CLOUD CALLING™

VOIPNOW® PROFESSIONAL

VoipNow Automation 2.5.3

Automation Scheduled Tasks

For more information about VoipNow Automation, check:
http://www.4psa.com

Automation Scheduled Tasks

Manual Version 87496.2 at 2012/02/17 12:05:57

For suggestions regarding this manual contact:

docs@4psa.com

Table of Contents

Preface

Who Should Read This Guide

This document must be read by the VoipNow Automation administrators. For a better understanding of the application, the VoipNow Automation User's Guide should be read first.

Chapter 1

The VoipNow Automation Scheduled Tasks

In order for the deployment process to function correctly VoipNow Automation has a number of predefined scheduled tasks, designed to automatically perform a series of actions.

These tasks are set to run after a default schedule that can be easily customized by the system administrator. When a scheduled deployment task is completed a report email notification will be sent to the admin and/or the customer, depending on their email notification preferences set up in the VoipNow Automation control panel.

You can set up the email templates either from the Setup >>Preferences Management page >> Email Templates for the general email templates or from the Setup >>Platform Plug-in Management page >> Email Templates for the email templates required by certain platform plug-ins.

Currently, VoipNow Automation includes the following predefined scheduled tasks:

- The Activate/suspend services scheduled task.
- The Invoices automatic payment scheduled task.
- The Automatic account deployment scheduled task.

- The Automatic invoices generation scheduled task.
- The Provisioning server utilization update scheduled task.
- The Exchange rates update scheduled task.

Note

Before a scheduled task is executed the system checks if another instance of that task is still running. If another instance is found the task to be executed will exit with an error message.

For example, let us assume that the activate/suspend scheduled task runs every 20 minutes. The first activate/suspend task will start at 8:00 A.M and will finish at 8:30 A.M. Normally, at 8:20 A.M a new activate/suspend task should start but because there is another instance of this task it will not start and will return an error message.

If a scheduled task becomes unresponsive then, after an hour, it will be ignored as a running task and the process will start again.

If during a task execution an unresponsive server was found it will be marked for soft delete and it will be ignored the next time the task starts.

The Activate/Suspend Scheduled Task

This scheduled task checks for services that have to be suspended or activated and takes the required actions. It is designed to suspend all the unpaid services and to reactivate those that were previously suspended due to unpaid invoices and were paid in the meantime. This task will also take the appropriate action for the services scheduled for suspension or activation by the system admin.

Note

The specific command line for the **Activate/suspend** task is:

```
/usr/local/voipnow/bin/automation/activate_suspend.sh
```

The task is scheduled to run every eleven minutes. Its location may depend on your installation.

The Activate/suspend scheduled task has two components:

- The Suspension component.
- The Activation component.

The Suspension Component

A service can be suspended manually from VoipNow Automation by the system administrator. He can choose either to suspend only the customer account or to suspend the customer account and its deployed services. For more information on how to manually suspend services or customers, please consult the VoipNow Automation User's Guide.

A service can automatically be suspended because of unpaid invoices. Any product for which there is at least one unpaid invoice with the due date exceeding the limit imposed for the service suspension is going to be scheduled for suspension.

When a new product is configured from the VoipNow Automation's Store items area, it is specified in the Suspension And Termination section after how many hours a product with an overdue invoice will be suspended. Therefor, after the hours that were set up expire the service will be marked for suspension.

Note

Please note that a service is also considered unpaid if the last invoice issued for it is either voided or refunded.

Add-on store items may or may not support suspension depending on their definition. For more details on specific add-ons please check out the VoipNow Automation User's Guide.

Only an **active** service can be suspended.

When the service is suspended, it cannot be accessed or used by anyone.

Before the scheduled task starts the system checks if the entry conditions are satisfied. For this task there is only one condition:

- The license is valid and the allowed number of customers was not exceeded.

  If this requirement is not met, the task exits, displaying an error message. Also, if there is a timeout in the connection protocol request that cannot be processed, the task stops executing, erroring out.

The behaviour of the Suspension component is the following:

1. The task schedules for suspension the unpaid services.
2. The task processes the suspension request.
   - If the suspension process is successful, the task performs the following operations:

    - Changes the services' status to Suspended.
    - Sends email notifications to the system administrator, to the customer and to any other custom email address, according to the specifications customized in the Email Templates page: the Service(s) suspension notification.
- If the suspension process is unsuccessful, the task performs the following operations:
    - Updates the number of failed attempts. The task will retry to suspend these services next time it runs. The number of maximum attempts is set up from the Setup >> Behaviour section >> Orders & Invoices tab. When the number of maximum attempts is reached the task takes out the services from the suspension queue.
    - Sends email notifications to the system administrator, to the customer and to any other custom email address, according to the specifications customized in the Email templates page: the Service deployment-related operation failure notification.

  The suspension process might be unsuccessful due to the following reason:

    - The provisioning server cannot be reached. The server might be inactive, in the Softdown mode or the server could not process successfully the suspension request.

      When the server cannot be reached, the Deployment server unreachable/down notification is sent to the system administrator and the server status is changed to Softdown.

> **Note**
>
> The order status is not affected by any operation performed by this task.

The Activation Component

A service suspended due to unpaid invoices is automatically queued for activation if the overdue invoices are paid.

