects it to be same as the property name mentioned in the service definition file. The ids of the fields are "conf_prop_" + name of field in service definition. This is applicable for filters, resource components as well as resource properties.

Citrix® CloudPlatform™ Connector Guide

The Citrix CloudPlatform Connector is installed in CloudPortal Business Manager during installation of CloudPortal Business Manager. This guide describes the steps necessary to enable the cloud service, add instances of the services, typical product and bundle definitions, usage types, and resources available through the CloudPortal Business Manager UI. The guide also explains how resources can be managed from the CloudPortal Business Manager UI.

Supported CCP versions are: 3.0.5, 3.0.6, 3.0.7, 4.2.0

Installing CCP Connector
Sending CCP APIs
Onboarding and Managing the CCP service
CCP Connector User Roles
Mapping of CPBM and CCP entities
CCP Product Onboarding
CCP Connector Resource Provisioning
Managing Resources
Setting Service Instance Specific User Settings
FAQs

Installing Citrix CloudPlatform Connector

The CCP connector is installed, by default, when you install CloudPortal Business Manager. The CCP connector is deployed into VIRGO's pickup directory. CloudPortal Business Manager extracts the information from the connector and copies the serviceDefinition.xml, logo, terms and condition files and UI implementation files to the /usr/share/cloud/portal/data directory.

Sending Citrix CloudPlatform APIs

The API access to Citrix CloudPlatform (CCP) managed by CloudPortal Business Manager (CPBM) is enabled by proxying API request through the CPBM server. This section provides the syntax for sending API calls to cloud endpoints.

The API endpoint to access CCP APIs through the API proxy is as follows:

`http://<Hostname of the CPBM server>:<port>/portal/client/apis/<API Proxy suffix name>/?command=<CCP`

Refer to [Adding an instance to the CCP service](#) for information on the API Proxy suffix name parameter.

All the parameters specified to the API proxy are forwarded to the specific CCP server.

For more information on each API parameter, see CloudPlatform Developer's guide and API Reference.

Onboarding and Managing the Citrix CloudPlatform service

Onboarding a service refers to the process of:

- Deploying a CCP connector into CloudPortal Business Manager.
- Enabling a CCP connector and adding a CCP service instance.

CCP Connector

The CCP connector is a Java based OSGi bundle which:

- Implements the CloudPortal Business Manager Cloud Service Interfaces
- Defines the following metadata:
 - Endpoint and credential parameters
 - Resource Metadata: Various resources that can be provisioned and metered
 - Custom UI Views for Subscription provisioning
 - Service Controls

The CCP connector implements set of hooks and CloudPortal Business Manager invokes these hooks at appropriate stages. The stages include:

1. Account Creation
2. User Creation
3. Resource provisioning
4. Product definition
5. Usage collection
6. Event collection

Multiple instances of Citrix CloudPlatform (CCP) can be added in CloudPortal Business Manager. You have to first enable the CCP connector. After enabling, roles defined by the CCP service are mapped to appropriate CloudPortal Business Manager profiles.

Note: The service onboarding for customer and customer's users depends on these role mapping. Ensure that roles are appropriately mapped to profiles.

Preparing Citrix CloudPlatform to work with CloudPortal Business Manager

Pre-requisite: Citrix CloudPlatform (CCP) is installed and fully operational.

1. Log in to CloudPlatform Management Server as Admin user.
2. Navigate to Accounts > Admin.
3. Select View users > Admin.
4. Click Generate Keys. This generates the API key and secret key.
5. Copy these keys to a text file.
6. Navigate to Global Settings:
 - a. Search for security.singlesignon.key value. Copy it to a text file.
 - b. Search for integration.api.port.
 - i. Specify the default api port value as 8096.
 - ii. Click Save and then, click OK.
7. Restart the CloudStack service:
 - a. SSH to the Citrix CloudPlatform Linux server.
 - b. Log in and type the command: `service cloudstack-management restart`.
8. Refresh the CCP UI page and log in as Admin user.
9. Install the Cloud Usage server on CCP. After the installation is complete start the cloud-usage server by typing this command: `service cloudstack-usage restart`.
10. Configure CloudPlatform database to support CloudPortal Business Manager. Create a new MySQL user and configure the proper permissions for this user, as this will allow CloudPortal Business Manager to access the CCP database.
 - a. Login to MySQL using the command:
`mysql -u root (use -p if you defined password for the MySQL root user).`
 - b. At the mysql prompt,

`create user cpbm identified by 'ANY PASSWORD';
GRANT ALL PRIVILEGES ON *.* TO 'cpbm'@%' WITH GRANT OPTION;
FLUSH PRIVILEGES;`
 - c. Exit MySQL by running to command: `quit`.

11. Configuring iptables on CCP: Allow incoming TCP ports 8096, 20400 and 3306 on the CCP Server by editing the iptables file.

a. Run the command:

```
vi /etc/sysconfig/iptables
```

b. Add these lines:

```
-A INPUT -p tcp -m tcp --dport 3306 -j ACCEPT  
-A INPUT -p tcp -m tcp --dport 8096 -j ACCEPT  
-A INPUT -p tcp -m tcp --dport 20400 -j ACCEPT
```

c. Save and exit the file and restart the iptables service using the command: `service iptables restart` .

Integrating CloudPortal Business Manager with Citrix CloudPlatform for Single Sign-on

As an Admin user, if you do not want to provide username and password while launching the Citrix CloudPlatform (CCP) UI Console from CloudPortal Business Manager, you have to provide Single Sign-on (SSO) integration between CloudPortal Business Manager and CCP.

On each CloudPlatform server, run the `./install.sh` script and select Integrate with CloudPlatform from the menu. This will place files within CloudPlatform that are needed for single sign-on integration with CloudPortal. Note that after the files are copied over for integration, you will no longer be able to access the CloudPlatform management server using `http://<hostname>:8080/client`. It is recommended to access the CloudPlatform management UI by first logging into CloudPortal and clicking the "Launch Cloud Console" button on the dashboard. As an alternative, the CloudPlatform management UI can be accessed directly by using `http://<hostname>:8080/client/?direct=true`.

```
# ./install.sh
Welcome to the Citrix CloudPortal Business Manager Installer.
What would you like to do?
D) Install the database server
I) Install & Configure CloudPortal Business Manager
C) Integrate with Citrix CloudPlatform
Q) Quit
> C
```

Choose option C and press Enter and wait for the process to complete. Please note that no confirmation will be displayed. Click q to exit the installer.

Enabling the Citrix CloudPlatform Connector

1. Log in to CloudPortal Business Manager UI as Root user.
2. Go to Home > All Services.
3. Select the Citrix CloudPlatform connector and click Enable.
4. Accept the Terms & Conditions defined by the CCP cloud service and click Accept.
5. Map the roles defined by the CCP cloud service to the appropriate CloudPortal Business Manager profiles.

Note: The service onboarding for customer and customer's users depends on these role mapping. Ensure that roles are appropriately mapped to profiles.

6. Click **Submit** and then click **Close**.

Adding an instance to the CCP service

The Add Service Instance wizard enables you to add an instance of the service. You can specify the parameters required to connect to the service, add products, and specify the utility prices for the products.

Pre-requisites:

- Citrix CloudPlatform IP
- Citrix CloudPlatform Database credentials
- Admin API key
- Admin Secret Key
- Single.Signon key
- Ensure the 8096 port is enabled and IPtables updated on Citrix CloudPlatform

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Home > All Services > Cloud Services. You will find the CCP service of the category IaaS.
3. Click **Add**. The Add Service Instance wizard opens.
4. Specify the identifying information for the instance such as name, code, and description and click Next.
5. In the Configuration page, specify the details:
 - Protocol (http or https): http
 - Public hostname: Public IP of Citrix CloudPlatform
 - Public Port: 8080
 - Cloud JDBC hostname: IP of the MySQL server where Citrix CloudPlatform is hosted
 - Cloud JDBC username: Citrix CloudPlatform DB user
 - Cloud JDBC password: Citrix CloudPlatform DB user password
 - SSO key: Citrix CloudPlatform single.signon.key value (which is used by root user for single sign on into Citrix CloudPlatform)
 - API key: Administrator's user API key
 - Secret key: Administrator's user Secret key

- CloudPlatform domain Uuid: ID of the CloudPlatform ROOT domain or any other domain you wish to use to create accounts and users
- CloudPlatform IP List (administration port): Citrix CloudPlatform IP:8096 (for example. 192.168.1.17:8096)
- CloudPlatform IP List: Citrix CloudPlatform IP:8080 (for example. 192.168.1.17:8080)
- API Proxy suffix: csapi (or any other suffix you wish to use)

Click **Advanced Settings** (optional) and specify the details:

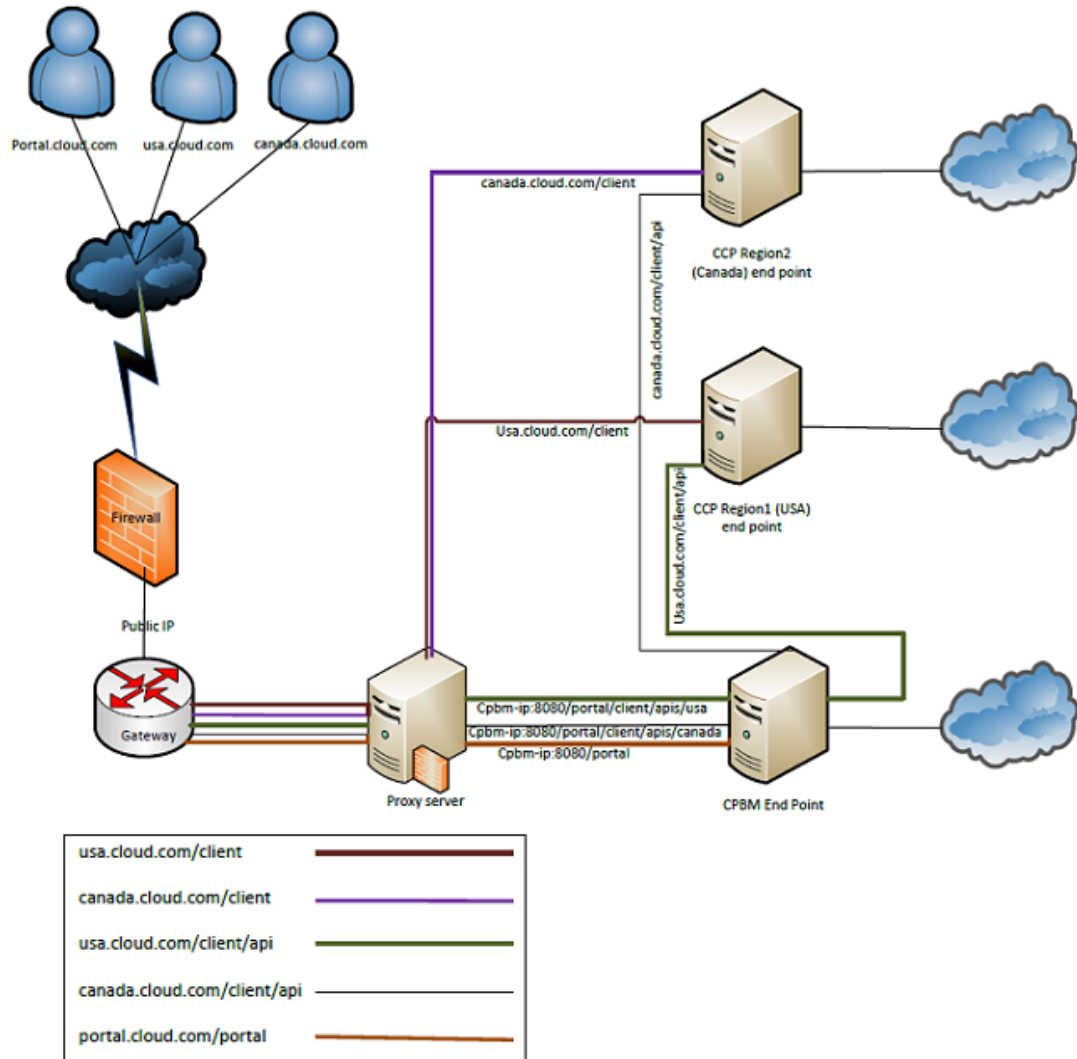
- Cloud database schema name:
- Cloud Usage JDBC Hostname: Enter the cloud JDBC hostname or IP.
- Cloud Usage JDBC username: Enter the cloud JDBC username
- Cloud Usage database schema name: Enter the cloud database schema name
- API Whitelist Suffix: Enter a comma separated white-list of API names.
- API Blacklist Suffix Enter a comma separated black-list of API names.
- Default keyboard device type for virtual machines: Valid value can be any one type from this list - de, de-ch, es, fi, fr, fr-be, fr-ch, is, it, jp, nl-be, no, pt, uk, us
- Custom disk offering maximum size in GB.

Note: If no value for "Custom disk offering max size in GB" is entered, a default value of 1024 GB will be set. Hence no disk offerings greater than 1024 GB will be displayed in the Start VM Wizard.

6. Click Next. The Default Product Selections page is displayed. This is a list of products based on defined UsageTypes. For more information on usage types, refer to the section on [Usage Types](#).
7. Select the products that you want to create at the time of creation of the service instance. You can also edit the product name, product code, category, and required product units for each of the default products.
8. Click Next and specify the utility charges (in all active currencies) for the products that you have selected.
9. Review and confirm the details.
10. Click Add and then, click Close.

Region support configuration

For supporting multiple CCP regions, the following is a sample configuration:



In the figure, portal.cloud.com, usa.cloud.com and canada.cloud.com map to the same proxy server which has the corresponding routing configuration. cpbm-ip is the IP of the LB to the CloudPortal Business Manager nodes, usa-ccp-ip is IP of the LB to USA region nodes, and canada-ccp-ip is the IP of the LB to Canada region nodes.

Here is a sample Apache HTTP proxy configuration for the proxy server to achieve the same:

```
NameVirtualHost *:80
```

```
<VirtualHost *:80>
  ServerName portal.cloud.com
  ProxyPass /portal http://cpbm-ip/portal
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerName usa.cloud.com
  ProxyPass /client http://usa-ccp-ip/client
  ProxyPass /client/api http://cpbm-ip/portal/client/apis/usa
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerName canada.cloud.com
  ProxyPass /client http://canada-ccp-ip/client
  ProxyPass /client/api http://cpbm-ip/portal/client/apis/canada
</VirtualHost>
```

Note: Here we have assumed that this proxy server is not load balancing the CPBM and the CCP nodes. For load balancing CPBM and CCP servers, refer to the **Setting up a Proxy Server** section of [Installing and configuring CloudPortal Business Manager](#).

