

Universal Module

Table of Contents

- 1 About The Universal Module
- 2 Using the Universal Module
 - 2.1 Installing the Universal Module
 - 2.2 Products
 - 2.3 Module Options
 - 2.4 Package Options
 - 2.5 Service Options
 - 2.6 Values Format
 - 2.7 Input Rules
 - 2.8 Notifications
 - 2.9 Package Option Notification Tags
 - 2.10 Service Option Notification Tags
- 3 Client Service Info
- 4 Admin Service Info
- 5 Creating Packages

About The Universal Module

The Universal Module is the module to use when no other module will do. In some cases you may find that you need a module to request fields from a client for a particular service that will be provisioned manually. Dedicated servers are a perfect example of this. With the Universal Module you can create these fields dynamically, and even have them send a post or email notification when the package or service is created or changes.

Using the Universal Module

Installing the Universal Module

1. Visit [Settings] > [Company] > [Modules] > Available.
2. Click the "Install" button within the Universal Module module listing.

 When the module is installed, it should automatically load the "Manage" screen for the module. You can get back here any time by visiting [Settings] > [Company] > [Modules] > Installed and by clicking the "Manage" button for the module.

Manage Universal Module

ADD PRODUCT

Products

Product Label	Options
Dedicated Server	Edit Delete

Products

A Universal Module Product is a collection of package and service request fields, along with a few rules that apply to the product. The product acts as a template for creating packages and services.

Edit Product

Module Options

Name

Dedicated Server

Package Options

ADD PACKAGE OPTION

Label ?	Name	Type ?	Required ?	Encrypt ?	Values ?	
Data Center	dc	Select ▾	No ▾	No ▾	wash:Washington DC dallas:Dallas TX prov:Providence Utah	Remove

Input Rules

Notice	Post URL/Email ?	HTTP Code ?	Response Contains ?
Add			
Edit			

☒ Email Notification

Module Options

Option	Description
--------	-------------

Name	The name for the product, shown when creating or editing a package.
------	---

Package Options

Package options are fields that are requested when a package is added or updated. A product can have any number of package options.

Option	Description																		
Label	The display name of the form field.																		
Name	The name of the input field. This is the name of the value sent in all post and email notifications, and is also used when validating input rules.  If you've set the label to Email address then you'd likely want to set the name to email_address.																		
Type	The type of form field. <table> <tr> <th>Option</th><th>Description</th></tr> <tr> <td>Text</td><td>A regular text input field.</td></tr> <tr> <td>Textarea</td><td>A large text area field, for inserting multiple lines of text.</td></tr> <tr> <td>Password</td><td>A password input field, masking the user's input.</td></tr> <tr> <td>Select</td><td>A drop-down menu.</td></tr> <tr> <td>Radio</td><td>A radio field.</td></tr> <tr> <td>Checkbox</td><td>A checkbox field.</td></tr> <tr> <td>Hidden</td><td>A hidden field not visible to the user.</td></tr> <tr> <td>Secret</td><td>A field that is never rendered or stored for the package, but is sent in all post and email notifications.</td></tr> </table>	Option	Description	Text	A regular text input field.	Textarea	A large text area field, for inserting multiple lines of text.	Password	A password input field, masking the user's input.	Select	A drop-down menu.	Radio	A radio field.	Checkbox	A checkbox field.	Hidden	A hidden field not visible to the user.	Secret	A field that is never rendered or stored for the package, but is sent in all post and email notifications.
Option	Description																		
Text	A regular text input field.																		
Textarea	A large text area field, for inserting multiple lines of text.																		
Password	A password input field, masking the user's input.																		
Select	A drop-down menu.																		
Radio	A radio field.																		
Checkbox	A checkbox field.																		
Hidden	A hidden field not visible to the user.																		
Secret	A field that is never rendered or stored for the package, but is sent in all post and email notifications.																		
Required	If enabled, will require that some value is given for the field.																		
Encrypt	If enabled, will store input values encrypted in the database.																		
Values	See Values Format .																		
Input Rules	See Input Rules .																		
Notifications	See Notifications .																		

Service Options

Service options are fields that are requested when a service is added or updated. A product can have any number of service options.