The system administrator can choose to manually activate the services he previously suspended when the customer account and its deployed services were suspended. For more information on how to manually suspend services or customers please consult the VoipNow Automation User's Guide.

Before the scheduled task starts the system checks if the entry conditions are satisfied. For this task there is only one condition:

- The license is valid and the allowed number of customers was not exceeded.

  If this requirement is not met, the task exits, displaying an error message. Also, if there is a timeout in the connection protocol request that cannot be processed, the task stops executing, erroring out.

The behaviour of the Activation component is as follows:

1. The task schedules for activation the paid services that were suspended due to unpaid invoices.
2. The task processes the service activation request.
   - If the activation process is successful, the task performs the following operations:
     - Changes service status to Activated.
     - Sends email notifications to the system administrator, to the customer and to any other custom email address, according to the specifications customized in the Email templates page: the Service(s) reactivation notification.
   - If the activation process is unsuccessful, the task performs the following operations:
     - Updates the number of failed attempts. The task will retry to activate these services next time it runs. The number of maximum attempts is set up from the Setup >> Behaviour section >> Orders & Invoices tab. When the number of maximum attempts is reached the task takes out the services from the activation queue.
     - Sends email notifications to the system administrator, to the customer and to any other custom email address, according to the specifications customized in the Email templates page: the Service deployment-related operation failure notification.

     The activation process might be unsuccessful due to the following reason:

     - The provisioning server cannot be reached. The server might be inactive, in the Softdown mode or the server could not process successfully the activation request.

     When the server cannot be reached, the Deployment server unreachable/down notification is sent to the system administrator and the server status is changed to Softdown.

After the task has finished running, the system administrator receives the Activate/suspend report notification. This notification provides details about all the customers' suspended and/or activated services.

Note

The order status is not affected by any operation performed by this task.

The Invoices Auto Payment Scheduled Task

This task automatically charges the unpaid invoices belonging to clients that have active subscriptions. The payment plug-in used for setting up a subscription must be within the ones configured by the system administrator in the Behavior section, under the Payment Preferences tab.

Caution

The **Invoices Auto Payment** task will not take any payment actions if:

- The order placed from VoipNow Automation was created with the **customer will process payment** option enabled. The invoice issued for this order can be later paid either by the customer from the public store, or by the system administrator, from the VoipNow Automation control panel.

  A new order can be manually added from the Orders >> **Order Management** page.

Note

The specific command line for the **Invoices auto payment** task is:

```
/usr/local/voipnow/bin/automation/auto_payment.sh
```

The task is scheduled to run daily, at 7:30. Its location may depend on your installation.

Before the task starts the system verifies if the entry conditions are satisfied. These conditions are:

- The license is valid and the allowed number of customers was not exceeded.
- The Automatic payment option must be enabled from the Setup >> Behaviour section >> Payment Preferences tab.
- There are payment processors configured and set up in the Setup >> Behaviour >> Payment Preferences tab.

  If these requirements are not met, the task exits, displaying an error message. Also, if there is a timeout in the connection protocol request that cannot be processed, the task stops executing, erroring out.

The behaviour of the Invoices Auto Payment task is the following:

1. The task fetches all the unpaid invoices belonging to customers with defined and active subscriptions that meet the following requirements:
   - The invoice status is Purchase (new) or Purchase (recurrent).
   - The invoice payment status is Pending.
2. For each invoice, the task verifies if the maximum number of payment attempts was reached.
   - If it has been reached, the task does not initiate any actions.
   - If it has been not reached, the task tries to pay the invoice using a payment gateway.

   > Note
   >
   > The customer is able to choose in the **Online store** >> **Payment Information** page the preferred payment method for the automatic payment and he can also set up a default subscription.

   The algorithm the task runs when selecting a payment method is explained below:

   a. The task verifies if a preferred payment method exists.
      - If the customer has defined a preferred payment method for the payment gateway (credit card processor or virtual payment system) in use, the task attempts to pay the invoices using this subscription.

        Otherwise the task looks after active subscriptions for the payment gateway in use and tries to pay the invoices using an active subscription.

   b. If no preferred payment method exists the task choses an active subscription for any payment gateway (credit card processor or virtual payment system) in use. If the selected payment processor has a default subscription the invoices are paid using this subscription.

   c. If, for any payment gateway, no active subscriptions are found the invoice will not be paid.
3. The corresponding email notifications are sent:
   - If the invoice payment is successful, the Automatic invoice payment notification is sent.

- If the invoice could not be paid and this was not the last permitted payment attempt, the Automatic invoice(s) payment failure notification is sent.
- If the invoice could not be paid and this was the last permitted payment attempt, the Automatic invoice(s) payment final failure notification is sent.

  The customer is informed that the payment must be done manually. For an easy payment process an URL is provided in the email with a direct location to the online store payment area.
- The task notifies the customers who do not have subscriptions about all the invoices that are not paid by sending them an email with the New pending invoice(s) automatic payment notification.
- When the task is completed, the system administrator is sent the Automatic charging report notification. This notification offers an overview of the entire automatic payment process, including information about all the invoices that were automatically paid or those for which the automatic payment could not be successfully completed.