Service Controls

The CCP connector supports the following set of service controls at the tenant level:

Parameter	Parameter Definition
Resource Sharing Mode	If the value is set to true, all resources that are provisioned for the account are shared by all users of the account. If the value is set to false, the resources are not shared and separate account is created for each user on the service side.
Per Account Public IP Count	The default total number of public IP addresses allowed for the account type.
Per Account Snapshot Count	The default total number of snapshots allowed for the account type.
Per Account Template Count	The default total number of templates allowed for the account type.
Per Account VM Count	The default total number of VMs allowed for the account type.
Per Account Volume Count	The default total number of disks allowed for the account type.
Per User Public IP Count	The default total number of public IP addresses allowed per user within this account type.
Per User Snapshot Count	The default total number of snapshots allowed per user within this account type.
Per User Template Count	The default total number of templates allowed per user within this account type.
Per User VM Count	The default total number of VMs allowed per user within this account type.
Per User Volume Count	The default total number of disks allowed per user within this account type.
Default Network Offering	Default network offering, which is a set of network services such as DHCP, DNS, port forwarding, firewall, VPN, and so on.

The limits are set for each account type.

You can over-ride these limits for a particular account from the Accounts tab.

Note: An account limit on service control cannot be reduced from the existing limit. For example, if an account type (corporate) has Per Account VM Count set to 10, you cannot reduce the limit to a number less than 10.

Editing a CCP instance

1. Log in to CloudPortal Business Manager UI as Root user.
2. Click Home > All Services > Cloud Services. You will find the CCP service of the category IaaS.
3. Click the gear wheel icon next to the instance and select Edit Configuration.
4. Edit the respective fields and click on Save.

CCP Connector user roles

After enabling the CCP service, roles defined by the CCP service are mapped to the appropriate CloudPortal Business Manager profiles. The CCP connector supports three kinds of roles.

- **CloudPlatform Administrator:** This role is available to CloudPortal Business Manager users in the Service Provider scope. This role has global admin privileges in CloudPortal Business Manager. This is enabled by default for the Service Provider administrator. This role has the ability to configure and manage the cloud infrastructure and manage all resources through CloudPlatform console.
- **CloudPlatform Domain Administrator:** This role is available to CloudPortal Business Manager users in the Customer scope. This role will have customer admin privileges in CloudPortal Business Manager. This role is enabled by default for the Master User profile. This role has the ability to create, read, update, and delete resources in a tenant.
- **CloudPlatform User:** This role is available to CloudPortal Business Manager users in the Customer scope. This role will have normal user privileges in CloudPortal Business Manager. This is the default role for CloudPlatform user.

Mapping of CPBM and CCP entities

CPBM entities:

- Channel: A channel usually represents a market segment or a geographic area (depending on the segmentation done by the service provider).
- Account: An account typically represents a customer of the service provider and is the basic unit of isolation. Accounts are grouped by channels (every account is tied to a specific channel). Accounts can be of different types, the characteristics of each type are decided by the service provider.
- Master user: Each account has a master user who can perform administrative operations for users who belong to that account.
- User: Each account has one or more users.

CCP entities:

- Domain: Domains usually contain multiple accounts that have some logical relationship to each other. Domain can have arbitrary levels of sub-domains.
- Account: An account typically represents a customer of the service provider. Accounts are grouped by domains.
- For each account created, the Cloud installation creates three different types of user accounts: root administrator, domain administrator, and user.
 - User: Users are like aliases in the account.
 - Domain Administrators: Domain administrators can perform administrative operations for users who belong to that domain.
 - Root Administrator: Root administrators have complete access to the system, including managing templates, service offerings, customer care administrators, and domains.

Mapping of CPBM and CCP entities:

The mapping between CPBM and CCP entities is decided by the resource sharing mode. The resource sharing mode is a setting that is specific to an account type and is set to either True or False by the service provider. When set to true, the shared account mode is enabled. This property is configured for an account type for a specific CCP instance by the administrator by clicking **Administration > Account Types > Selected Account Type > Service Controls > CCP Service Instance > Edit**. While the initial setting for an account depends on account type, resource sharing mode can be changed for specific accounts by editing the account.

Mapping between CPBM and CCP when resource sharing mode is set to True:

CloudPortal Business Manager entity maps to	Citrix CloudPlatform entity
Channel	Domain

Mapping of CPBM and CCP entities

Customer account	Sub Domain
Master User <ul style="list-style-type: none"> Manages the customer account Can view VMs of all users in the customer account Can create new users in the customer account 	Domain admin account
Users <ul style="list-style-type: none"> All resources of the customer account are accessible Cannot create new users in the customer account 	Users in the master user Domain admin account

When the resource sharing mode is set to true:

- For each account in CPBM, a sub domain is created in CCP - where every user is the Domain Admin.
- Every user's usage gets accumulated at the tenant level.
- All the entitlements across all users are scoped to tenant level.
- Normal User can see resources launched by Master user as well, on resources tab.
- Unlimited entitlements are always scoped to subscription level.

Mapping between CPBM and CCP when resource sharing mode is set to False:

CloudPortal Business Manager entity maps to	Citrix CloudPlatform entity
Channel	Domain
Customer account	Sub Domain
Users <ul style="list-style-type: none"> Each user gets his own VLAN network, public IP and Virtual router Normal users can access their VMs only 	Account
Master User <ul style="list-style-type: none"> Manages the customer account Can view VMs of all users in the customer account Can create new users in the customer account 	Domain admin account

When Resource Sharing mode is set to false:

Mapping of CPBM and CCP entities

- For each tenant in CPBM, a sub domain is created in CCP, where master user is the Domain Admin.
- For each user, a user account is created in the sub domain of CCP.
- Entitlements and usages are scoped to user level. Hence, each user gets an invoice for his usage charges.
- Normal User can view only the VMs launched by him.
- Unlimited entitlements are always scoped to subscription level.

Product Onboarding

Products can be created in two ways:

- Basic products for CCP usage types are created, at the time of creation of service instance, from the Default Product Selections page of the Add Service Instance wizard. You can specify the name, code, category, and unit of the product.
- Advanced products are created from the Catalog tab, when you plan the next version. The Add Product wizard is used to create advanced products.

CCP Usage Types

The connector collects usage data for the following usage types so that they are billable in CloudPortal Business Manager. Each Usage Type's generated usage unit can be converted to other predefined scales in CloudPortal Business Manager which is provided by the CCP connector. Each Usage Type can have multiple discriminators, and these are attributes which are used to uniquely identify a usage data of a Usage Type.

Usage Type	Generates usage in	Description	Predefined Scales	Discriminators
Running VM	Compute-Hours	Running VM	Compute-Days	HA Enabled Guest OS ISO Image
Allocated VM	Compute-Hours	Allocated VM - the VM can be in any state	Compute-Days	HA Enabled Guest OS ISO Image
Root Volume	Byte-Hours	Root volume in primary storage	GB-Months TB-Months GB-Days TB-Days	Pool Type Volume Tag
Data Volume	Byte-Hours	Non-root volume (i.e. data volume) in primary storage	GB-Months TB-Months GB-Days TB-Days	Pool Type Volume Tag
Template Storage	Byte-Hours	Storage consumed by user created templates in secondary storage	GB-Months TB-Months GB-Days TB-Days	
ISO Storage	Byte-Hours	Storage consumed by user created ISOs in secondary storage	GB-Months TB-Months GB-Days TB-Days	

CCP Usage Types

Snapshot Storage	Byte-Hours	Storage consumed by user created snapshots in secondary storage	GB-Months TB-Months GB-Days TB-Days	
IP Address	IP-Hours	Acquired public IP address	IP-Days IP-Months	
Network Bytes Received	Bytes	Network bytes received from internet	GB, TB	Redundant Router Service Enabled VLAN Enabled Network Type
Network Bytes Sent	Bytes	Network bytes sent to internet	GB, TB	Redundant Router Service Enabled VLAN Enabled Network Type
Security Group	SecurityGroup-Hours	Security group	SecurityGroup-Days SecurityGroup-Months	
Load Balancer Policy	Policy-Hours	Load balancer policy	Policy-Days Policy-Months	
Port Forwarding Rule	Rule-Hours	Port forwarding rule	Rule-Days Rule-Months	
VPN Users	Vpn-Hours	VPN users	Vpn-Days Vpn-Months	

CCP Resource Types

CCP has three Resource Types which can be provisioned, namely **Compute**, **Storage**, and **Networking**. Each Resource Type can generate multiple Usage Types.

Each Resource Type can also specify multiple groups, which specify the multiple ways that the Resource Type can be provisioned. Each group can contain multiple components which are the requirements of that group which needs to be satisfied at the time of provisioning.

The Usage Types and Groups of Resource Types are as follows:

Resource Type	Generated Usage Types	Resource can be provisioned using
Compute	<ul style="list-style-type: none">• Running VM• Allocated VM• Root Volume	<ul style="list-style-type: none">• Template and Compute Offering• ISO, Hypervisor and Compute Offering
Storage	<ul style="list-style-type: none">• Data Volume	<ul style="list-style-type: none">• Disk Offering
Networking	<ul style="list-style-type: none">• Network Bytes Received• Network Bytes Sent	<ul style="list-style-type: none">• VPC and VPC Network Offering• Non VPC Network Offering

While creating a product bundle for a Resource Type, any usage that can be attributed to that Resource Type can be given as **unlimited** entitlement in that bundle.

Entitlements are accounted either to usage accrued to a given subscription or in a consolidated manner at the user level. Whether usage is accrued to a subscription or to the user is dependent on the specific product being entitled and the specific type of the bundle being subscribed to. For example, VM running time, can be accrued to a virtual machine, but data transfer is usually not attributable to the virtual machine as it shares the network on which the traffic is being generated and accounting for data transfer is done at the router. So, when purchasing a virtual machine, data transfer entitlements in the associated bundle will be added to the user's overall entitlements for data traffic, whereas when purchasing a network, the data transfer on that network will be tied specifically to that subscription.

In a **Compute** bundle, any products based on a combination of Running VM/Allocated VM usage type or only Root Volume Usage type can be given as unlimited entitlement.

In a **Storage** bundle, any products based on Data Volume usage type only can be given as unlimited entitlement.

For **Networking** bundle, any products based on combination of Network Bytes Received/Network Bytes Sent usage type can be given as unlimited entitlement.

Sample CCP Products

You can create basic products from the **Add Service Instance** wizard.

Examples:

- To create a **Stopped VM** product, **combine** Allocated VM and **exclude** Running VM.
- To create a **Network Bandwidth** product, **combine** Network Bytes Sent and **combine** Network Bytes Received.
- To create a **ROOT VOLUME** product, use Root Volume usage type.

The following are some typical products defined:

Product	Unit	Description	Price
Small Instance	per Hour	Small running VM (1 CPU@500MHz, 512MB RAM)	Specified in active currencies
Network Bytes Received	per GB	Data transfer from Internet	Specified in active currencies
IP Address	per Month	Public IP address	Specified in active currencies
Root Volume	Per GB per Month	Storage used by root volume of the VM	Specified in active currencies
CentOS 5.6 Template	per Hour	CentOS 5.6 template used by running VMs	Specified in active currencies

Product configuration for a sample product "Small Instance"

Small Instance		
Product Description:		
Name	Small Instance	
Code	CloudPlatformCloud_SMALL_INSTANCE	
Category	Infrastructure: Compute	
Product Details:		
1. Usage Type	Generates Usage in Units	Aggregation Handling
Running VM	Compute-Hours	Combine
2. Mediation Rules		
Compute Offering	INCLUDES	Small Instance
3. Conversion Factor		

Sample CCP Products

Generates Usage in Units	Required Product Units	Conversion Factor
Compute-Hours	Compute-Hours	1
Product Charges		
EUR (€)	US (\$)	
0.0300	0.0400	

Sample CCP Product Bundles

To create a **Compute bundle** (which can be used to launch only windows template), Choose **Template** provisioning constraint and include all **Windows** templates.

Note: To a compute bundle, only products created using "Running VM", "Allocated VM", and "Root Volume" usage type can be assigned unlimited entitlements.

The following are some typical product bundles defined:

Product Bundle	Description	One-time charge	Recurring charge, Unit
Small CentOS Instance Bundle	Includes small instance, CentOS OS, 20GB root volume	Specified in active currencies	Specified in active currencies, Month
20GB Data Volume	Includes 20GB data volume	Specified in active currencies	Specified in active currencies, Month
Silver Network Package	Includes 10GB of network bytes sent and 10GB of network bytes received	Specified in active currencies	Specified in active currencies, Month
Standard Support	Include a standard support package for the customer	nil	Specified in active currencies, Month

Product bundle configuration for a sample product bundle "Small CentOS Instance Bundle"

Small CentOS Instance		
Bundle Description:		
Name	Small CentOS Instance	
Code	CloudPlatformCloud_SmallCentOSInstance	
Description	This is a small CentOS instance (1 CPU@500MHz, 512MB RAM) charged at monthly rate.	
Bundle Details:		
Resource Type:	Compute	
Allow Subscriptions:	At any level; no limits on number of subscriptions	
Charge Frequency:	Monthly	
Eligible for Trial:	Yes	
Enable Notification:	Yes	

Sample CCP Product Bundles

Provisioning Constraints:		
Template	INCLUDES	CentOS 5.6 (64-bit) no GUI (XenServer)
Compute Offering	INCLUDES	Small Instance
Bundle Charges:		
Charges	EUR (€)	US (\$)
One-time	4.0000	5.0000
Recurring	20.0000	26.0000
Entitlements:		
Small Instance	Unlimited Compute-Hours	
CentOS 5.6 Template	Unlimited Compute-Hours	
Root Volume	20 GB-Months	

CCP Resource Provisioning

The CCP connector has three Resource Types, which are **Compute**, **Storage** and **Networking**. These resources can be provisioned by subscribing to the corresponding bundle.

A Resource Type can be provisioned by satisfying one of the groups.

A **Compute** resource type can be provisioned by either specifying a **Template** and a **Compute Offering** or by specifying an **ISO**, **Hypervisor**, and a **Compute Offering**.

A **Storage** resource type can be provisioned by specifying a **Disk Offering**.

A **Networking** resource type can be provisioned by either specifying a **VPC** and a **VPC Network Offering** or by specifying a **Non VPC Network Offering**.