Service Options

+

ADD SERVICE OPTION

Label ?	Name ?	Type ?	Required ?	Encrypt ?	Values ?	
Hostname	hostname	Text ▾	Yes ▾	No ▾		Remove
NS1 Prefix	ns1	Text ▾	Yes ▾	No ▾		Remove
NS2 Prefix	ns2	Text ▾	Yes ▾	No ▾		Remove
Root Password	password	Password ▾	No ▾	No ▾		Remove
Custom Partitioning	partition	Textarea ▾	No ▾	No ▾		Remove
Custom Setup	setup	Textarea ▾	No ▾	No ▾		Remove
Operating System	os	Select ▾	Yes ▾	No ▾	centos-62-64-bit:CentOS 6.2 64-bitfedora-16-64-bit:Fedora 16 64-bitubuntu-14.04-bit:Ubuntu 14.04 LTS 64-bit	Remove

Input Rules

Option	Description
Label	The display name of the form field.
Name	The name of the input field. This is the name of the value sent in all post and email notifications, and is also used when validating input rules. If you've set the label to Email address then you'd likely want to set the name to email_address. A field name of qty has special meaning. It will set the quantity recorded for the service.

Type	The type of form field. <table> <tr> <th>Option</th><th>Description</th></tr> <tr> <td>Text</td><td>A regular text input field.</td></tr> <tr> <td>Textarea</td><td>A large text area field, for inserting multiple lines of text.</td></tr> <tr> <td>Password</td><td>A password input field, masking the user's input.</td></tr> <tr> <td>Select</td><td>A drop-down menu.</td></tr> <tr> <td>Radio</td><td>A radio field.</td></tr> <tr> <td>Checkbox</td><td>A checkbox field.</td></tr> <tr> <td>Hidden</td><td>A hidden field not visible to the user.</td></tr> <tr> <td>Secret</td><td>A field that is never rendered or stored for the service, but is sent in all post and email notifications.</td></tr> </table>	Option	Description	Text	A regular text input field.	Textarea	A large text area field, for inserting multiple lines of text.	Password	A password input field, masking the user's input.	Select	A drop-down menu.	Radio	A radio field.	Checkbox	A checkbox field.	Hidden	A hidden field not visible to the user.	Secret	A field that is never rendered or stored for the service, but is sent in all post and email notifications.
Option	Description																		
Text	A regular text input field.																		
Textarea	A large text area field, for inserting multiple lines of text.																		
Password	A password input field, masking the user's input.																		
Select	A drop-down menu.																		
Radio	A radio field.																		
Checkbox	A checkbox field.																		
Hidden	A hidden field not visible to the user.																		
Secret	A field that is never rendered or stored for the service, but is sent in all post and email notifications.																		
Required	If enabled, will require that some value is given for the field.																		
Encrypt	If enabled, will store input values encrypted in the database.																		
Values	See Values Format .																		
Input Rules	See Input Rules .																		
Notifications	See Notifications .																		

Values Format

The values field is a serialized list of field data in the format of **name:value**, delimited by the | (pipe) character. The **name** is the value stored for the field and **value** is the value displayed. For example, if you would like to request a list of check boxes of various flavors of ice cream you might enter: vanilla:Vanilla|chocolate:Chocolate|strawberry:Strawberry|rocky_road:Rocky Road



Using ':' or '|' in your fields

Both : and | are reserved characters, so if you'd like to use them in your value entry you must escape those characters with a \ (back slash).

Input Rules

Input rules are a JSON encoded object of input validation rules. They allow you to specify how each field should be validated against errors on input. For additional details on input rules see [Error Checking](#). Let's take a look at an example:

Example Input Rules

```
{
  "hostname":{
    "valid":{
      "rule":"isEmpty",
      "negate":true,
      "message":"Hostname must not be empty."
    }
  },
  "password":{
    "valid":{
      "rule":["matches", "/^[a-z0-9\\$\\%\\^]{10,20}$/i"],
      "message":"Please enter a password using alphanumeric characters between 10 and 20
characters in length. You may also include special characters like '$%^'."
    }
  }
}
```

In the above example we're validating the **hostname** field using the **isEmpty** rule and negating it using the **negate** attribute. This ensures that the hostname field is *not empty* when submitted. If the rule validation fails (hostname is empty) the message we defined using the **message** attribute will be displayed to the user. We are also validating the **password** field at the same time using the **matches** rule, which evaluates a given regular expression. This rule requires that the password contain between 10 and 20 characters, inclusive, all of which must be characters of any combination from A through Z, upper or lower case, the digits 0 through 9, and the special characters \$, %, and ^.

Note that special character literals like \$, %, and ^ should be escaped where appropriate in the regular expression via *double backslashes*. Additionally, so should special characters in JSON format, like quotes (") and back slashes (\) themselves. Creating invalid JSON will cause all fields to pass validation because the JSON cannot be parsed to retrieve the rules. Creating invalid regular expressions will cause unexpected behavior.