The Automatic Service Deployment Scheduled Task

This task deploys all services with paid invoice. The system administrator can also schedule services for deployment from the Orders Deployment Management page. The Automatic Service Deployment schedule task periodically tries to deploy all the services available in the deployment queue.

A service is scheduled for deployment if the following conditions are fulfilled:

- All corresponding invoices are paid:
  - If the payment was made by wire transfer or by check, the invoice payment status should be Captured.
  - If the payment was made by credit card, the invoice payment status should be Authorized or Captured.

  Note

  A service is considered unpaid if the last invoice is voided or refunded.
- The status of the corresponding order is New or Partially fulfilled and the customer account is Active.

- Master service must be deployed, active or suspended in the case of slave services.

> Note
>
> The specific command line for the **Automatic service deployment** task is:
>
> `/usr/local/voipnow/bin/automation/deploy_account.sh`
>
> The task is scheduled to run every 10 minutes. Its location may depend on your installation.

Before the task starts the systems verifies if the entry conditions are satisfied. For this task there is only one condition:

- The license is valid and the allowed number of customers was not exceeded.

  If this requirement is not met, the task exits, displaying an error message. Also, if there is a timeout in the connection protocol request that cannot be processed, the task stops executing, erroring out.

The behaviour of the Automatic Service Deployment task is the following:

1. The task fetches the services scheduled for deployment.
2. The task searches all active provisioning servers and computes the server load and score using the formulas:
   - (Load) `L = server.<deployment_type>.utilization/server.<deployment_type>.limit` where:

     > Note
     >
     > For example, if we have a server with the `server.extension.utilization` = 50 and the `server.extension.limit` = 100, then the load is **L = 50/100 = 0.5**.

     - `server.<deployment_type>.utilization` represents the number of accounts of a certain `<deployment_type>` (reseller, client, conference, etc.) currently available on the provisioning server.
     - `server.<deployment_type>.limit` represents the maximum number of accounts that the deployment server can host.
   - (Score) `S = L*weight*100` where:

> Note
>
> For example, if the server's `weight` is 1 and `L` is the one computed before, 0.5, than the server's score is **S = 0.5*1*100 = 50**.

- `L` represents the server Load, computed using the formula above.
- `Weight` represents the server's `weight`

3. For each service scheduled for deployment, the application follows the steps described in the following sections.

> Note
>
> Currently, only one item family is supported: **VoipNow Professional account**.

4. The same steps are repeated until all the services scheduled for deployment are checked by the `Automatic service deployment` scheduled task.
5. When the task is completed the system administrator is sent the `Order deployment report notification`. This notification provides details about all the services that have been successfully deployed, as well as about the services that could not be deployed due to certain reasons.

Master Service

The task sorts the valid provisioning servers by score, taking into consideration the services' deployment type (like `Reseller`, `Client` or `Extension`). A service is deployed on the server with the lowest score if the following conditions are met:

- The server utilization + one new deployed service is not higher than the maximum utilization allowed for that server.

> Note
>
> If the first provisioning server is not valid the task will then try to deploy the service on the next valid server. If no valid server is found the task stops displaying an error message.

1. If a valid deployment server is found at the previous step, the service deployment action may be initialized on that server. The following steps are performed:

a. The task verifies if the charging plan it tries to add exists and is enabled on the provisioning server. For that, the following actions are performed:

   - When a charging plan is deployed, it will be signed on the provisioning server with a hash made up from its content. The task searches for this signature and checks the one from the provisioning server against the signature from the building brick.
   - The scheduled task checks if
     - the service was previously deployed on the same provisioning server
     - the service was not deleted
     - the charging plan was not changed on the provisioning server or on the VoipNow Automation server
     - the charging plan exists in the parent account's context the service will be deployed under

     If this requirements are met the task deploys the service using the same charging plan as from the previously deployed service.

     > Note
     >
     > In order for an existing charging plan to be used for deployment it must not be altered in VoipNow Automation or on the provisioning server. An altered charging plan is marked on the provisioning server an the task does not use this charging plan for deployment.

   - If no charging plan was found a new one will be created under the parent account the service will be deployed. This new charging plan takes all the information from the charging plan related to the service.

b. The task will create the new user account (Reseller, Client or Extension) using the information from the customer's contact information. If there is a duplicate login or minor constraints the task uses the force method to fill in the new account information. If no information is found the data will be random generated.

   The charging plan will be associated with the new account.

   > Note
   >
   > For some types of services (**Client** or **Extension**) if the account under which the deployment must be done is not created, an error message will inform the system administrator about this outcome.

c. The task will set up the account using the template specified for that cloud building brick:

- For the Reseller and Client building brick, only the permissions and limits are set.
- For the Extension building brick, the extension is also set up. If the sounds used by the extension template exist on the provisioning server the new account will be set up with this sounds. Otherwise the default sounds are used.

d. Any slave services linked to this master service will be deployed as well. A new charging plan identical to the master service is going to be deployed under the newly created account on the provisioning server. Each slave service will use by default this charging plan. It can be later on modified on the respective provisioning server. The slave service will follow the same rules as a master service in the deployment process. If the deployed service is configured with DIDs, these will be assigned from the master service's poll.

e. Any add-on services attached to this master service will be deployed as well.

f. If the account is created and fully configured, then the Automatic Service Deployment task:

- Sets the order status to Fulfilled or Partially fulfilled.
- Assigns DIDs to the newly created account, if the Direct Inward Dialing numbers exist and if the deployed service is configured to use them.