Compute and Networking Resource Types can also be provisioned using the manage resource UI, which are charged as a utility subscription in CloudPortal Business Manager.

To Provision a Compute Resource Type using a Template and Compute Offering:

1. Log in to CloudPortal Business Manager UI as a normal user.
2. Click Catalog.
3. Select the **Compute** resource type from the selection box and use the Compute Offering drop down list to choose the desired **Compute Offering**.
4. Use the Template/ISO drop down list to filter/select the desired **Template**.

The product bundles will be filtered based on the selection.

5. Choose the desired product bundle and click **Configure and Subscribe Using Compute Offering, Template**.
6. Fill in the details (optional): For the compute resource, select a Disk offering(optional), a network using which the compute resource will be provisioned and a SSH key (optional).
7. After providing all the required information, accept the terms and conditions by checking in the corresponding checkbox and click on subscribe and provision button to provision the compute resource.

Async Job

The Citrix CloudPlatform Connector Async job, is an asynchronous scheduled job which polls on CloudPlatform's async jobs list for resource provisioning and reset password requests and sends out different emails in case of their success or failure.

In case of a resource provisioning, if it is a success then it sends out an email to the user that the resource has been provisioned successfully and it also sends out the password in case of compute resource if it is provisioned using a password enabled template. If the provisioning fails, then it sends out an email reporting the same.

If there is a reset password for VM request, then it sends out the new password to the requestor and to the owner of the VM, if they are not same.

Managing resources

This section describes the CCP resources available through CloudPortal Business Manager. It also gives you a list of activities that can be performed in the CloudPortal Business Manager UI. The CPBM UI also displays the status of the resource which maps to the state of the CCP resource (each CCP resource has its states).

Instances
Volumes
Snapshots
IP Addresses
Networks
Templates
ISOs

Instances

An instance is another term used for a virtual machine. Virtual machines are software implementations of physical computers, and can execute computer programs. You can power on, power off, or suspend a virtual machine just like you would a physical computer.

VM Details tab

Display Name: A unique name given to your VM by your service provider.

Group: The group to which the VM belongs. This is set when the VM is created.

Zone: A unique zone in which your VM is placed.

Template: The template from which your VM was created.

Compute Details: Details about the compute resources. CPU, RAM, and so on.

Account: The account this VM was created in.

Domain: The domain in which this VM resides.

Created: The date on which this VM was created.

HA enabled: HA stand for High Availability. If this is enabled for the selected VM, the downtime for the VM is minimal.

Attached ISO: If the value is Yes, it means that the VM container has an attached ISO.

Filter Instances

The Filter menu shows a list by which you can order your VMs:

- **All Instances:** Select this menu option to view all your VMs regardless of their state.
- **My Instances:** VMs that you created.
- **Running Instances:** The VMs that are currently running (powered on).
- **Stopped Instances:** The VMs that are currently stopped (powered off, but available to be powered on again).
- **Destroyed Instances:** The VMs that are destroyed (no longer available).

Tasks performed

Move your cursor over the gear wheel icon. You see a list of actions that you can perform for the selected VM.

Edit VM: Click this option and change the Display Name. You can also change the Group to which this VM belongs.

Reboot VM: Click this option to reboot the selected VM.

Start VM: Click this option to start (power on) the selected VM. Only VMs in stopped state can be started.

Stop VM: Click this option to stop the selected VM. Only VMs that are running can be stopped.

Restore VM: Click this option to restore the VM to its original template.

Destroy VM: Click this option if you want to remove the VM completely. You cannot destroy a VM that is running.

Reset Password: Click this option to reset the password for this VM. The VM must be password enabled for you to be able to reset the password.

Attach/Detach ISO: An ISO image is an exact copy of an existing file system. You can attach an ISO to your VM with this option. Attached ISOs can be detached. The ISO that you select must be compatible with the VM hypervisor for the ISO to be attached successfully. Click the option to attach or detach an ISO.

Reset SSH Key: Click this option to reset the SSH Key for the selected VM.

Accessing the VM Console

Any user can access their own VMs and perform actions on the VMs.

1. Log in to CloudPortal Business Manager as user.
2. Click **My Services**.
3. Select the cloud service and the service instance.
4. Click **Instances** and select the VM that you want to access.
5. Click **View Console**.

Resource properties for provisioning VM

The properties used to provision a VM are described in detail in the table below. For provisioning VM, following resource properties are needed.

Resource Property Name	Description	Required/Optional
displayName	User generated name for the virtual machine	Optional
group	Group for the virtual machine	Optional
diskOfferingUuid	The ID of the disk offering for the virtual machine. If the template is of ISO format, the diskOfferingId is for the root disk volume. Else, this parameter is used to indicate the offering for the data disk volume. If the templateId parameter passed is from a Template object, the diskOfferingId refers to a DATA Disk Volume created. If the templateId parameter passed is from an ISO object, the diskOfferingId refers to a ROOT Disk Volume created.	Optional
keypair	Name of the ssh key pair used to login to the virtual machine	Optional
diskSize	The size for the DATADISK volume.	Optional
keyboard	Keyboard device type for the virtual machine. Valid value can be one of de,de-ch,es,fi,fr,fr-be,fr-ch,is,it,jp,nl-be,no,pt,uk,us	Optional
networkIds	List of network ids used by virtual machine. Cannot be specified with ipToNetworkList parameter	Optional
securitygroupIds	List of security groups id separated by comma that are going to be applied to the virtual machine. Should be passed only when vm is created from a zone with Basic Network support.	Optional
userData	Binary data that can be sent to the virtual machine upon a successful deployment. This binary data must be base64 encoded before adding it to the request. Currently only HTTP GET is supported. Using HTTP GET (via querystring), you can send up to 2KB of data after base64 encoding.	Optional
ipAddress	The ip address of default vm's network	Optional

Resource properties for provisioning VM

ipToNetworkList	ip to network mapping. Cannot be specified with networkIds parameter. Example: ipAddress1,networkid1;ipaddress2,networkid2;....	Optional
serviceOfferingUuid	The ID of the service offering for the virtual machine	Required
templateUuid	The template ID for the virtual machine	Required
isoUuid	The ID of the ISO attached to the virtual machine	Required
hypervisorType	The hypervisor on which to deploy the virtual machine	Required

Volumes

Just like a physical computer has storage, a virtual machine also has a storage volume associated with it. This is called the Root volume. You can choose to have additional storage as well. In the Volumes section you can view the details of your VM volumes, attach to, and detach volumes from your VMs, create volume templates and perform other tasks related to storage volumes.

Details Pane

A red circle next to a volume indicates that the volume is allocated, but not ready yet. A green circle indicates that the volume is ready to be attached, detached, downloaded and so on. as is applicable to that volume.

To view the details of a volume, select it from the list of volumes in the left pane.

ID: The unique identification code assigned by your service provider.

Type: The volume type. Root volume, Data volume

Zone: The CloudStack zone in which this volume resides.

Device ID: A unique system generated identifier for this volume.

Size: The storage space in GB. For example, 8 GB.

Created: The date and time the volume was created.

Account: The user who created this volume.

Domain: The domain assigned by your service provider.

Managing volumes

The following section describes the menu items of the gear wheel icon. The menu shows the actions that you can perform. The menu items change depending on the Type (Root, Data etc.) and State (attached, detached) of the volume. For example, you cannot attach a volume that is already attached to a VM. In such a case, only the Detach Volume is available for that volume.

Select a volume. Move your cursor over the gear wheel icon. You see a list of actions that you can perform for the selected volume:

Taking a Snapshot: In general, a snapshot is the representation of a storage volume at a point in time. You can take a snapshot, revert to it later, and create volumes and templates from existing snapshots. Click **Take Snapshot** and accept the terms and conditions. Click **Confirm**.

Scheduling Snapshots: You can schedule the system to take snapshots at regular intervals. For example, you can schedule the snapshots of the volume to be taken at six hour intervals. If you enter ten in the Keep field, only the ten most recent snapshots will be retained, and the earlier ones discarded. Depending on the time intervals at which you specify the snapshots to be taken, the system sets an upper limit on the Keep field. For example, if you choose hourly snapshots to be taken, the Keep field takes a maximum value of eight. Click **Recurring Snapshots** and specify the details of the schedule. Accept the terms and conditions and click **Apply**.

Attaching a volume: You can select a data volume and attach it to a VM. Click **Attach Volume** and specify the details. Accept the terms and conditions and click **OK**.

Downloading Volume: You can select a data volume and download it. Click **Download Volume**. Accept the terms and conditions and click **OK**.

Creating a Template: Click **Create Template** and specify the details for the template that you want to create from the volume. Accept the terms and conditions and click **OK**.

Deleting a volume: Click **Delete Volume**. Click **Confirm**.

Adding a volume

To add a volume, follow these steps:

1. Log in to CloudPortal Business Manager as user.
2. Click My Services.
3. Select the cloud service and the service instance.
4. Click Manage Resources.
5. Click Volumes and then click **Add New**.
6. Specify the details for the new volume.
7. Select the checkbox to accept the terms and conditions and click **OK**.

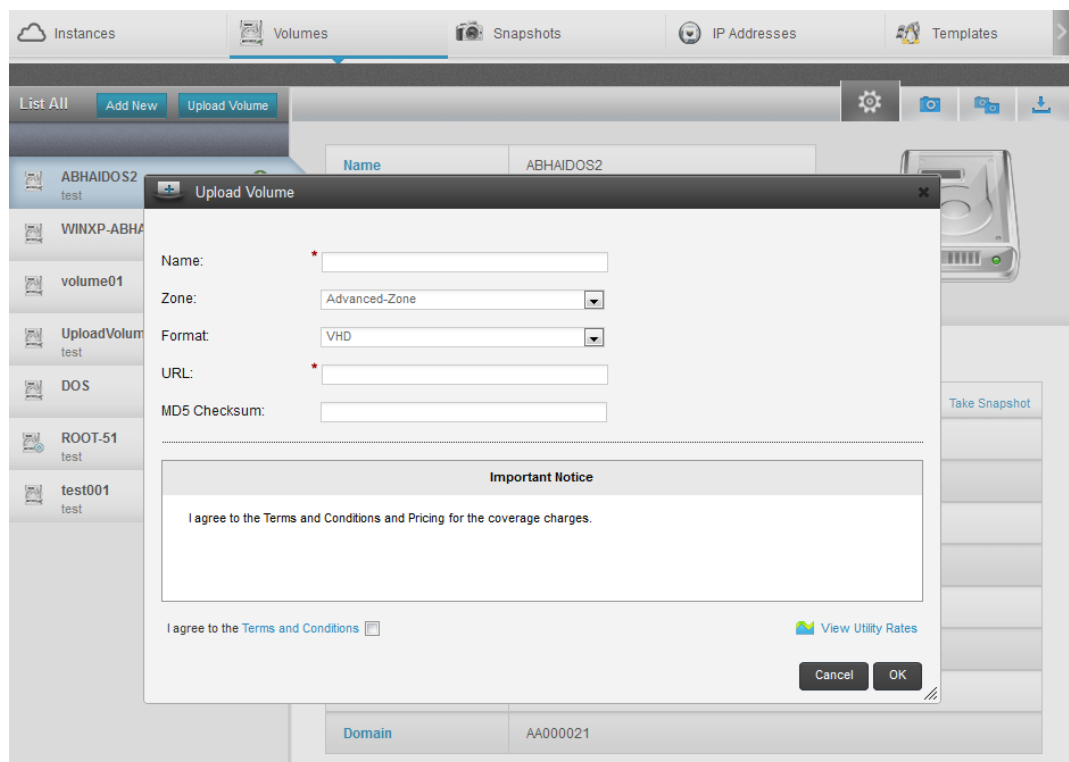
Uploading a Volume

Uploading a volume refers to the process where the existing data can be made accessible to a virtual machine. Root administrators, domain administrators, and end users can all upload existing volumes to VMs. The upload is performed using HTTP. The uploaded volume is placed in the zone's secondary storage.

You cannot upload a volume if the preconfigured volume limit has already been reached. The default limit for the cloud is set in the global configuration parameter `max.account.volumes`, but you can also set per-domain limits that are different from the global default.

To Upload a volume:

1. Create an MD5 hash (checksum) of the disk image file that you are going to upload. After uploading the data disk, CloudPlatform will use this value to verify that no data corruption has occurred.
2. Log in to CloudPortal Business Manager as user.
3. Click My Services.
4. Select the cloud service and the service instance.
5. Click Manage Resources.
6. Click Volumes.
7. Click Upload Volume.



8. Enter the name.
9. Select the zone where you want to store the volume.
10. Select the disk image format of the volume. Available formats are:

Hypervisor	Format
XenServer	VHD
VMware	OVA
KVM	QCOW2
OVM	RAW

11. Enter the secure HTTP or HTTPS URL that CloudPlatform can use to access your disk.

Note: The type of file at the URL must match the value chosen in Format.

12. Use the hash that you created in step 1. and Click OK.

Snapshots

In general, a snapshot is the representation (read-only copy) of a storage volume or a VM at a point in time.

The snapshots that you take in the Volumes and Instances sections appear in the left pane. You can filter the snapshots by Volumes and Virtual machines.

Note: VM Snapshots are applicable from CCP 4.2 onwards.

Volume Snapshots:

The Details Pane: To view the details of a snapshot, select it in the left pane.

ID: A unique ID assigned by your service provider.

Name: A name assigned by your service provider, with the date and time of creation of the snapshot at the end.

Volume: The volume from which this snapshot was made.

State: Backed Up, In Progress, and so on.

Interval Type: This shows if the snapshot is a recurring snapshot.

Created: When this snapshot was created (time and date).

Account: The user who created this snapshot.

Domain: The domain assigned to this snapshot by your service provider.

The Gear Wheel Menu

In the snapshots tab, filter by volume, then select a snapshot and move your cursor over the gear wheel icon to see the actions that you can perform. You can create a volume or a template or both from this snapshot. You can also delete this snapshot.

Create Volume: Enter a unique name in the Name field. Select the check box to agree with the terms and conditions, and click OK.

Note: The volume that you create here appears in the list of volumes in the Volumes page.

Create Template: You can create a template from an existing snapshot. You can launch virtual machines based on this template.

Name: A unique name.

Display Text: A descriptive text.

OS Type: The Operating System type (CentOS, Debian, Fedora etc.)

Public: If you set this to Yes, other users will be able to view and use this template. If you set this value to No, only you can use this template.

Password Enabled: If you want this template to be password protected, set this to Yes.

Note: The templates that you create here appear in the Templates tab.