Notifications

Notifications allow the universal module to post to a given URL or email a given address when a certain action occurs.

Notice	Post URL/Email ?	HTTP Code ?	Response Contains ?
Add	https://api.domain.com	200	
Edit	https://api.domain.com		success
Suspend	email@domain.com		
Unsuspend	email@domain.com		
Cancel	https://api.domain.com	200	
Renew			
Package Change			

Email Notification

From Email

Subject

HTML
Text

Source
Find
Print
Table of Contents
Search
Help

B
I
U
abc
X₂
X²
Style
Format
Font
Size
A+
A-

```
{% debug %}
```

[EDIT PRODUCT](#)

Package Option Notifications

Package option notifications are executed when an action is performed on the package.

Option	Description
Notice	The action triggers the notification.
Post URL /Email	The URL to post to, or the email address to email, when the notice is triggered
HTTP Code	The HTTP Response Code returned by the URL that signifies a successful transmission.
Response Contains	Some portion of the output returned by the URL that signifies a successful transmission. This can be used separately, or in conjunction with the HTTP Code for more refined control over successful responses.
From Email	The address to send email notifications from.
Subject	The subject of the email notification.

HTML/Text	The HTML and Text body content for the email notification.
-----------	--

Package Option Notification Tags

The package option notification emails allow for the following tags:

Tags	Description	Notes	Version
{*}	The value submitted for this custom field	{*} is NOT an actual tag. Every custom package field has a tag associated with it, this tag is labeled by the fields name represented here by * (e.g. {my_field})	
{*_name}	The label associated with the submitted value for fields of type select or radio	This tag is only available for radio and select fields (e.g. {my_field_name})	4.1
{*_values}	A list of checked boxes for this field each containing the box's label and value	This tag is only available for checkbox fields (e.g. {my_field_values})	4.1
{_action}	The code of the action for which this notification is being sent		4.1
{_package.id}	The system-level service ID	The _package tag is not available for the package addition notification	4.1
{_package.description}	The package's description		4.1
{_package.module_row}	The ID of the module row the package is associated with		4.1
{_package.status}	The status of the package		4.1



Email Tags

The {*_values} tag contains data in the following format. For a guide on using email tags in Blesta see the [Customizing Emails](#) page.

```
custom_1_values = array(
    array(
        'value' => '1',
        'name' => 'Option 1'
    ),
    array(
        'value' => '2',
        'name' => 'Option 2'
    )
);
```

Due to the nature of tag objects containing several fields, many of which are likely irrelevant for use in email templates, but may be useful to you in certain circumstances, a dump of the tags are shown below.

The {package} object

```

stdClass Object
(
    [id] => '94'
    [id_format] => '{num}'
    [id_value] => '92'
    [module_id] => '2'
    [name] => 'Test Universal Package'
    [description] => 'A test package for the universal module.'
    [description_html] => '<p>A test package for the Universal Module.</p>'
    [qty] => null
    [module_row] => '1'
    [module_group] => null
    [taxable] => '1'
    [single_term] => '0'
    [status] => 'active'
    [company_id] => '1'
    [prorata_day] => null
    [prorata_cutoff] => null
    [id_code] => '92'
    [email_content] => Array
    (
        [0] => stdClass Object
        (
            [lang] => 'en_us'
            [html] => '<p>Welcome email for the Universal Module.</p>'
            [text] => 'Welcome email for the Universal Module.'
        )
    )
    [pricing] => Array
    (
        [0] => stdClass Object
        (
            [id] => '628'
            [pricing_id] => '875'
            [package_id] => '94'
            [term] => '1'
            [period] => 'month'
            [price] => '2500.0000'
            [setup_fee] => '0.0000'
            [cancel_fee] => '0.0000'
            [currency] => 'USD'
        )
    )
    [meta] => stdClass Object
    [groups] => Array
    (
        [0] => stdClass Object
        (
            [id] => '10'
            [name] => 'Test Universal Package Group'
            [description] => null
            [type] => 'standard'
            [company_id] => '1'
            [allow_upgrades] => '1'
        )
    )
    [option_groups] => Array
    (
        [0] => stdClass Object
        (
            [id] => '26'
            [name] => 'Test Universal Option Group'
            [description] => 'A list of options'
        )
    )
)

```

Service Option Notifications

Service option notifications are executed when an action is performed on the service.