> Note
>
> The **Automatic service deployment** scheduled task will only assign the possible number of DIDs as found on the server.
>
> For example, let assume that when you added a new product in **VoipNow Automation** you set the **Assign `{value}` DIDs from the server pool** field to 5, but the server has only 2 available DIDs. In this case the task will assign only 2 DIDs and will inform the system administrator about this outcome.

- Goes directly to Step 6.

> Note
>
> These operations are performed independent of the result on add-on and slave service(s) deployment.

g. If one of the steps from i to iii fails, than the task rolls back, removing from the provisioning server all the service's components that have previously been put in place.

   When the Maximum number of operation attempts is reached the task will eliminate the services for which the deployment has failed.

> **Note**
>
> The **Maximum number of operation attempts** can be setup up from the Setup >> **Behaviour** section, under the **Order & Invoices** tab.

2. If no eligible server is found (but active servers exist) or if the service deployment failed, a message is displayed informing the administrator about this situation.
3. When a service is successfully deployed, the task sets the status of the orders to Fulfilled if all the services on the order were successfully deployed or to Partially fulfilled if only a number of services were successfully deployed.

Independent Slave Service

An independent slave service is that whose master service was deployed in a previous execution of these task. For some reason, the slave service could not be deployed along with the master and a further attempt to fulfill the task is being made.

1. If the provisioning server on which the master service has been deployed is enabled, the service deployment action may be initialized on that server. The following steps are performed:

   a. The task verifies if the charging plan it tries to add exists and is enabled on the provisioning server. For that, the following actions are performed:

      - When a charging plan is deployed, it will be signed on the provisioning server with a hash made up from its content. The task searches for this signature and checks the one from the provisioning server against the signature from the building brick.
      - The scheduled task checks if
        - the service was previously deployed on the same provisioning server
        - the service was not deleted
        - the charging plan was not changed on the provisioning server or on the VoipNow Automation server

    - the charging plan exists in the parent account's context the service will be deployed under

  If this requirements are met the task deploys the service using the same charging plan as from the previously deployed service.

> Note
>
> In order for an existing charging plan to be used for deployment it must not be altered in VoipNow Automation or on the provisioning server. An altered charging plan is marked on the provisioning server an the task does not use this charging plan for deployment.

  - If no charging plan was found a new one will be created under the parent account the service will be deployed. This new charging plan takes all the information from the charging plan related to the service.

b. The task will create the new user account (`Client` or `Extension`) using the information from the customer's contact information. If there is a duplicate login or minor constraints the task uses the force method to fill in the new account information. If no information is found the data will be random generated.

   The charging plan will be associated with the new account.

c. The task will set up the account using the template specified for that cloud building brick:

   - For the `Client` building brick, only the permissions and limits are set.
   - For the `Extension` building brick, the extension is also set up. If the sounds used by the extension template exist on the provisioning server the new account will be set up with this sounds. Otherwise the default sounds are used.

d. If the account is created and fully configured, then the `Automatic Service Deployment` task:

   - Assigns `DIDs` to the newly created account, if the Direct Inward Dialing numbers exist and if the deployed service is configured to use them.

> Note
>
> The **Automatic service deployment** scheduled task will only assign the possible number of DIDs as found on the server.
>
> For example, let assume that when you added a new product in **VoipNow Automation** you set the **Assign `{value}` DIDs from the server pool** field to 5, but the server has only 2 available DIDs. In this case the task

will assign only 2 DIDs and will inform the system administrator about this outcome.

> **Note**
>
> These operations are performed independent of the result on add-on and slave service(s) deployment.

e. If one of the steps from i to iii fails, than the task rolls back, removing from the provisioning server all the service's components that have previously been put in place.

   When the Maximum number of operation attempts is reached the task will eliminate the services for which the deployment has failed.

> **Note**
>
> The **Maximum number of operation attempts** can be setup up from the Setup >> **Behaviour** section, under the **Order & Invoices** tab.

2. If the service deployment failed, a message is displayed informing the administrator about this situation.

Independent Add-on Service

An independent add-on service is that whose master service was deployed in a previous execution of these task. For some reason, the add-on service could not be deployed along with the master and a further attempt to fulfill the task is being made.

1. If the provisioning server on which the master service has been deployed is enabled, the service deployment action may be initialized on that server. The following steps are performed:

   a. The task will deploy the add-on service under the master. For this precise item family, the master account's credit for incoming as well as outgoing calls will be increased with a certain amount of money.

      When the Maximum number of operation attempts is reached the task will eliminate the services for which the deployment has failed.

> **Note**
>
> The **Maximum number of operation attempts** can be setup up from the Setup >> **Behaviour** section, under the **Order & Invoices** tab.