Virtual Machine Snapshots:

The Details Pane: To view the details of a snapshot, select it in the left pane.

ID: A unique ID assigned by your service provider.

Name: A name assigned by your service provider, with the date and time of creation of the snapshot at the end.

Description: Description for the snapshot.

Parent Name: Name of the parent snapshot.

Current: Whether the snapshot is current or not.

Type: Either disk or disk and memory

Virtual Machine: The Virtual Machine that the snapshot belongs to.

State: State of the snapshot.

Created: When this snapshot was created (time and date).

The Gear Wheel Menu:

In the snapshots tab, filter by virtual machine, then select a snapshot and move your cursor over the gear wheel icon to see the actions that you can perform. You can revert VM to this snapshot. You can also delete this snapshot.

Revert VM to snapshot: Confirm your action by clicking **Confirm**.

Delete snapshot: Confirm your action by clicking **Confirm**.

IP Addresses

To view information about your IP address allocations, click the IP Addresses tab. You see a list of the public IP addresses allocated to your account.

Your virtual machines reside behind a virtual router. Subsequently this IP address could be that of your virtual router, and not of a particular virtual machine (which in turn could have a private IP address). You can see the following information in the **Details** section.

- **ID:** The unique ID for this IP address, assigned by your service provider.
- **Account:** The account to which the IP is assigned.
- **Zone:** The zone in which this IP address resides.
- **VLAN:** Information about the virtual local area network.
- **Source NAT:** Whether Network Address Translation is enabled on this virtual router.
- **Network ID:** The unique ID provided by your service provider.
- **Associated Network ID:** Associated network ID provided.
- **Domain:** The domain associated with the account.
- **Allocated:** The date and time at which this IP address was allocated.
- **Static NAT:** Is the NAT static or dynamic? (Request an additional IP address for your virtual router if static NAT is Yes.)

Acquiring an IP address

To acquire a new IP address, click **Acquire IP**. Specify if it is for a VPC or network, select the network or VPC, select the check box for terms and conditions, and click **Confirm**.

Firewall

Firewall rules defined. You can add more firewall rules.

Port Forwarding

Public Port: The port range that you would like to use.

Private Port: The corresponding private port range.

Protocol: The network protocol applicable. TCP, UDP, and so on.

Virtual Machine: The virtual machine to which to forward the port range.

Load Balancer

Name: A name for this Load Balancer rule.

Public Port: The externally accessible port that you would like to balance.

Private Port: The corresponding private port on a VM.

Algorithm: The process by which this balance will occur. Choose from the drop-down options.

VPN

If your VPN is already enabled, you see the message below telling you so.

Tasks performed:

Select the IP address. Move your cursor over the gear wheel icon. You see a list of actions that you can perform for the selected IP address.

Enabling or Disabling Static NAT: A static NAT rule maps a public IP address to the private IP address of a VM to allow Internet traffic to it. You can enable or disable static NAT for a particular IP address in a VPC. Click **Enable Static NAT**. Select the network and the VM and click **OK**. To disable Static NAT, click **Disable Static NAT** and then, click **Confirm**.

Releasing an IP: If you no longer need a particular IP, you can disassociate it from its VPC and return it to the pool of available addresses. Click **Release IP** and then, click **Confirm**.

Adding a Site-to-Site VPN connection: A Site-to-Site VPN connection helps you establish a secure connection from an enterprise datacenter to the cloud infrastructure. Click **Add Site-to-Site VPN**. Specify the details. Select the check box to accept the terms and conditions and click **Confirm**.

Networks

In the Networks tab, you see a list of default networks offered by your service provider.

In the **Details** pane, you see the following details:

Network Name: The name of the network.

Network Desc: A descriptive name to help you identify the network.

Zone: The zone in which this network resides; determined by your service provider.

Type: If it a shared or isolated network.

VLAN: Information about the virtual local area network.

Gateway: Gateway used.

Netmask: Net mask in use on the subnet.

Network Domain: Domain where the network is available.

Tags: Network tags.

Domain: Domain to which the account belongs.

Account: User account.

Adding a network

To add a new network, follow these steps:

1. Click **Add New**.
2. Specify the following details:
 - **Network name:** Name for the network
 - **Network Desc:** Description of the network
 - **Zone:** Zone to which the network will be added
 - **VPC:** Whether the network is Virtual Private Cloud-enabled
 - **Network Offering:** Network offering to use for the network
 - **Gateway:** Gateway to be used
 - **Netmask:** The netmask in use on the subnet

- Network Domain: Domain where the network will be available
 - Tags: Network tag to specify which physical network to use
3. Select the checkbox to accept the terms and conditions and click **OK**.

Adding VPC

To add a VPC, follow these steps:

1. Click **Add VPC**.
2. Specify the following details:
 - VPC name: Name of the VPC
 - Description: Description for the VPC
 - Zone: Zone where the VPC will be available
 - Internal Super Net:
 - Network Domain: Domain name
3. Select the checkbox to accept the terms and conditions and click **OK**.

Templates

A template is a reusable configuration for virtual machines. When users launch VMs, they can choose from a list of templates. A template is a virtual disk image that includes an OS type and is associated with a hypervisor type.

Template details:

Name: Name of the template

Hypervisor: Hypervisor of the template

State: Description of the template

ID: Template ID

Zone ID: ID of the zone where the template is available

Display Text: Description of the template

Type: Type of template, whether public or private

OS Type: OS type of the template

Password Enabled: Whether the template has password change script installed

Public: Whether the template is accessible to all users of the service

Account: Account to which the template belongs

Domain: Domain to which the template belongs

Created: Timestamp when the template was created

Adding a Template

1. Click Add New.
2. Specify the following:
 - Name: Name for the template
 - Display Text: Description of the template
 - URL: The URL that hosts the image
 - Zone: Zone where the template will be available
 - OS Type: Select the OS type of the image
 - Hypervisor: Hypervisor type
 - Format: Image format
 - Password Enabled: Specify yes for password reset feature that allows users to set a temporary admin or root password as well as reset the existing admin or root password.
3. Select the check box to accept the terms and conditions and click OK.

ISOs

An ISO is a read-only file that has an ISO/CD-ROM style file system. Users can upload their own ISOs and mount them on their guest VMs. ISOs are uploaded based on a URL. ISOs may be public or private, like templates. ISOs are not hypervisor specific.

Details

Name: Name of the ISO

ID: ID of the ISO

State: State of the ISO

Zone: Zone where the ISO is available

Zone ID: ID of the zone where the ISO is available

Display Text: Description of the ISO

Size: Size of the ISO in GB

Bootable: Whether or not a guest could boot off this ISO image

Public: Yes, if the ISO is available to other users

Featured: If the ISO is made more prominent for users to select

Cross Zones: If the ISO is available across zones

OS Type: OS of the ISO

Account: User who has created the ISO

Domain: Domain to which the ISO belongs

Created: Timestamp of creation of the ISO

Adding an ISO

1. Click Add New.
2. Specify the following:
 - Name: Name for the ISO image
 - Display Text: Description of the ISO
 - URL: The URL that hosts the image
 - Zone: Zone where the ISO will be available
 - Bootable: Whether or not a guest could boot off this ISO image
 - OS Type: Type of the OS
3. Select the checkbox to agree to the terms and conditions and click OK.

Setting service instance specific user settings

In addition to the username and password authentication, Citrix CloudPlatform supports using SSH keys to log in to the cloud infrastructure for additional security for your cloud infrastructure.

1. Log in to CloudPortal Business Manager UI.
2. Click on Root user at the top right hand corner of the UI.
3. Select My Profile from the drop-down.
4. Click My Services tab.
5. You can see the list of all applicable service instances (enabled for the user) on the right pane. Select the desired service.
6. Click My Settings. This will redirect you to SSH Key settings page where you can generate SSH Keys and upload SSH Keys.

FAQs

How to enable a shared account?

1. Login as root user.
2. Navigate to the **Administration > Account types**.
3. Select the Account Type.
4. Click **Service Controls > Edit**.
5. Enable **Resource Sharing Mode**.

What is the "Resource Sharing Mode" setting?

Resource Sharing Mode disabled by default. This means that each user of an account will see his own resources. For example, user 1 creates a VM called "SQL" and user 2 creates a VM called "webserver", user 2 cannot see "SQL" and user 1 cannot see "webserver". If resource sharing mode is enabled, all users in an account share the resources used and the VMs created. The second option is used for customers who act as managed service providers and manage the environment for their customers to some extent. They could have a service user in the customer account which has access to all resources and simultaneously the customer can have users who views the same resources.

User Guide

This guide includes information that you need to understand the CPBM UI and describes the activities that a user can perform in CloudPortal Business Manager UI.

User Management
Managing Company Setup
Managing User Profiles and Preferences
Viewing API Credentials
SSH Keys
Managing your Spend
Understanding your Bill
Browse Catalog and Subscriptions
Viewing Service Health
Understanding the Dashboard
My Services Tab

User Management

This section describes the different aspects of user management.

Users and profiles
Adding a user
Managing information of your account users
Unlocking accounts

Users and Profiles

About Users

A user is a computer's representation of a person who interacts with the computer software. CloudPortal saves a variety of information about each of its users.

The minimum set of information that the portal needs to know about each user is:

- First and last name
- Email address
- Profile
- Username
- Password

Optionally, users can provide additional details:

- Photo
- Time zone

About Profiles

A profile is a set of permissions that is saved under a name, such as Administrator or Manager. Every user of your portal is assigned a profile, including yourself. The profile is assigned when the user is added to the portal.

Profiles are useful for saving a frequently-used set of permissions so that they can be easily granted to a user in a single step. This is easier and less error-prone than setting each individual permission for every user. Profiles also ensure that multiple users with equivalent duties and needs are all granted the same permissions. For example, you might want to grant the "Create Customer Account" permission for the Sales profile but not for the Help Desk profile.

CloudPortal ships with a predefined set of profiles. You cannot add or delete a profile. If you have the "Create/Manage Profile" permission, you can select which permissions to grant under each profile name. Given below are the profiles in CloudPortal. The CloudPortal UI includes a fair amount of information on the predefined profiles, role names and their descriptions.

Customer Profiles

- Master User
- Billing Admin

Users and Profiles

- Power User
- User

Adding a User

As a master user of a customer account, you can add users for the customer account. These users can then access the full range of cloud services from CloudPortal Business Manager.

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click **Users > Add New**.
3. Enter the user details such as name and email address of the user in the Add User page.
4. Select the user's profile.
5. If required, click **Customize Welcome Email** to add a subject and custom welcome message in addition to the default welcome message that is provided by the system.
6. Click **Finish**.

The user will receive a email which contains a link to verify his email and to set the password. The user must log in through this link the first time to confirm that the email address was valid. If the user has lost the email or has not received the email, the master user can resend the email once again by using the resend verification email link.

To resend the verification email:

1. Click **Users**. You can view all users of your customer account along with user details.
2. Select the user whose information you want to manage.
3. Place your cursor over the manage icon (that looks like a gear wheel) and click **Re-send Verification**.

Note:

- Please note that the email id provided by user is not editable. Ensure that the email id is valid and active.
- If you are just testing the Add User functionality and sending the welcome email to yourself, log out of CloudPortal Business Manager before you click the email's login link. Trying to log in as two different users simultaneously can cause the new user's details to be stored incorrectly.

Managing information of your account users

As a master user, you can view user information, edit the information, deactivate a user, or remove a user. You can manage user information such as user details, reset user passwords and preferences, and email addresses for notifications.

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click the **Users** tab. You can view all users of your customer account along with user details.
3. Select the user whose information you want to manage.
4. Place your cursor over the manage icon (that looks like a gear wheel) and click **Edit**.
5. Make the desired changes. You can modify any of the user properties such as time zone, language, user details, and user password.
6. To add an image, click Manage your profile image at Gravatar and follow the instructions to upload a photo to gravatar.com.
7. To change the time zone and preferred language, follow these steps:
 - a. Click **Edit** in the Preferences pane.
 - b. Select the new time zone and preferred language and click Save.
8. To change the user details, follow these steps:
 - a. Click **Edit** in the Details pane.
 - b. Make the required changes and click Save.
9. To request the user for a password reset, follow these steps:
 - a. Click **Reset Password** in the Login Information pane.
 - b. Click **Confirm** to send a mail to the user to reset the password.
10. To configure the notification settings, click Notifications and if required, specify additional email address to use for notifications.
11. To deactivate the user, click the **Deactivate** tab next to the user name.
12. To remove the user, click the **Remove** tab next to the user name.

Unlocking accounts

CloudPortal Business Manager temporarily disables a user account and prevents a user from logging in after a certain number of failed login attempts. The number of failed login attempts is determined by the value that you have specified for the `logins.lockThreshold` parameter during security configuration. The user can request for a password reset and unlock the account.

1. Open the CloudPortal Business Manager UI.
2. Click **Request a reset**.
3. Specify your user name and click **Submit**. CloudPortal Business Manager sends an email to your registered email address with instructions to reset your password.
4. Follow the instructions and reset your password. CloudPortal Business Manager resets your password and unlocks your account. You can now log in.

Note: If the Phone Verification feature is enabled, CloudPortal Business Manager asks you for your phone number for verification before you can reset your password.

Managing company settings

As a master user of a customer account, you can manage your company settings such as company contact information, credit card information, and billing address. You can monitor your account information and billing details. The actions that you can perform depend on your account type.

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click the drop-down menu icon in the top right corner next to your name and click **Company Setup**.
3. Click the **Contact** tab.
4. In the Edit Company Info section, click **Edit**.
5. Make the changes as required and click **Save**.
6. You are required to provide a credit card number and other payment details when first signing up with CloudPortal Business Manager. To edit your credit card information and billing address:
 - a. Click the **Billing and Payment** tab.
 - b. To edit your credit card information, click **Edit** in the Credit Card section, make the required changes, and click **Save**.
 - c. To edit your billing address information, click **Edit** in the Billing Address section, make the required changes, and click **Save**.

Note:

- The Billing and Payment tab is not visible to a Trial Master user (if the account type is trial).
 - Credit Card information is visible to a Retail Master user (if the account type is retail).
7. Retail and Trial account users can request for a conversion of their account type:
 - a. Click **Request Account Type Conversion**.
 - b. Select the account type and click **Save**. By default, the following conversions are supported (can be customized by the service provider):
 - Trial to Retail
 - Trial to Corporate
 - Retail to Corporate
 8. To view your account details, click the **General** tab. You can view the activation details, billing details, account information such as account type, customer ID, and the master user of the customer account.