Option	Description																
Notice	The action triggers the notification.																
	<table><tr><th>Option</th><th>Description</th></tr><tr><td>Add</td><td>Executed when a service is created.</td></tr><tr><td>Edit</td><td>Executed when a service is updated.</td></tr><tr><td>Suspend</td><td>Executed when a service is suspended.</td></tr><tr><td>Unsuspend</td><td>Executed when a service is unsuspended.</td></tr><tr><td>Cancel</td><td>Executed when a service is canceled.</td></tr><tr><td>Renew</td><td>Executed when a service is renewed. That is, the invoice generated for the service renewal is paid.</td></tr><tr><td>Package Change</td><td>Executed when the package is changed for the service.</td></tr></table>	Option	Description	Add	Executed when a service is created.	Edit	Executed when a service is updated.	Suspend	Executed when a service is suspended.	Unsuspend	Executed when a service is unsuspended.	Cancel	Executed when a service is canceled.	Renew	Executed when a service is renewed. That is, the invoice generated for the service renewal is paid.	Package Change	Executed when the package is changed for the service.
	Option	Description															
	Add	Executed when a service is created.															
	Edit	Executed when a service is updated.															
	Suspend	Executed when a service is suspended.															
	Unsuspend	Executed when a service is unsuspended.															
	Cancel	Executed when a service is canceled.															
	Renew	Executed when a service is renewed. That is, the invoice generated for the service renewal is paid.															
Package Change	Executed when the package is changed for the service.																
Post URL /Email	The URL to post to, or the email address to email, when the notice is triggered																
HTTP Code	The HTTP Response Code returned by the URL that signifies a successful transmission.																
Response Contains	Some portion of the output returned by the URL that signifies a successful transmission. This can be used separately, or in conjunction with the HTTP Code for more refined control over successful responses.																
From Email	The address to send email notifications from.																
Subject	The subject of the email notification.																
HTML/Text	The HTML and Text body content for the email notification.																

Service Option Notification Tags

The service option notification emails allow for the following tags:

Tags	Description	Notes	Version
{*}	The value submitted for this custom field	{*} is NOT an actual tag. It is only a placeholder in this documentation. Every custom service field has a tag associated with it, this tag is labeled by the fields name represented here by * (e.g. {my_field})	
{*_name}	The label associated with the submitted value for fields of type select or radio	This tag is only available for radio and select fields (e.g. {my_field_name})	4.1
{*_values}	A list of checked boxes for this field each containing the box's label and value	This tag is only available for checkbox fields (e.g. {my_field_values})	4.1
{_action}	The code of the action for which this notification is being sent		4.1
{_service.id}	The system-level service ID	The _service tag is not available for the service addition notification	4.1
{_service.status}	The service status		4.1
{_service.date_added}	The UTC datetime stamp of the date the service was added	Formatted as "YYYY-MM-DD HH:MM:SS"	4.1
{_service.date_renews}	The UTC datetime stamp of the service renew date	Formatted as "YYYY-MM-DD HH:MM:SS"	4.1
{_service.date_last_renewed}	The UTC datetime stamp of the service's last renew date	Formatted as "YYYY-MM-DD HH:MM:SS"	4.1
{_service.date_suspended}	The UTC datetime stamp of the date the service was suspended	Formatted as "YYYY-MM-DD HH:MM:SS"	4.1
{_service.date_canceled}	The UTC datetime stamp of the date the service was canceled	Formatted as "YYYY-MM-DD HH:MM:SS"	4.1

{_package.id}	The system-level service ID		4.1
{_package.description}	The package's description		4.1
{_package.module_row}	The ID of the module row the package is associated with		4.1
{_package.status}	The status of the package		4.1
{_other.*}	The _other tag contains additional information, most of which is received through user input	Here * is the name of the field to access from the other tag (e.g. pricing_id)	



Email Tags

The {*_values} tag contains data in the following format. For a guide on using email tags in Blesta see the [Customizing Emails](#) page.

```
custom_1_values = array(
    array(
        'value' => '1',
        'name' => 'Option 1'
    ),
    array(
        'value' => '2',
        'name' => 'Option 2'
    )
);
```

Due to the nature of tag objects containing several fields, many of which are likely irrelevant for use in email templates, but may be useful to you in certain circumstances, a dump of the tags are shown below.