2. If the service deployment failed, a message is displayed informing the administrator about this situation.

The Automatic Recurrent Invoices Generation Scheduled Task

In VoipNow Automation, there are services that require a recurrent fee to be paid at a certain date. For these services, an invoice must be issued for every period of time (monthly, quarterly or yearly), at the date specified in the Setup >> Behaviour section, under the Order & Invoices tab. The Issue invoices on option allows the system administrator to set the day of the month when the recurrent invoices should be issued. This process is handled by the Automatic Recurrent Invoices Generation scheduled task. Besides this, the task is responsible for destroying all the items that have exceeded the termination period, or those that have been scheduled for termination by the system administrator. When an item is terminated the task will generate an invoice that accounts the consumption and the unpaid fees with the exception of slave services which are not invoiced.

Invoices are generated for both active or suspended products.

The next section describes how the task handles the recurrent invoice generation process for every situation that might appear.

> Caution
>
> Have in mind that the time zone used by the provisioning servers may be different than the one set up on the VoipNow Automation server! This fact must be taken into consideration when verifying the calculated consumption for a certain time period.

- The first case describes the invoice generation process for when the deployment is successfully fulfilled on the date of the purchase, and the tolerance interval is not exceeded:

  Purchase Date: 10th October

      Service charge between 10th October - 9th November

  Deployment Date: 10th October
  Tolerance interval: 10 days
  Issue invoice on: 3rd of every month

  Invoice Generation Date: 3rd November
  Charge for:

      Service between 10th November - 9th December

```
          Consumption between 10th October - 2nd November
```

Invoice Generation Date: 3rd December
Charge for:

```
          Service   between  10th December - 9th January
          Consumption  between   3rd November - 2nd December
```

Invoice Generation Date: 3rd January
Charge for:

```
          Service between  10th January - 9th February
          Consumption between 3rd December - 2nd January
```

- The second case describes the invoice generation process for when the deployment is successfully fulfilled on the date of the purchase, and the tolerance interval is exceeded:

Purchase Date: 10th October

```
    Service charge between 10th October - 9th November
```

Deployment Date: 10th October
Tolerance interval: 5 days
Issue invoice on: 3rd of every month

Invoice Generation Date: 3rd November
No invoice is generated because on 3rd November we already have
a number of days greater than the tolerance interval paid in advance.
Next recurrent invoice is issued on 3rd December.

Invoice Generation Date: 3rd December
Charge for:

```
          Service between 10th November - 9th December
          Consumption between 10th October - 2nd December
```

Invoice Generation Date: 3rd January
Charge for:

```
          Service between  10th December - 9th January
          Consumption between 3rd December - 2nd January
```

- The next case describes the invoice generation process for when the deployment is fulfilled after the date of the purchase, and the tolerance interval is not exceeded:

Purchase Date: 10th October

```
    Service charge between 10th October - 9th November
```

Deployment Date: 18th October
Tolerance interval: 10 days
Issue invoice on: 9th of every month

Invoice Generation Date: 9th November
Charge for:
Service between 10th November - 17th December
Consumption between 10th October - 8th November

> Note
>
> When calculating the service period for this invoice the difference between the purchase date and the deployment date is added. In our case, 9th December + 8 days already paid for last month = 17th December

Invoice Generation Date: 9th December
Charge for:
Service between 18th December - 17th January
Consumption between 9th November - 8th December

Invoice Generation Date: 9th January
Charge for:
Service between 18th January - 17th February
Consumption between 9th December - 8th January

- The next case describes the invoice generation process for when the deployment is fulfilled after the date of the purchase, and the tolerance interval is exceeded:

Purchase Date: 10th October

Service charge between 10th October - 9th November
Deployment Date: 18th October
Tolerance interval: 10 days
Issue invoice on: 3rd of every month

Invoice Generation Date: 3rd November
No invoice is generated because on 3rd November we already have
a number of days greater than the tolerance interval paid in advance.
Next recurrent invoice is issued on 3rd December.

Invoice Generation Date: 3rd December
Charge for:

Service between 10th November - 17th December
Consumption between 10th October - 2nd December

Note

When calculating the service period for this invoice the difference between the purchase date and the deployment date is added. In our case, 9th December + 8 days already paid for last month = 17th December

Invoice Generation Date: 3rd January
Charge for:
Service between 18th December - 17th January
Consumption between 3rd December - 2nd January

- The next case describes the invoice generation process for when the invoices cannot be issued on time and the tolerance interval is not exceed.

Purchase Date: 10th October

Service charge between 10th October - 9th November
Tolerance interval: 10 days
Issue invoice on: 9th of every month

Invoice Generation Date: 9th November
Charge for:
Service between 10th November - 9th December
Consumption between 10th October - 8th November

Invoice Generation Date: 3rd December
Assuming that, for an unknown reason the invoice could not be issued, the task will attempt to generate the recurrent invoice in each of the following day until the invoice is successfully issued.
Let us assume that the invoice is generated on 10th December.

Invoice Generation Date: 10th December
Charge for:
Service between 10th December - 9th January
Consumption between 9th November - 9th December

Invoice Generation Date: 9th January
Charge for:
Service between 10th January - 9th February
Consumption between 10th December - 8th January

- The next case describes the invoice generation process for when the invoices cannot be issued on time and the tolerance interval is exceed.

  Purchase Date: 10th October

  Service charge between 10th October - 9th November

  Tolerance interval: 5 days

  Issue invoice on: 3rd of every month

  Invoice Generation Date: 3rd November

  No invoice is generated because on 3rd November we already have a number of days greater than the tolerance interval paid in advance.