Managing user profiles and preferences

Your personal profile stores details such as a photo, email address, contact information, and login credentials. You can change your password and other data about yourself.

1. Log in to CloudPortal Business Manager UI as a user.
2. Click the drop-down menu icon in the top right corner next to your name and click **My Profile**.
3. To add an image, click **Manage** your profile image at Gravatar and follow the instructions to upload a photo to gravatar.com.
4. To change the time zone and preferred language, follow these steps:
 - a. Click **Edit** in the Preferences pane.
 - b. Select the new time zone and preferred language and click **Save**.
5. To change your personal details, follow these steps:
 - a. Click **Edit** in the Details pane.
 - b. Make the required changes and click **Save**.
6. To change your password, follow these steps:
 - a. Click **Change Password**.
 - b. Enter your old and new password and confirm
 - c. Click **Save**.
7. To provide an additional email address to use for notifications, follow these steps:
 - a. Click the **Email Addresses** tab.
 - b. Specify the email address and click **Add Email**.

Note: You can change your primary email address and also add secondary email addresses.

To change your primary email address:

- i. Add the secondary email address.
 - ii. Verify the secondary email address.
 - iii. Make this email address as your primary email address.
8. To configure your notification settings, click the **Notification Preferences** tab and specify.
 9. To view your API credentials, follow these steps:

- a. Click the **API Credentials** tab.
- b. Enter your password to proceed (as an additional security measure) and click **Proceed**.
- c. View your API credentials.

Changing your primary email address

You can change your existing primary email address, if required, and provide a new email address as the primary email address.

1. Log in to CloudPortal Business Manager UI as a user.
2. Click the drop-down menu icon in the top right corner next to your name and click **My Profile**.
3. Provide the new email address:
 - a. Click the **Email Addresses** tab.
 - b. Specify the email address and click **Add Email**.
4. Verify the secondary email address.
5. Navigate to **My Profile > Email Addresses**.
6. Click **Make Primary**.

Viewing API Credentials

When you access the CloudPortal Business Manager BSS APIs, the API request has to be signed so that the user is authenticated and it is verified if the user is authorized to execute the command. A user must have both, the API key (uniquely identifies the account) and the secret key for the account before proceeding with the signing process.

The CPBM UI displays your BSS API credentials and the API credentials (for your account) that enable you to make API requests to the enabled cloud service instances. Some cloud services (such as CloudPortal Services Manager) do not have API credentials and for such cloud instances you can only view your username.

To view the API credentials:

1. Log in as the root user of the account.
2. Click your username at the top right hand corner of the UI.
3. Select My Profile from the drop-down list.
4. Click the API Credentials tab.
5. Enter your password to proceed (as an additional security measure) in the pop-up box.
6. Click **Proceed**. You will see the API credentials.

SSH keys

SSH keys are used for authentication into a newly launched VM. You can connect to a VM without using a username and password, by using the SSH keys. For this to work, user has to either upload a private key for which the public key is saved locally, or generate a key pair and save the generated public key locally. And then, select any key while launching the VM. This option is available for the CCP cloud service.

1. Log in to CloudPortal Business Manager UI as a user.
2. Click My Services tab.
3. You can see the list of all applicable service instances (enabled for the user) in the right pane. Select the desired service instance.
4. Click My Settings. This will redirect you to SSH Key settings page where you can generate SSH Keys and upload SSH Keys.
5. To generate the SSH key, provide the key name and click **Generate**.
6. To upload the SSH key, provide the key name and then upload the key.

Managing your spend

As a master user, you can set your own account budget and spend alerts. You can track your spend by setting alerts, so that you get a notification when your spend reaches a certain percentage of your spend budget.

Setting your budget

As a master user, you can set the spend budget for your account.

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click the **All Alerts** tab or click the **Set Budget** link in the Spend vs. Budget area of the dashboard.
3. Click **Set Budget**.
4. Specify the budget amount and click **OK**.

Creating alerts

As a master user, create spend alerts so that you get notifications when your spend crosses the set limit.

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click the **All Alerts** tab or click the **Manage Spend Alerts** link in the Spend vs. Budget area of the dashboard.
3. Click **Add New**.
4. Specify the percentage value of your spend budget after which you will receive the an alert.

Viewing spend summary

The Spend Vs. Budget pane shows your spend details. You can view your actual spend, last month's spend and the net balance. You can click the Net Balance link to view your renewal charges, new charges, payments and credits, and the net balance. the budget that you have set, your renewal charges, new charges, and current balance. You can click the Usage & Billing tab to view in detail your spend.

You can click the appropriate links to set your account budget and spend alerts.

Viewing spend notifications

You can configure spend alerts and view the alert notifications that are generated when your spend crosses pre-defined threshold limit. You will receive the notifications by email or phone depending on your notification preferences. You can view the alerts in the Notifications tab.

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click the All Notifications tab. A list of all notifications is displayed.
3. View the notifications of category "Account Spend."

Understand Your Bill

CloudPortal Business Manager generates invoices for products/product bundles that you have used. It tracks the resources that you use, calculates the amounts owed, and handles invoicing and payments for you automatically. Auto-payment is done in accounts with payment information on file with CloudPortal Business Manager, such as CyberSource, using your credit card information updated in CPBM. All the credit card details are stored by the gateway provider.

The **Usage & Billing** tab provides information about your resource usage, billing, subscriptions, and payments.

Note: Only certain administrative profiles, such as Finance Admin can view the billing information of other users.

To navigate to the Usage & Billing tab

1. Click on the View Dashboard icon on the right pane
2. The Dashboard page appears
3. Click on the **Usage & Billing** tab
4. The information is displayed under four tabs, such as **Activity**, **Subscriptions**, **Invoices**, and **Payments**

Activity

The **Activity** tab under the **Usage & Billing** tab displays all the account statements on the left pane. Clicking on an invoice displays the following details:

UI Element	Description
Balance Forward	The amount carried forward from the previous invoice. If there is no amount from the previous invoice, the amount will be 0.
Payments + Credits	The sum of Payments and Credits. <ul style="list-style-type: none">• Payments: The payments that you have made towards the last invoice.• Credits: A credit adds funds to your balance in CloudPortal Business Manager. Any future charges incurred by you will be paid out of this credit balance first. Your Administrator can issue credits.
Net Balance	The amount for which you must make payment. The amount is obtained using the formula: Balance Forward - (Payments + Credits) + New Charges

Renewal Charges	The amount that is associated for certain subscriptions. Renewal charges are fixed charges.
Subscription Charges	The amount that is associated with every product subscription such as a one-time fee followed by monthly renewal fees. Subscription Charges are fixed charges.
Utility Charges	The amount charged for using the portal resources that is directly proportional to the usage. Utility Charge also includes debit notes for cancelled credits. Note: To view more details about invoice, specific rows displays additional information about the product and their charges.
Discount(s)	The discount amount, if applicable.
Sub-total	The sum obtained from the formula: Subscription Charges + Utility Charges - Discount(s)
Taxes	The tax amount that is applicable.
New Charges	The amount obtained from the formula: Sub-total + Taxes

In addition, a detailed computation of the following is displayed in the **Activity** tab:

- Payments + Credits
- New Charges
- Renewal Charges

Subscriptions

The **Subscriptions** tab under the **Usage & Billing** tab displays the list of all the subscriptions on the left pane. For more information, refer to [Browse Catalog and Subscriptions](#)

The gear wheel icon on the right side enables you to

- Terminate Subscription - Terminates the selected subscription after confirmation
- Provision Resource - Takes you to the **Infrastructure** screen where the resource can be provisioned

Note:

- A green colored circle adjacent to a subscription indicates that it is active and the resource has been provisioned
- An orange colored circle adjacent to a subscription indicates that it is active but the resource is not provisioned

Clicking on a subscription from the left pane displays the following:

UI Element	Description
Product Bundle	The name of the product bundle.
Service Instance	The name of the cloud service for which an instance is created.
State	The state of the specific subscription. The states will be one of the following: <ul style="list-style-type: none"> · Active · Expired · Cancelled · New
Subscription Charges	This tab displays the following details: <ul style="list-style-type: none"> · Charges: The type of the subscription such as One-time and Recurring · Prices: The amount charged for the type of subscription
Entitlements	This tab displays the following details: <ul style="list-style-type: none"> · Item: The name of the resource used · Unit Price: The rate for the specific resource · No charges for usage up to: The number of units up to which the resource usage is not charged
Resource Details	This tab displays the following details: <ul style="list-style-type: none"> · Name: The detail of the type of resource/service · Value: The value for the specific resource/service
Configuration Data	The configuration data of the resource that is subscribed and provided by the underlying cloud service. For example, in case of Citrix CloudPlatform, the configuration data for a storage type resource contains the following: <ul style="list-style-type: none"> · Zone: The name of the zone · Disk Offering: The name of the disk offering · Disk Size in GB (Only for Custom Disk Offerings): The size of the disk

Invoices

The **Invoices** tab under the **Usage & Billing** tab displays the list of all the invoices generated (both current and previous) in the left pane. Invoices have invoice line items that detail the charges that make up the invoice.

Clicking on a invoice from the left pane displays the following:

UI Element	Description
State	The state of the invoice for the specific billing period, either Provisional or Posted.
Type	The type of the invoice. The types of invoices are: <ul style="list-style-type: none"> · Subscription · Close Out · Renewal · Debit Note
Subscription Bundle	The name of the product bundle that you have subscribed
Detail	The details of: <ul style="list-style-type: none"> · Total Amount: The invoice amount · Due Amount: The balance amount to be paid, if applicable

Payments

The **Payments** tab under the **Usage & Billing** tab displays the list of all the payments made on the left pane. Clicking on a payment date from the left pane displays the following:

UI Element	Description
Date	The date when the payment was made.
UUID	Universal Unique Identifier
Amount	The amount for which the payment was made
Details	This tab displays the following: <ul style="list-style-type: none"> · Description: The detail about the specific payment · Type: The type of payment · State: The state of the payment transaction · Transaction ID: The unique ID for the specific payment transaction <p>Clicking on the Issue Charge Back link reverses the payment transaction after confirmation from the user.</p>

Browse Catalog and Subscriptions

Product catalogs are defined for each channel by the service provider. As a user, you can browse through the product catalog to view the various offerings of the service provider. A service provider can also access a customer account dashboard and view the product catalog for that account.

A user can access the Catalog page by clicking either the Catalog tab or the Browse Catalog link on the Dashboard. An account user registered in a particular channel will see the product catalog defined for that particular channel, with product prices displayed in the currency that is specified for that account.

The service provider decides how the service offerings are displayed to the users while browsing the catalog. There are two modes of display, Pay-as-you-go and Product Bundle. The service provider sets the pay-as-you-go configuration parameter to true or false, during the installation of CloudPortal Business Manager. This setting controls how a user can view the product offerings, in a pay-as-you-go mode or in a product bundle mode.

Pay-as-you-go mode:

When you browse the catalog, you will see the Utility Rate Card that lists the price for each resource or product and you are charged for the amount of your actual consumption. For each one of the resources (products) that generates a usage based on your activity, you are charged by the prices shown in the rate card. When you provision a resource using this option, a subscription is created by the name "Utility Bundle" and is listed in your list of subscriptions that can be accessed from the Usage & Billings tab. You can terminate the subscription at any time. However, you cannot reprovision or reconfigure the resource.

Product Bundles mode:

When you browse the catalog, you can view the product bundles that are defined for and available to your channel. The bundles are further classified by the frequency of the recurring charges, for example, monthly, quarterly, or annual. This information is displayed in the tabs of the Pricing section.

Based on your resource needs, you can select the resource type that you want and filter by provisioning constraints (or resource components) and frequency of recurring charges to get a list of product bundles that match your selection criteria.

Each product bundle displays in its details, the bundle charges such as one time charge for the activation of the bundle, the recurring charges and their frequency, and entitlements that are given when you subscribe to the product bundle. The prices are specified in the currency set for the account. Apart from the charges specified for the bundle, charges may apply for any product consumption that is not covered by the bundle. The utility rates for such usage are listed in the Utility Rate card that is accessible from the bundle details.

CloudPortal Business Manager enables you to select a product bundle, specify the resource components, configure, and subscribe to the bundle.

You can either subscribe to a fresh subscription, reprovision an existing resource, or reconfigure an existing subscription. After subscribing to a product bundle, you can choose

to provision the resources immediately or later.

Anonymous Browsing

An anonymous user can view the product offerings of the service provider, without logging in to CloudPortal Business Manager UI. An anonymous user will view the catalog defined for the default channel. As an anonymous user, you can view the product catalog to check if the products offerings suit your resource needs and then choose to sign up with the cloud service provider. The anonymous user can access the catalog by clicking the Browse Catalog link on the Login page.

Understanding the Subscription Cycle

A subscription is a plan that allows the customer to make recurring payments for the use of a product or service. CloudPortal Business Manager enables you subscribe to the cloud services that are offered by the service provider. These offerings are presented to the customers in the form of various plans or product bundles. These are unique to the underlying service instance and to the channel that a customer belongs to.

As a user, you can select the resource type and filter by provisioning constraints to get a list of the product bundles that satisfy your selection criteria and business needs. You have to select a bundle that best suits your requirements and configure and subscribe to it.

After you accept the terms and conditions of the customer agreement, you can subscribe to a product bundle and either provision the resource immediately or choose to do it later. The subscription gets created regardless of whether it is provisioned or not. If you provision a resource, CloudPortal Business Manager will invoke the connector calls to provision the underlying resource and the subscription is created regardless of whether the underlying provisioning succeeds or not. In case of failure, an email notification is sent to inform that provisioning has failed.

If you do not provision the resource or if provisioning fails, then your subscription state is active (but unused).

All subscriptions are charged immediately on subscribing in CloudPortal Business Manager. If the user account has been configured with the setting Auto Payment Needed set to true, payment is deducted from the user immediately on subscribing. If a user chooses annual, the user is charged for twelve months up-front. Proration happens for recurring charges and entitlements. Same entitlements are given every month and the user has to pay for the month in advance.

A user can terminate a subscription at any time. However, if you terminate a subscription at any time during a billing period, no refund is given as the subscription charges have already been paid for, in advance.