  Next recurrent invoice is issued on 3rd December.

  Invoice Generation Date: 3rd December

  Assuming that, for an unknown reason the invoice could not be issued, the task will attempt to generate the recurrent invoice in each of the following day until the invoice is successfully issued.

  Let us assume that the invoice is generated on 4th December.

  Invoice Generation Date: 4th December

  Charge for:

  Service between 10th November - 9th December

  Consumption between 10th October - 3rd December

  Invoice Generation Date: 3rd January

  Charge for:

  Service between 10th December - 9th January

  Consumption between 4th December - 2nd January

- The next case describes the invoice generation process for when the deployment is fulfilled before the first recurrent invoice generation date, on the same month.

  Purchase Date: 10th October

  Service charge between 10th October - 9th November

  Deployment Date: 10th October

  Tolerance interval: 5 days

  Issue invoice on: 15th of every month

  Invoice Generation Date: 15th October

  No invoice is generated because on 15th October we already have a number of days greater than the tolerance interval paid in advance.

Next recurrent invoice is issued on 15th November.

Invoice Generation Date: 15th November
Charge for:
Service between 10th November - 9th December
Consumption between 10th October - 14th December

Invoice Generation Date: 15th December
Charge for:
Service between 10th December - 9th January
Consumption between 15th November - 14th December

Invoice Generation Date: 15th January
Charge for:
Service between 10th January - 9th February
Consumption between 15th December - 14th January

- The next case describes the invoice generation process for when the invoice generation date is between the date of purchase and the deployment date.

Purchase Date: 10th October

Service charge between 10th October - 9th November
Deployment Date: 18th October
Tolerance interval: 10 days
Issue invoice on: 15th of every month

Invoice Generation Date: 15th November
Charge for:
Service between 10th November - 17th December
Consumption between 10th October - 14th November

Invoice Generation Date: 15th December
Charge for:
Service between 18th December - 17th January
Consumption between 15th November - 14th December

Invoice Generation Date: 15th January
Charge for:
Service between 18th January - 17th February
Consumption between 15th December - 14th January

> **Note**
>
> The specific command line for the **Automatic recurrent invoices generation** task is:
>
> ```
> /usr/local/voipnow/bin/automation/generate_invoices.sh
> ```
>
> The task is scheduled to run daily, at 6:45. Its location may depend on your installation.

Before the task starts the system verifies that the entry conditions are satisfied. For this task there is only one condition:

- The license is valid and the allowed number of customers was not exceeded.

  If this requirement is not met, the task exits, displaying an error message. Also, if there is a timeout in the connection protocol request that cannot be processed, the scheduled task stops executing, erroring out.

The behaviour of the Automatic recurrent invoices generation task is as follows:

1. The task checks if recurrent invoices should be generated at current date. If the invoice date is set to 31, 30 or 29 and that value is not possible for the current month, the next lowest value is used.

   Example:

   In February (when the month has only 28 days), if the generation date is set to the 31st, the invoice will be generated on the 28th of February.

   > **Note**
   >
   > The system administrator can choose the date when the recurrent invoices are issued using the **Issue invoices on** option available in the **Invoices & Orders** tab.

   If recurrent invoices should be generated at the current date the task will take the appropriate action.

2. The task schedules for termination all the services that exceed the termination interval (invoice due date + terminate interval < current date) and generates the invoices for those services. These invoices will account the consumption and the amount due to unpaid subscription fees. The amount due is computed for the period between the last invoice and the

termination date, that corresponds to the time interval the service was used by the customer.

Note

When an item is destroyed it is removed from the provisioning server. An item is scheduled for destruction when the termination period expires. The termination period is configured from the Store items >> **Add New Item** page >> **Suspension And Termination** section. In the *Destroy item if payment is due for* field specify after how many hours the item is terminated.

The examples below explains how the task processes the invoices for the services scheduled for termination:

The first example describes the termination period for when the tolerance interval is not exceeded.

Purchase Date: 10th October

Service charge between 10th October - 9th November
Tolerance interval: 10 days
Issue invoice on: 3rd of every month
Generated invoices are due in: 5 days
Suspend item if payment is due for: 72 hours (3 days)
Destroy item if payment is due for: 144 hours (6 days)

Invoice Generation Date: 3rd November
Charge for:
Service between 10th November - 9th December
Consumption between 10th October - 2nd December

If this invoice is not paid until 11th November (3rd November + 5 days (from generated invoices are due in) + 3 days (from suspend item if payment is due for)) the Activate/suspend scheduled task will suspend the item. If this invoice is not paid until 14th November (3rd November + 5 days (from generated invoices are due in) + 6 days (from destroy item if payment is due for)) the Automatic recurrent invoices generation scheduled task will terminate the item and issue a new invoice:

Invoice Generation Date: 14th November
Charge for:
Consumption between 3rd November - 14th November

The second example describes the termination period for when the tolerance interval is exceeded.

Purchase Date: 10th October

Service charge between 10th October - 9th November
Tolerance interval: 5 days
Issue invoice on: 3rd of every month
Generated invoices are due in: 5 days
Suspend item if payment is due for: 72 hours (3 days)
Destroy item if payment is due for: 144 hours (6 days)

Invoice Generation Date: 3rd November
No invoice is generated because on 3rd November we already have
a number of days greater than the tolerance interval paid in advance.
Next recurrent invoice is issued on 3rd December.