Subscription states

A subscription can be in any one of the following states:

- New
- Active
- Expired
- Cancelled

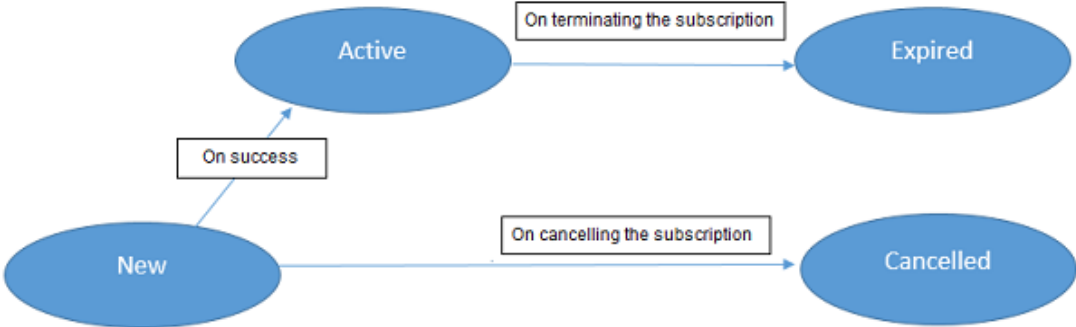


Figure 1. Subscription states

State	Description
New	The user has subscribed, but some action is pending before it can be moved to the Active state (for example, approval is pending). This subscription may move to the active state once the pending action is completed.
Active	When all pending activities are completed and if resource needs to be provisioned, on successful provisioning of resource, the subscription moves to the active state. The subscription is now in use and the user is charged for it.
Expired	The subscription has been terminated by the user. A subscription moves from New to Active and remains in the Active state until it is terminated by the user. The user is charged for the subscription in the billing period in which the subscription is terminated.
Cancelled	The subscription is not approved either because the payment does not happen or there is a workflow configured for subscription activation and the master user does not approve or for any other reason. The subscription is moved to the Cancelled state either by CPBM or by the user. The user is not charged for the subscription.

Subscribing to a Product Bundle

As a user, you can select the resource type that you want and filter by provisioning constraints to get a list of product bundles that satisfy your selection criteria. You can then configure and subscribe to a product bundle.

1. Log in to CloudPortal Business Manager UI as a user.
2. Either click **Catalog** or click Home > Browse Catalog. The Catalog page displays the list of services enabled for you, along with the service instances.
3. Select the service and the service instance.
4. Select the resource type and specify the filters and select the resource components. The Pricing tab at the top left displays the bundles categorized by the frequency of recurring charges. The prices are displayed in the currency that is set for your account.
5. Select the appropriate pricing tab to view the product bundles that match your selection criteria.
6. Select the product bundle and click Configure & Subscribe.
7. In the Configure and Subscribe page, select the resource components if you have not already selected them earlier and complete the configuration (optional). In the right pane, you can view the following:
 - Selections
 - One-time charges
 - Recurring charges
 - Utility charges (for additional usage that may be generated for the resources that you have selected)
8. Review the user policy and accept the terms and conditions of the Customer Agreement.
9. Click Subscribe and Provision or click Subscribe and Provision Later. The subscription get created immediately, regardless of whether it is provisioned or not. On subscribing, you can choose to navigate to the dashboard, my resources page, catalog page, or subscriptions page. In the Subscriptions page, the bundle is listed as Active (unused). You can now provision the resource.

Provisioning a resource

A resource can be provisioned by subscribing to a bundle. All subscriptions are listed in the Subscriptions page under the Billing and Usage tab. If a resource is not provisioned or if the provisioning has failed, there is no resource tied to the subscription and the subscription is listed as Active (unused). You can provision a resource for the subscription.

1. Log in to CloudPortal Business Manager UI as a user.
2. Click Usage & Billing > Subscriptions.
3. Select the subscription, move the cursor over the gear wheel and click Provision Resource.
4. Select the resource components and provide the resource properties. Your selections are listed in the right pane.
5. Accept the terms and conditions and click **Provision**.

Reconfiguring a resource without changing the bundle or subscription

If you want to reconfigure a resource, you can do so by reconfiguring the subscription. You can select a subscription in active state and reconfigure the resource. Only those components that are reconfigurable can be modified (the reconfigurable flag is set to true in the connector service definition). Except the resource components, no other changes are possible. If the bundle definition does not match your selection, the subscription is listed as incompatible. You can also reconfigure using any of your un-provisioned subscriptions. In this case, your resource is no longer attached to the original subscription and gets attached the selected un-provisioned subscription.

When you want to reconfigure a resource, you can also use any of your un-provisioned subscriptions to reconfigure a resource.

1. Log in to CloudPortal Business Manager UI as user.
2. Click Usage & Billing > Subscriptions.
3. Select the active subscription where you want to reconfigure the resource.
4. Move the cursor over the gear wheel and click **Change Subscription**.
5. Select the reconfigurable components with characteristics that suit your current needs.. If your selection is compatible with the given subscription, you can reconfigure. If your selection is incompatible with the selected subscription, it is listed as incompatible. If there are any other un-provisioned subscriptions, you can choose to configure using such a subscription.
6. Click **Reconfigure**.
Note: You cannot modify any resource properties.
7. Accept the terms and conditions and click **Provision**. There is no change in the charges applicable, as you have already been charged for the bundle.

Reconfiguring with a new bundle or subscription

You can reconfigure a resource if the resource characteristics no longer suit your needs. In such cases, you can choose to reconfigure your resource to match your unprovisioned subscriptions. However, if your selection is incompatible with your existing or unprovisioned subscriptions, you can choose to reconfigure your resource using a new subscription.

1. Log in to CloudPortal Business Manager UI as user.
2. Click Usage & Billing > Subscriptions.
3. Select the active subscription where you want to reconfigure the resource.
4. Move the cursor over the gear wheel and click **Change Subscription**.
5. Select the reconfigurable components with characteristics that suit your current needs.
6. If your selection is incompatible with the current subscriptions or with any of your unprovisioned subscriptions, click the **For new subscriptions** tab.
7. Select the bundle that best suits your needs and click **Configure and Subscribe**.

Note: Your resource is no longer attached to the original subscription and gets attached to the new subscription. **You cannot modify any resource properties.**

8. Accept the terms and conditions of the customer agreement and click **Reconfigure**.

Note: Since you have subscribed to a new bundle, you are charged as per the charges specified in the bundle and a new subscription is created, which is listed in the Subscriptions page.

Filter Subscriptions

Users can view their subscriptions and filter them based on the state of the subscription. A master user can view all subscriptions of the account and filter them based on user and subscription state.

1. Log in to CloudPortal Business manager as Master user or user.
2. Click Usage & Billing > Subscriptions.
3. Select the user from the **Filter by** drop down list. The option to filter by user is visible only to the Master user.
4. To filter the subscriptions by their state, select the subscription state from the **State** drop down list.
5. View the subscriptions based on your filter selection.

Terminating a subscription

You can terminate a subscription if you no longer need the resource.

1. Log in to CloudlPortal Business Manager UI as a user.
2. Click Home > Usage & Billing.
3. In the **Subscriptions** tab, select the subscription that you want to terminate.
4. Place the cursor over the manage icon (that looks like a gear wheel) and click **Terminate Subscription**.
5. Confirm your action.

Provisioning a resource from the pay-as-you-go mode

You can provision a resource using the pay-as-you-go mode. You are charged on the basis of usage generated and the pricing is determined by the utility rate card.

1. Log in to CloudPortal Business Manager UI as a user.
2. Either click **Catalog** or click Home > Browse Catalog. The Catalog page displays the list of services enabled for you, along with the service instances.
3. Select the service and the service instance.
4. Select the resource and click **Go**.
5. Click the resource components option that best suits your requirement.
6. Select the resource components and complete the configuration (optional). Your selections and the utility rates are displayed in the right pane.
7. Accept the terms and conditions of the customer agreement and click **Provision**. The subscription gets created and it is listed as Utility Bundle in your list of your subscriptions, with the state shown as active.

Filing support tickets

As a tenant user (Master and his own users) you can file a support ticket to report an issue and request for its resolution.

1. Log in to CloudPortal Business Manager UI as a Master user or normal user.
2. Click Support > Tickets.
3. Click **Add New**.
4. Enter a title and describe the issue for which the support ticket is being raised.
5. Click **Submit**.

Customer Users having "Account Ticket View" role can see all the tickets in their own accounts. By default Account tickets view role is enabled for all customer users.

Customer Users having "Account Ticket Manage" role can create, comment, and close all the tickets which are created by all the users in their own accounts. By default this role is enabled for the Master User.

If the service provider wants any other customer user to have the ability to create, comment, and close the tickets created by all the users in their account, then the "Account Ticket Manage" role needs to be enabled for that customer user by the service provider.

Viewing service health

You can view the current status of your service from the Service Health pane of the dashboard. You can view the service status on any particular day from the Support tab. Service health is visible to users within your account and is a quick way to let your users how the system is doing. You can also view when maintenance activity has been scheduled.

1. Log in to CloudPortal Business Manager UI as Master user.
2. You can view the current status from the Service Health pane. Alternatively, you can click the **Support** tab. Service health status can be either issue, disruption, or resolution.
 - Issue: There is an issue with the service.
 - Disruption: The service is not functioning.
 - Resolution: The service issue has been resolved.
3. Click **Maintenance** to find out when maintenance is scheduled.

Note: Service Health is displayed for service instances which are enabled for that particular account.

Health status for a particular day is determined by the latest service health status for the day. The Status History in the right pane displays the health status history for the day, including the time.

Understanding the dashboard

When you log in to CloudPortal Business Manager, the landing page is the dashboard. The dashboard gives you an at-a-glance view of your account. The dashboard provides a summary view which varies depending on which user is logged in. Links are provided to allow you to navigate to the pages to view more information or take necessary action.

Who can use the Dashboard

Every user, no matter their role, has a dashboard.

- The Master user can see information on service health, spend, spend vs budget, active users and active subscriptions, the billing information, notifications, service tickets, and so on.
- A normal user can see information on service health, active subscriptions (of that user only), spend, notifications, and tickets.

How to Use the Dashboard

1. Log in to CloudPortal Business Manager UI as Master user (or normal user).
2. At any time, you can return to the dashboard by clicking **Home > Dashboard**.

Master user dashboard view

- **Active users:** Number of active users in the account .
- **Active subscriptions:** Number of active subscriptions for the account.
- **Spend:** The spend in the currency specified for the account.
- **Spend vs Budget:** You can view the budget set, the actual spend, and the net balance. Click the appropriate links to set the budget and manage spend alerts.
- **Understand your bill:** You can see the start date of your billing period. Click the appropriate links to help you understand and manage various aspects of your bill.
- **Service Health:** You can get an at-a-glance summary of the service health of the cloud services enabled for your account. Click View All to see the details.
- **Getting Started:** You can click the link to browse the product catalog, see the product offerings, and subscribe to product bundles.
- **Notifications:** You can view the notifications for events and issues, for the current date and the previous date. Click the appropriate links to view all notifications and manage your notification preferences.
- **Tickets:** You can view any outstanding support tickets that are not yet addressed. Click the appropriate links to view all tickets filed and the maintenance schedule or submit a

support ticket.

- **My Tasks:** You can see all the pending tasks (if any) for the master user. Click **View all Tasks** to view all tasks.
- **Users:** You can view all the users present in that account. Click the appropriate links to edit user information or to add a new user, respectively.

Normal user dashboard view

- **Active subscriptions:** Number of active subscriptions for the your user account. Does not include the subscriptions of other users of your account.
- **Spend:** The spend in the currency specified for the account.
- **Service Health:** You can view the current service health of the cloud services enabled for your account. Click **View All** to see the details, submit support tickets, and view the maintenance schedule.
- **My Tasks:** You can view all the pending tasks for you. Click **View all Tasks** to view all tasks.
- **Notifications:** You can view the notifications for events and issues, for the current date and the previous date. Click the appropriate links to view all notifications and manage your notification preferences.
- **Tickets:** You can view any outstanding support tickets that are not yet addressed. Click the appropriate links to view all tickets filed or submit a support ticket.
- **Getting Started:** You can click the link to browse the product catalog, see the product offerings and subscribe to the product bundles.

My Services Tab

The My Services tab allows the Master user to sign up for services offered by the service provider. The various services offered by the service provider are listed by the cloud service type and service instance.

Only the Master user can enable a service. In the My Services tab, for a new cloud service connector, the Master user will see the Enable button that must be clicked to enable the new service.

Once a service instance is enabled by the Master user, all normal users of that account can access the resource management console, subscribe to bundles, view the utility rates, and manage their settings from this tab.

Enabling a service

As a Master user, you can enable a service for the users of your account. Refer to [Enabling a service](#).

Links

All services enabled for the account are listed in the My Services tab. For each service instance enabled for your account, the user has the following links:

- [Manage Resources](#)
- [Subscribe to Bundles](#)
- [My Settings](#)
- [View Utility Rates](#)

In addition, all enabled service instances are listed in different tabs. A user can also click the links to directly access the resource management console of that service instance.

Managing Resources

You can access the resource management console and manage the resources. For more information on managing the resources, contact your service provider for the user guide for a particular service.

Subscribing to Bundles

For more information on subscribing to bundles, refer to the section on Subscriptions.

My Settings

The My Settings link allows you to generate SSH keys and upload them. This option is available for the CCP and ACS cloud services.

SSH Keys:

SSH keys are used for authentication into a newly launched VM. You can connect to a VM without using a username and password, by using the SSH keys. For this to work, user has to either upload a private key for which the public key is saved locally, or generate a key pair and save the generated public key locally. And then, select any key while launching the VM. This option is available for the CCP cloud service.

1. Log in to CloudPortal Business Manager UI as a user.
2. Click **My Services** tab.

You can see the list of all applicable service instances (enabled for the user) in the right pane. Select the desired service instance.

3. Click **My Settings**.

This will redirect you to SSH Key settings page where you can generate SSH Keys and upload SSH Keys.

4. To generate the SSH key, provide the key name and click **Generate**.
5. To upload the SSH key, provide the key name and copy the public key in the text box, and then click **Upload**.

Viewing Utility Rates

A product is a basic unit of consumption in CloudPortal Business Manager and every product has a utility or unit price associated with it. You can view utility rates for all products of the service instance.

1. Log in to CloudPortal Business Manager as a normal user.
2. Click **My Services**.
3. Select the service instance and click **View Utility Rates**.
4. View the products and their corresponding per unit rates.

Enabling a service

As a Master user, you can enable a service for the users of your account.

1. Log in to CloudPortal Business Manager UI as Master user.
2. Click **My Services** tab.
3. Select the service that you want to enable and click **Enable**.

CloudPortal Services Manager Connector Guide

This document describes how to install and administer the Citrix CloudPortal Business Manager connector for CloudPortal Service Manager (CPSM). The connector provides the following functionalities:

- Customer admin (master user) self-service customer account creation in CPSM
- Customer admin (master user) self-service customer plan provisioning for Hosted Exchange, Hosted Apps and Desktops, and SharePoint 2010
- Single sign on into CPSM portal from CPBM
- Usage collection and billing for Hosted Exchange, Hosted Apps and Desktops, and SharePoint 2010 customer plans and user plans

This connector is experimental and not intended for production deployment. This document covers the following topics:

System Requirements
Connector Installation
Connector Service Configuration
Connector User Roles
Connector Usage Types
Connector Resource Types
Connector Account Configuration
Single Sign On (SSO)

System Requirements

The connector has the following system requirements:

- CloudPortal Business Manager v2.1
- Cloud Portal Service Manager v11.0

When the customer administrator is provisioned into CPSM from CPBM, CPSM automatically generates a password. To email the CPSM generated default password to the customer administrator, the email server has to be configured in CPSM.

It is recommended to create Customer Plans, User Plans, and Offerings in CPSM before adding the service instance in CPBM. This will enable creating products and product bundles in CPBM immediately after adding the Service. Customer Plans, User Plans, and Offerings can be added later in CPSM.

Note: Ensure that names of Customer Plans, User Plans, and Offerings (when configured or added) are unique in CPSM, so that there is no ambiguity when creating products and product bundles in CPBM.

Installing the Connector

Following are the steps to install the connector for CloudPortal Service Manager in CloudPortal Business Manager.

1. SSH into CloudPortal Business Manager installation.
2. `sudo service cloud-portal stop`
3. Transfer the binaries `com.citrix.cpsm-<x.x.x>.jar` and `com.citrix.cpsm.fragment-<x.x.x>.jar` to the location `/usr/share/vts3/connectors`.
4. Add the line `<artifact type="bundle" name="com.citrix.cpsm" version="[<x.x.x>, <x.x.x>]"/>` within the plan tag in the file `com.citrix.cpbm.connector.plan` located at `/usr/share/vts3/connectors` within the section marked by the xml comment `<!--CloudService Connectors-->`.
5. Add the line `<artifact type="bundle" name="com.citrix.cpsm.fragment" version="[<x.x.x>, <x.x.x>]"/>` within the plan tag in the file `com.citrix.cpbm.connector.plan` located at `/usr/share/vts3/connectors` within the section marked by the xml comment `<!--CloudService Connectors-->`.
6. `sudo service cloud-portal start`

Note: x.x.x is the version of the connector.

Enabling the CloudPlatform Services Manager Connector

1. Log in to CloudPortal Business Manager UI as Root user.
2. Go to Home > All Services.
3. Select the CPSM service and click **Enable**.
4. Accept the Terms & Conditions defined by the CPSM cloud service and click **Accept**.
5. Map the roles defined by the CPSM cloud service to the appropriate CloudPortal Business Manager profiles.

Note: The service onboarding for customer and customer's users depends on these role mapping. Ensure that roles are appropriately mapped to profiles.

6. Click **Submit** and then click **Close**.

Connector Service Configuration

The following are the configuration items when adding a CPSM service instance to the connector:

- **Cortex API URL:** CPSM exposes an xml based RESTful API known as the Cortex API. The value provided for this property should be the endpoint URL for the CPSM instance's Cortex API.
- **eCommerce API URL:** In addition to the Cortex API, CPSM exposes another API, called eCommerce API. The value provided for this property should be the endpoint URL for the CPSM instance's eCommerce API. Please note that the Cortex and eCommerce API's offer different capabilities, and are not interchangeable.
- **Base SSO URL:** CPSM exposes a Single Sign On (SSO) URL which the connector can invoke with appropriate parameters to obtain a login token for a specified user. The value provided for this property should be the SSO URL supported by CPSM.
- **CPSM Admin User Name:** The value provided for this property should be the admin username in CPSM.
- **CPSM Admin Password:** The value provided for this property should be the admin password in CPSM.
- **CPSM Location Name:** Every CPSM customer must belong to a location. The value for this property indicates the default CPSM location to be used when creating a customer in CPSM from the connector.
- **Host suffix for shared SSL certificate:** When provisioning a shared SharePoint customer plan, CPSM needs the base URL, relative to which the customer site will be created. This property contains the value of that base URL. This is required only for installations having SharePoint.

Connector User Roles

The connector supports four kinds of roles, which are defined below:

- **CPSM Cloud Management:** This role is available to CloudPortal Business Manager (CPBM) users in the **Service Provider** scope. This role has global admin privileges in CloudPortal Services Manager (CPSM). This is enabled by default for the Service Provider admin.
- **CPSM Ops User:** This role is available to CPBM users in the **Service Provider** scope. This role has ops privileges in CPSM. As part of enabling the connector, the Service Provider admin has to enable this role for the **Ops Admin** profile.
- **CPSM Cloud Account Management:** This role is available to CPBM users in the **Customer** scope. This role has customer admin privileges in CPSM. This role is enabled by default for the **Master User** profile. As part of enabling the connector, the Service Provider admin can enable this role for the **Power User** profile.
- **CPSM Cloud User:** This role is available to CPBM users in the **Customer** scope. This role has normal user privileges in CPSM. As part of enabling the connector, the Service Provider admin has to enable this role for the **User** profile.

Connector Usage Types

The connector collects usage data for the following usage types so that they are billable in CloudPortal Business Manager.

- **Hosted Exchange Customer Plan Usage:**

To create a product for Hosted Exchange Customer Plan and charge for it, select 'Hosted Exchange' as usage type, and use mediation rule to narrow it down to the specific 'Hosted Exchange Customer Plan'. The unit for this usage type is 'Days', which represents the number of days the plan is used by the customer account.

- **Hosted Exchange User Plan Usage:**

To create a product for Hosted Exchange User Plan and charge for it, select 'Hosted Exchange' as usage type, and use mediation rule to narrow it down to the specific 'Hosted Exchange User Plan'. The unit for this usage type is 'Days', which represents the total number of days the plan is used by all users within the customer account.

- **Hosted Apps and Desktops Customer Plan Usage:**

To create a product for Hosted Apps and Desktops Customer Plan and charge for it, select 'Hosted Apps and Desktops' as usage type, and use mediation rule to narrow it down to the specific 'Hosted Apps and Desktops Customer Plan'. The unit for this usage type is 'Days', which represents the number of days the plan is used by the customer account.

- **Hosted Apps and Desktops User Plan Usage:**

To create a product for Hosted Apps and Desktops User Plan and charge for it, select 'Hosted Apps and Desktops' as usage type, and use mediation rule to narrow it down to the specific 'Hosted Apps and Desktops User Plan'. The unit for this usage type is 'Days', which represents the total number of days the plan is used by all users within the customer account.

- **SharePoint 2010 Customer Plan Usage:**

To create a product for SharePoint 2010 Customer Plan and charge for it, select 'SharePoint 2010' as usage type, and use mediation rule to narrow it down to the specific 'SharePoint 2010 Customer Plan'. The unit for this usage type is 'Days', which represents the number of days the plan is used by the customer account.

- **SharePoint 2010 User Plan Usage:**

To create a product for SharePoint 2010 User Plan and charge for it, select 'SharePoint 2010' as usage type, and use mediation rule to narrow it down to the specific 'SharePoint 2010 Customer Plan' and 'SharePoint2010UserPlanId'. The unit for this usage type is 'Days', which represents the total number of days the plan is used by all users within the customer account.

Connector Resource Types

The connector supports three types of resources that can be provisioned from CloudPortal Business Manager (CPBM) to CloudPortal Services Manager (CPSM). They are as follows:

- **Hosted Exchange Customer Plan:**

The connector supports provisioning Hosted Exchange Customer Plan. This resource type has a constraint of 'singleton', which means only one instance of this resource type can be provisioned per account in CPBM.

- **Hosted Apps and Desktops Customer Plan:**

The connector supports provisioning Hosted Apps and Desktops Customer Plan. This resource type has a constraint of 'singleton'.

- **SharePoint 2010 Customer Plan:**

The connector supports provisioning SharePoint 2010 Customer Plan. This resource type has a constraint of 'account', which means only this resource can be provisioned at account level in CPBM. The following are provisioning parameters:

- **SharePoint Instance Name:**

The value for this property must specify the name of the SharePoint 2010 instance to be provisioned. This name is also used to generate the SharePoint 2010 site URL. If 'Shared Hosting' Customer Plan is selected, then the site URL will be `{instanceName}-{code}.{sharedSSLCertHostSuffix}`, whereas if 'Dedicated Hosting' Customer Plan is selected, then the site URL will be `{instanceName}.{domain}`

- **SharePoint Site Template:**

This is a drop down field containing a list of all the templates supported by the underlying SharePoint 2010 service. A template should be selected when provisioning the service. Examples of templates are 'Blog', 'Discussion Group', and so on.

Connector Account Configuration

When master user of an account enables CPSM service in CPBM, the connector creates a CPSM customer and customer admin. The following are configuration parameters for customer and customer admin creation:

- **Username:**

This property should contain the username for the customer admin of the CPSM customer.

Note: CPBM usernames are not used directly as CPSM usernames because it might not meet the CPSM username format. The username string should be between 5 to 11 characters. It can contain characters from a-z (both, upper and lower case), 0-9, and the '_' character.

- **Company domain name:**

This property should contain a value for the customer's domain name (e.g. citrix.com). The domain name specified here is used to create an internal email for the master user (and other users for this customer), as well as the code (within CPSM) for this customer. The string used for the domain must conform to the following regular expression "[a-zA-Z0-9_\\.]{3,8}\\.[a-zA-Z]{2,10}\$". This regular expression specifies that the string must be in the format 'name.ext', where name can be between 3 to 8 characters long (and can contain characters in the range a-z, A-Z, 0-9, and '_'), and the ext can be between 2 to 10 characters long. The ext can only contain characters in the range a-z, A-Z, and 0-9. Again, taking this example further, from the string 'name.ext', the 'name' is chosen as the code for this customer in CPSM. Hence if the domain entered is citrix.com, 'citrix' will become the customer's code and the entire string citrix.com will be used to generate the internal email, as well as the UPN for the users.

- **Password for the customer administrator:**

This property should contain the password for the customer admin of the CPSM customer.

Single Sign On

The connector supports single sign on (SSO) to the CPSM portal. A customer admin can single sign on to the CPSM user interface and provision users and also provision user plans and offerings for the users.

Utilities

This section describes the available Utility tools.

[Active Directory Import Tool](#)

Active Directory Import Tool

Introduction

The Active Directory Import Tool is shipped with CloudPortal Business Manager 2.x and allows enterprise customers to auto-import users from their corporate Active Directory to CPBM. The tool can also be scheduled to run at regular intervals and handle the syncing of the users from Active Directory (AD) to CPBM.

This document covers the following topics:

Setting up the Customization Package
Setting up AD Import Tool
Triggering the AD Import Tool on Demand
Configuring AD CPBM Account Mapping
AD Configuration
AD Import Run Reports

Setting up the Customization Package

Follow the [Customization Guide](#) instructions to import the customization package. The tool needs to be configured as custom job in the customization package
citrix.cpbm.custom.common.

Setting up Active Directory Import Tool

Prerequisites: Follow the instructions in [Active Directory Connection Configuration](#) to configure Active Directory in CloudPortal Business Manager. As a service provider, you must configure `adsync.workdirectory` and `adsync.report_email` with appropriate values.

Note:

- All users created under AD must have email ids configured.
- `ad-import-mapping.csv` must be added to the location mentioned in `adsync.workdirectory`

All setup instructions are to be done in `citrix.cpbm.custom.common/src/main/resources` folder

1. Add this import in Import-Package section in `manifest.mf`

```
com.citrix.cpbm.custom.adsync
```

2. Add this line to `meta-inf/spring/osgi-context.xml`

```
<osgi:reference interface="com.citrix.cpbm.custom.adsync.ADImportService" id="aDImportService" />
```

3. Add these lines to `meta-inf/spring/applicationContext-jobs.xml`

```
<bean name="adImportJob" class="org.springframework.scheduling.quartz.JobDetailBean">
<property name="jobClass" value="com.citrix.cpbm.custom.adsync.ADImportJob" />
<property name="requestsRecovery" value="false" />
<property name="volatility" value="false" />
</bean>
```

```
<bean id="adImportJobTrigger" class="org.springframework.scheduling.quartz.SimpleTriggerBean">
<property name="jobDetail" ref="adImportJob" />
<property name="startDelay" value="10000" />
<property name="repeatInterval" value="3600000" />
<property
name="volatility" value="false" />
<property
name="misfireInstructionName"
value="MISFIRE_INSTRUCTION_RESCHEDULE_NEXT_WITH_EXISTING_COUNT"></property>
</bean>
```

4. Add these lines to `meta-inf/spring/applicationContext-scheduler.xml`

```
<ref bean="adImportJobTrigger" />
```

```
inside <property name="triggers">
<entry key="aDImportService"><ref bean="aDImportService" /></entry>
```

```
inside <property name="schedulerContextAsMap">
```

5. Run maven clean install for the project and drop the `citrix.cpbm.custom.common.x-x-x.jar` inside `vts3/default_impl`

Updating the CPBM plan to include the AD Import tool bundle

1. Add this entry before the `citrix.cpbm.custom.common` entry in the `com.citrix.cpbm.plan` file present in `../vts3/pickup` directory

```
<artifact type="bundle" name="citrix.cpbm.custom.adsync" version="[2.0.0, 3.0.0)"/>
```

Also drop the `citrix.cpbm.custom.adsync-x.x.x.jar` file into `vts3/default_impl` folder

Triggering the Active Directory Import Tool On Demand

Run the `runImportNow.sh` script found in the `util` folder of this tool. This script connects to the CloudPortal Business Manager database and updates the `nextTrigger` time for the AD Import Job.

Note:

- Be sure that cloud-portal service is in running state before running `runImport.sh`
- The script `runImport.sh` must run on the database server of CloudPortal Business Manager.

Configuring the Active Directory CloudPortal Business Manager Account Mapping

A group or OU in Active Directory (AD) maps to the CloudPortal Business Manager (CPBM) tenant account. The account mapping is specified in `ad-import-mapping.csv` file. Be sure about the name of the config file as we use the file name to pick-up the mapping file. Configure the AD work folder in Administration > B/OSS `adsync.workdirectory` field.

Drop the `ad-import-mapping.csv` file in the `{adsync.workdirectory}` folder. Check the `ad-importmapping.csv.sample` in the doc folder for the sample mapping. The CPBM account can be mapped to either an OU in AD or a group in AD.