Invoice Generation Date: 3rd December
Charge for:
Service between 10th November - 9th December
Consumption between 10th October - 2nd December

If this invoice is not paid until 11th December (3rd December + 5 days (from generated invoices are due in) + 3 days (from suspend item if payment is due for)) the Activate/suspend scheduled task will suspend the item. If this invoice is not paid until 14th December (3rd November + 5 days (from generated invoices are due in) + 6 days (from destroy item if payment is due for)) the Automatic recurrent invoices generation scheduled task will terminate the item and issue a new invoice:

Invoice Generation Date: 14th December
Charge for:
Service between   10th December - 9th January
Consumption between 3rd December - 14th December

Note

If a customer has an invoice with multiple services and only some of them exceed the termination interval, the task schedules for termination only those services, not the entire order.

> For example, if a client has three services on an invoice and only two exceed the termination interval, the task will schedule for termination only these two services.

3. The task checks if there are invoices that were not generated in the past and if any are found it will take the appropriate action. The invoices are always generated, if possible, at the date set up in the interface. If, for some reasons, the task was not run, it will account the consumption and it will issue all the required invoices the first time it will run.

   ```
   Purchase Date: 10th October

       Service charge between 10th October - 9th November
   Tolerance interval: 10 days
   Issue invoice on: 9th of every month

   Invoice Generation Date: 9th November
   Charge for:
             Service between 10th November - 9th December
             Consumption between 10th October - 8th November

   Invoice Generation Date: 3rd December
   Assuming that, for an unknown reason the invoice could not be issued,
   the task will attempt to generate the recurrent invoice in each of
   the following day until the invoice is successfully issued.
   Let us assume that the invoice is generated on 10th December.

   Invoice Generation Date: 10th December
   Charge for:
             Service between  10th December - 9th January
             Consumption between 9th November - 9th December

   Invoice Generation Date: 9th January
   Charge for:
              Service between  10th January - 9th February
              Consumption between 10th December - 8th January
   ```

4. For each customer, the task follows these rules:

   - Each invoice is generated using the currency set up in the customer's default billing details.
   - For each service, no matter to which order it belongs, the default billing details are used.

- The VAT used for all the invoices is, as well, the one defined in the customer's default billing details.

For each service added to an invoice, the following actions are performed:

- The task fetches the consumption.
- If the service is scheduled for termination, the task will initiate its removal. For the currently supported family item, VoipNow Professional account, when a service is terminated the task will attempt to destroy the charging plan associated with that service but only if the charging plan is not used for another service.
- If the consumption could not be fetched, the service is added to the Invoices generation report notification that is sent to the system administrator, under the *Failed to fetch resources* category.

  The service is not added to the invoice sent to the customer.
- If the consumption could not be fetched and the service is scheduled for termination, the service is added to the Invoices generation report notification that is sent to the system administrator, under the *Failed to terminate* category.

  The service is not added to the invoice sent to the customer.
- If the consumption could be fetched and the service is scheduled for termination, the service is added to the Invoices generation report notification that is sent to the system administrator, under the *Terminated* category.

  The service is added to the invoice sent to the customer. The order that contains the service is terminated.
- The successfully generated recurrent invoices are added to the Invoices generation report notification that is sent to the system administrator, under the *Successfully generated* category.
- The Service(s) termination notification is sent to every customer that has products terminated by the Automatic Invoices Generation scheduled task.
- The task notifies the customers about the new generated invoices using two notifications:
  - The New pending invoice(s) automatic payment notification if the customer has a defined subscription set up in VoipNow Automation .
  - The New pending invoice(s) manual payment notification if the customer does not have any subscriptions defined in the system and has to manually make the payment.

Note

This task generates **Purchase (recurrent)** type invoices.

The Exchange Rates Update Scheduled Task

This task updates the exchange rates of all the currencies used in VoipNow Automation. The task runs automatically on a daily basis and every time the system administrator wants to manually update the exchange rates.

If the system administrator wants to obtain the latest exchange rates, he must click the Automatically update exchange rates icon available in the Setup >> Currency exchange >> Currency Exchange Rates History page.

Note

The specific command line for the **Exchange rates update** task is:

```
/usr/local/voipnow/bin/automation/update_exchange.sh
```

The task is scheduled to run daily, at 1:15. Its location may depend on your installation.

Before the task starts the system verifies if the entry conditions are satisfied. For this task there is only one entry condition:

- The license is valid and if the allowed number of customers was not exceeded.

  If this requirement is not met, the task exits, displaying an error message. Also, if there is a timeout in the connection protocol request that cannot be processed, the task stops executing, erroring out.

The behaviour of the Exchange Rates Update task is as follows:

1. To obtain the latest exchange rates, the task contacts a specialized third party that provides real time updates for all the available currencies.

   Note

   The exchange rate is automatically updated only for those currencies that were configured with the **Update exchange daily** option enabled. If this check box is not selected, the exchange rates can only be manually updated, by clicking the **Automatically update exchange rates** icon.

2. The task downloads a .csv file with the latest exchange rates and then updates the currency values in the VoipNow Automation database.

Note
Exchanged rates for removed currencies are kept as historical records.