Active Directory Connection Configuration

Active Directory configuration is to be done by using CloudPortal Business Manager portal.

1. Login as Root user.
2. Navigate to Administration > Configuration > B/OSS Integration.
3. Click **Directory Server > Configure** and set up the Active Directory.

It is not required to set “enable” variable to true.

Mapping attributes from AD to CPBM

The mapping of AD user’s fields to CPBM user’s fields is defined in the Administration > Configuration > B/OSS Integration > Directory Server section. The fields such as name.mapping, email.mapping can be edited to specify the AD to CPBM field mapping.

Also set the handle.to.match.in.directory.server field in the configuration to
(sAMAccountName={ 0 })

sAMAccountName is the principal name/username field for a user in AD.

If you want to use the user’s email address or any other field such as username for the user, replace the sAMAccountName in the handle.to.match.in.directory.server expression with the appropriate AD field name.

Tool Working

For a given CPBM AD Group/OU mapping, the import run:

- Creates a user if it is present in AD Group/OU but not present in CPBM. The following fields from active directory is used while creating the user:
 - login-name/accountName
 - first-name
 - last-name
 - email
 - phone number
 - language preference
- Language preference is set only if the respective AD field contains the locale in the format compatible to the ISO 639 alpha-2 or alpha-3 language code.
- Updates the user fields if the user is present in AD Group/OU as well as CPBM. The fields that are updated are first-name, last-name, email, phone number and language preference.

- If the user is present in CPBM but not in AD Group/OU, the tool gives a warning in the report generated indicating that "Following CPBM users cannot be found in AD". The admin has to take appropriate action on this. Same is the case if the user is moved from an AD group to other.

Active Directory Import Run Reports

Each run of the Active Directory Import Job produces a report in the {adsync.workdirectory}/report folder. Make sure appropriate permissions are set on {adsync.workdirectory} folder so as to be writable by the cloud portal service.

The report is also mailed to the email address configured in adsync.report_email field in under `Active Directory > Configure` section.

Frequently Asked Questions

Setup and Configuration

How to setup redundant CPBM management server?

Both instances will need to be behind a load balancer and a few settings need to be modified in cloud.properties:

```
vmops.jms.url=tcp://x.x.x.x:9125
```

(by default this is set to tcp://localhost:9125, you must change it to the IP of the server)

```
vmops.jms.client.url=failover:(tcp://server1:9125,tcp://server2:9125)
```

(you must add, in the same order on each server, the list of CPBM server(s) that are part of the installation: server1 and server2 can be either FQDNs or IP addresses)

```
vmops.mgmt.server.publicHost=xxx
```

(this is the FQDN/IP on which the load balancer listens for incoming requests)

```
vmops.mgmt.server.publicPort=xxx
```

(this is the port on which the load balancer listens for incoming requests)

```
vmops.mgmt.server.publicProtocol=http/https
```

(this is the protocol on which the load balancer listens for incoming requests)

What functionality breaks if /client is not on the same FQDN as /portal?

If there is a different FQDN for CPBM and CCP, you will run into problems. For example, you are using cpbm.company.com for CPBM and ccp.company.com for CCP. Admin logs into the portal at https://cpbm.company.com/portal. After clicking the Launch Cloud Console button, user is directed to https://ccp.company.com/client. However, CCP will now try to make AJAX calls through the CPBM API proxy which it assumes is at https://ccp.company.com/portal/client. This will fail because /portal is not a valid location on the CCP server.

Which is the location to see the log for a Payment Gateway Charges failure?

The errors can be seen in regular logs such as catalina.out as well as specifically in BillingActivity.log which is present in the same directory as catalina.out . So for your reference you should be able to see it in /tmp/BillingActivity.log.

Can ReCAPTCHA be disabled in CPBM?

Presently, it is not possible to turn this off through configuration or customization.

How to change the Service Provider logo in CPBM?

Change the logo as follows:

1. Set the image path in the global configuration, example: /custom/images
2. Create the folder /custom/images on each CPBM host
3. Login as root user and select customize logo
4. Select the logo that must be used

What is CAS? Why does CPBM need it?

CAS: CAS stands for Central Authentication Service. CAS is an authentication system originally created by Yale University to provide a trusted way for an application to authenticate a user. CAS became a Jasig project in December 2004. CAS provides enterprise single sign on (SSO) service:

- An open and well-documented protocol
- An open-source Java server component
- A library of clients for Java, .Net, PHP, Perl, Apache, uPortal, and others
- Integrates with uPortal, BlueSocket, TikiWiki, Mule, Liferay, Moodle and others
- Community documentation and implementation support
- An extensive community of adopters

For more information, visit <http://www.jasig.org/cas>

What is CAS technically: CAS is an Enterprise Java solution to web application authentication that also provides the benefit of SSO. Technically, SSO can be achieved because the authentication can be removed from the web application and handled centrally. And, when this authentication is handled by a single service, access to many services can be granted once and "remembered" for the life of the web session or even longer though not recommended. Also remember that the SSO feature does not have to be employed and yet CAS authentication still provides a quality authentication mechanism. CAS is a collaborative effort that has produced a very solid central authentication service using the Java development platform. CAS also provides a broad selection of clients for use with all of the enterprise and scripted web technologies. For more information, visit <https://wiki.jasig.org/display/CASUM/Technical+Overview>

Why does CPBM need CAS: CPBM supports the use of CAS for SSO, which enables access to multiple password protected systems after logging in just once.

What languages does CPBM support out-of-the-box (OOB)?

Languages supported by CPBM OOB:

- English, en_US
- Japanese, jp
- German, de_DE

- Simplified Chinese, zh_CN
- Korean, kr
- International Spanish, es
- Brazilian Portuguese, pt_BR
- French, fr

How to change default channel?

1. Go to Configuration > Account Management > default.channel
2. Specify the channel name that you want to make default channel

The existing account will point to the old channel. You do not have to restart the server for reflecting the changes.

Why is the CloudPortal Business Manager not accessible after upgradation from 2.0 to 2.1?

This may happen if the "CloudPortal Start-up Encryption key" is not set prior to upgrade for CPBM 2.0. Perform the following steps to gain access to CPBM 2.1:

1. SSH into the CPBM server
2. Open file /etc/init.d/cloud-portal
3. Search for variable *IS_CLOUD_PORTAL_ENCRYPTED*
4. Change the value of the variable to 1 as *IS_CLOUD_PORTAL_ENCRYPTED=1*
5. Save the file
6. Restart the CPBM using the command `service cloud-portal restart`

Note: This must be done on all the nodes where CPBM is installed.

How to configure CPBM to access internet via proxy address?

1. Go to the path /usr/share/vts3/bin/
2. Edit the dmk.sh file as follows:

```
$JAVA_OPTS \  
-Dhttp.proxyHost=<IP Address of Proxy Host> \  
-Dhttp.proxyPort=8080 \  
-Xbootclasspath/p:$KERNEL_HOME/endorsed/com.springsource.javax.mail-1.4.0.jar \  

```

How to update timezone in CPBM server?

1. Go to the path /usr/share/vts3/bin/
2. Edit the dmk.sh file as follows:

```
$JAVA_OPTS \  
-Duser.timezone=<Update you timezone ID here> \  
-Xbootclasspath/p:$KERNEL_HOME/endorsed/com.springsource.javax.mail-1.4.0.jar \  

```

Note: The timezone id must be a valid timezone ID.

The Tomcat Server from a CPBM host is configured to use a fixed value of 1024MB of RAM. Can I increase the assigned memory, if there is a lot of RAM available?

Yes. You can increase the `-Xmx` to half of the available RAM.

The values you see in the JVM options are default values and is the minimum requirement for the CPBM to run for POC or demo purposes. In production environment higher configurations are recommended.

What is the difference between "Scale up VM" and "Change Service" options?

- **Scale up:** This option enables you to increase the CPU and/or RAM size to change the service offering without stopping the VM

Note: You cannot decrease the CPU and/or RAM size.

- **Change service:** This option enables you to increase or decrease the CPU and/or RAM size to change the service offering only when the state of the VM is stopped

How to setup ActiveMQ accurately in a multinode setup?

It is very important to make sure that ActiveMQ is accurately configured in a multinode setup to prevent issues during CloudPortal startup. When a node gets started

- The ActiveMQ broker on the node comes up first and tries to acquire a lock on the `cloud_portalmq` database. If it acquires the lock then the broker gets started successfully, else it keeps on trying to acquire the lock. ActiveMQ broker uses the following configurations to connect to the `cloud_portalmq` database:

```
# JDBC URL for the MQ database  
vmops.mq.jdbc.url=jdbc:mysql://localhost/cloud_portalmq?relaxAutoCommit=true  
vmops.mq.jdbc.username=cloud  
vmops.mq.jdbc.password=cloud  
mq.jdbc.database.schemaname=cloud_portalmq  

```

When the ActiveMQ broker gets started it is available at the following configuration:

```
# URL to start the message broker at  
vmops.jms.url=tcp://localhost:9125  

```

Note: All the configurations specified earlier are available in the `cloud.properties` file.

- The ActiveMQ listener comes up next and tries to connect to the available brokers. It uses the following configuration to connect to the broker:

```
vmops.jms.client.url=failover:(tcp://cpbmhost1:9125,tcp://cpbmhost2:9125)  

```

This configuration provides all of the broker URLs in the cluster. The listener tries to connect to the broker in the 'failsafe' mode, which means that it tries to connect to first broker and if it is not successful then tries to connect to the second broker.

Note: It is very important to note here that all the nodes must allow other nodes to connect on port 9125. So make sure that nodes are able to connect to each other on port 9125.

In a multinode setup, the node that comes up first gets the lock on the `cloud_portalmq` database and becomes the master broker. The other nodes become slave and keep trying to acquire the lock. If the master broker goes down, one of the slave, which gets the lock on `cloud_portalmq` first, becomes the master broker. JMS listener on all the nodes starts listening to the master broker. When the master broker is down the listener tries to listen to next broker provided in config `vmops.jms.client.url`

Why does CloudPortal Business Manager service take too long to start?

From all the CPBM nodes in a cluster only one node has the ActiveMQ broker running on port 9125. The other nodes keep waiting and can start the ActiveMQ broker only when the first node goes down. So, at any given moment of time only one node has the ActiveMQ broker running on port 9125 and all the other nodes push their messages to this broker.

Account Management

When does an account get suspended/restricted?

An account gets suspended/restricted due to one of the following reasons:

- The SpendLimit is crossed
- The campaign has expired
- The payment failure threshold has reached

How to reset the locked Root user account?

1. Set `failed_login_attempts=0` in users table in `cloud_portal` schema as follows:

```
UPDATE cloud_portal.users SET failed_login_attempts = 0, enabled = 1, locked = 0
WHERE id = 1;
```

2. Open a new browser session (different browser or incognito window) to login

How to reset CPBM root password when it is not the default?

1. If the root email is set properly, reset password will work
2. If the root email is not set properly, run the following SQLquery:

```
update cloud_portal.users set salt='1300869425505AbRaCaDaBrA1943660404',password='b8bdee346e8d4e'
```

Product Management

Can there be bundles that include products from multiple cloud services?

A bundle can be used to provision only one resource. So you cannot have a bundle with multiple resources from across services. However, you can entitle (that is give away as part of) products from other services as part of a bundle. For example, you can create a VM bundle and include a few mailboxes.

How does "This product is replacing one or more products" work?

When selected, this product will not be available in a catalog until the catalog is synced and a new version is activated after that. This is done in conjunction with retiring one or more products as being replaced. This is required because in scenarios where a set of products are replaced by another set, the combined changes have to be automatically visible to the catalog/end customer.

What is the "Sync with Reference Catalog" feature?

The feature syncs the channel catalog with the current plan of the reference catalog. Charges are copied for the new products that come into being. Any new bundles created can be consumed in the channel catalog. Also, retired products are marked as retired in the channel catalog. Since syncing is a modification to the channel catalog plan, the changes will take effect only in the next plan of the channel catalog.

How to remove bundles in CPBM?

The service provider user with "Product create manage" role can unpublish the product bundle under the "Catalog" tab. Unpublishing makes the specific bundle unavailable to the user.

Why a service for a user cannot be enabled?

- If the user does not have appropriate cloud roles enabled, the services does not get enabled for that user
- If the master user of the account has not enabled a cloud service for the account, then the users in the specific account cannot see those services
- Sometimes if the cloud service instance is down/inaccessible while activating the user, the services will not be registered. In that case contact your master user to enable them manually

As a Product Manager, how to preview the catalog before activating the channel?

1. Login as root/product manager
2. Navigate to Channels > Select Channel > Catalog > Next Planned Version/Current/History
3. Choose Preview Catalog option from the gear icon

How to modify prices for bundles and products?

1. Login as root

2. Navigate to Channels > Select Channel > Catalog > Next Planned Version
3. Click Utility Card > Edit Pricing if you want to modify the product price
4. Click Bundle > Edit Pricing if you want to modify the bundle price
5. Click Schedule Activation and specify the new date
6. Select Preview Catalog to verify the changes

How to upload product/bundle images?

1. Create a directory on the CPBM server and provide write permissions
2. Login as root
3. Go to Administration > Portal > Settings
4. Set the directory path

How to enable initial deposit for an account?

1. Login as root user
2. Navigate to the Administration > Account Types
3. Select the Account Type
4. Click on On-boarding Controls > Edit
5. Enable Initial Deposit Required

When does the usage price show zero for a product?

The usage price for a product shows zero if

- The entitlements provided are unlimited/more than usage
- The utility price for the product is set to zero

How to cancel a credit that is issued?

1. Login as root/finance admin
2. Select the account for which credit is issued
3. Navigate to tenant's dashboard
4. Click Payments tab
5. Click Cancel Credit

How to improve the overall performance of the CPBM UI pages?

Customers can potentially improve the overall performance of UI pages by following the recommendations given below:

1. Ensure that the connection is not getting closed in response to HTTP requests made from CPBM pages. This can be verified by using tools such as Firebug and by looking into response headers. The value of header "Connection" should be "keep-alive". If the value of the header is "closed", then check the configuration at reverse proxy server (web server) and make sure that connection is alive.

- Example: (Apache web server)

KeepAlive On

2. Enable compression of response at the reverse proxy server (web server).

- Example: (Apache web server)

SetOutputFilter DEFLATE

AddOutputFilterByType DEFLATE text/html text/css text/plain text/xml text/json application/x-jav