The Provisioning Server Utilization Update Scheduled Task

This task updates the provisioning servers' utilization, providing details about their load level.

There is a limit for every account type hosted by a provisioning server, used by the Provisioning Server Utilization Update task to calculate the current load. This limits are configured by the system administrator during the provisioning server setup.

Note

For the currently supported family item, **VoipNow Professional account** the utilization of each account type is given by the number of current accounts deployed on the server.

The task updates the server utilization automatically and every time the system administrator requires it.

The system administrator can trigger the execution of the Provisioning Server Utilization Update task by clicking the Refresh link available in the Orders deployment >> Provisioning servers page.

Note

The specific command line for the **Provisioning server utilization update** task is:

```
/usr/local/voipnow/bin/automation/getup2date.sh
```

Through this command line the task fetches the information regarding the number of resellers, clients, and extensions' type. Then the server utilization information is updated with the new figures. Keep in mind that the information the task fetches is only available for the currently supported family item **VoipNow Professional account**.

The task is scheduled to run daily, every 18 minutes. Its location may depend on your installation.

Before the task starts the system verifies if the entry conditions are satisfied. For this task there is only one entry condition:

- The license is valid and if the allowed number of customer was not exceeded.

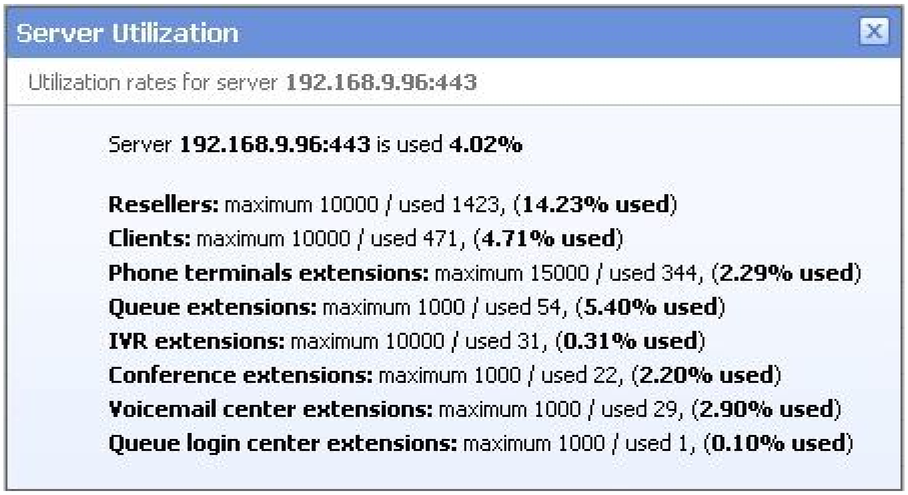
  If this requirement is not met, the task exits, displaying an error message. Also, if there is a timeout in the connection protocol request that cannot be processed, the task stops executing, erroring out.

The Provisioning Server Utilization Update task behaviour is as follows:

1. The task requests the provisioning server its updated utilization:
   - If the deployment server does not answer, the Provisioning Server Utilization Updater scheduled task updates the server's status to Softdown.
   - If the deployment server answers, the task refreshes the information available in the database with the new figures.

> Note
>
> When the system administrator clicks the Refresh link to update the server's utilization, a pop-up panel is displayed, providing the following statistics:
>
> - Server **{hostname}** is used {utilization}
> - **Resellers:** maximum {x}/ used {y} **{z% used}**
> - **Clients:** maximum {x}/ used {y} **{z% used}**
> - **Phone terminals extensions:** maximum {x}/ used {y} **{z% used}**
> - **Queue extensions:** maximum {x}/ used {y} **{z% used}**
> - **IVR extensions:** maximum {x}/ used {y} **{z% used}**
> - **Conference extensions:** maximum {x}/ used {y} **{z% used}**
> - **Voicemail center extensions:** maximum {x}/used {y} **{z% used}**
> - **Queue login center extensions:** maximum {x}/used {y} **{z% used}**

The pop-up panel will look like this:



Server Utilization

Utilization rates for server **192.168.9.96:443**

Server **192.168.9.96:443** is used **4.02%**

Resellers: maximum 10000 / used 1423, (**14.23% used**)
Clients: maximum 10000 / used 471, (**4.71% used**)
Phone terminals extensions: maximum 15000 / used 344, (**2.29% used**)
Queue extensions: maximum 1000 / used 54, (**5.40% used**)
IVR extensions: maximum 10000 / used 31, (**0.31% used**)
Conference extensions: maximum 1000 / used 22, (**2.20% used**)
Voicemail center extensions: maximum 1000 / used 29, (**2.90% used**)
Queue login center extensions: maximum 1000 / used 1, (**0.10% used**)

The server utilization panel for the currently supported family item VoipNow Professional account

2. The refreshed server load is displayed in the Provisioning Server Management page, under the Utilization column (e.g. 1.73%).

Chapter 2

Contact and Support

For online help and support please visit:

- Support Zone: https://help.4psa.com
- Knowledge Base: http://kb.4psa.com
- Documentation: http://help.4psa.com/docs/

For mailing addresses and phone numbers from our offices:

http://www.4psa.com/contactus

If you have any question, do not hesitate to contact us.