The {service} object

```
stdClass Object
(
    [id] => 234
    [parent_service_id] =>
    [package_group_id] => 1
    [id_format] => PD-{num}
    [id_value] => 229
    [pricing_id] => 22
    [client_id] => 3
    [module_row_id] => 6
    [coupon_id] =>
    [qty] => 1
    [status] => in_review
    [date_added] => 2013-05-13 18:33:14
    [date_renews] => 2013-06-13 18:33:14
    [date_last_renewed] =>
    [date_suspended] =>
    [date_canceled] =>
    [id_code] => PD-229
    [fields] => Array
        (
            [0] => stdClass Object
                (
                    [key] => cpanel_confirm_password
                    [value] => password
                    [serialized] => 0
                    [encrypted] => 1
                )
            [1] => stdClass Object
                (
                    [key] => cpanel_domain
                    [value] => domain.com
                    [serialized] => 0
                    [encrypted] => 0
                )
        )
    )
```

```

    )
[2] => stdClass Object
(
    [key] => cpanel_password
    [value] => password
    [serialized] => 0
    [encrypted] => 1
)
[3] => stdClass Object
(
    [key] => cpanel_username
    [value] => domainco
    [serialized] => 0
    [encrypted] => 0
)
)
[package_pricing] => stdClass Object
(
    [id] => 22
    [package_id] => 13
    [term] => 1
    [period] => month
    [price] => 8.0000
    [setup_fee] => 0.0000
    [cancel_fee] => 0.0000
    [currency] => USD
)
[package] => stdClass Object
(
    [id] => 13
    [id_format] => {num}
    [id_value] => 1
    [module_id] => 4
    [name] => Bronze Standard
    [description] =>
    [description_html] => <p>
    <span style="color:#008000;"><em>Good Value!</em></span></p>
    [qty] =>
    [module_row] => 6
    [module_group] =>
    [taxable] => 0
    [status] => active
    [company_id] => 1
)
[name] => domain.com
[coupon] => stdClass Object
(
    [id] => 2
    [code] => knW3Lesn
    [company_id] => 1
    [used_qty] => 16
    [max_qty] => 0
    [start_date] => 2017-02-02 08:00:00
    [end_date] => 2018-02-03 07:59:59
    [status] => active
    [recurring] => 0
    [limit_recurring] => 0
    [apply_package_options] => 1
    [internal_use_only] => 0
    [type] => exclusive
    [amounts] => Array
    (
        [0] => stdClass Object
        (
            [coupon_id] => 2
            [currency] => USD
            [amount] => 5.0000
            [type] => amount
        )
    )
)

```

```

[packages] => Array
(
    [0] => stdClass Object
        (
            [coupon_id] => 2
            [package_id] => 80
        )
)
)

```

The {package} object

```

stdClass Object
(
    [id] => '94'
    [id_format] => '{num}'
    [id_value] => '92'
    [module_id] => '2'
    [name] => 'Test Universal Package'
    [description] => 'A test package for the universal module.'
    [description_html] => '<p>A test package for the Universal Module.</p>'
    [qty] => null
    [module_row] => '1'
    [module_group] => null
    [taxable] => '1'
    [single_term] => '0'
    [status] => 'active'
    [company_id] => '1'
    [prorata_day] => null
    [prorata_cutoff] => null
    [id_code] => '92'
    [email_content] => Array
        (
            [0] => stdClass Object
                (
                    [lang] => 'en_us'
                    [html] => '<p>Welcome email for the Universal Module.</p>'
                    [text] => 'Welcome email for the Universal Module.'
                )
        )
    [pricing] => Array
        (
            [0] => stdClass Object
                (
                    [id] => '628'
                    [pricing_id] => '875'
                    [package_id] => '94'
                    [term] => '1'
                    [period] => 'month'
                    [price] => '2500.0000'
                    [setup_fee] => '0.0000'
                    [cancel_fee] => '0.0000'
                    [currency] => 'USD'
                )
        )
    [meta] => stdClass Object
    [groups] => Array
        (
            [0] => stdClass Object
                (
                    [id] => '10'
                    [name] => 'Test Universal Package Group'
                    [description] => null
                    [type] => 'standard'
                    [company_id] => '1'
                    [allow_upgrades] => '1'
                )
        )
)

```

```
[option_groups] => Array
(
    [0] => stdClass Object
        (
            [id] => '26'
            [name] => 'Test Universal Option Group'
            [description] => 'A list of options'
        )
)
```

Client Service Info

Client Service Info renders the information shown in the expandable area and on the Information tab when clicking to manage the service within the client area when clicking on the service row.

Example Code (Replace tags with appropriate service or package tags. In this example, we have service fields called vpnuser, vpnpass, vpnhost)

```
<div class="table-responsive">
<table class="table table-curved table-striped">
    <thead><tr>
        <th>VPN USer</th>
        <th>VPN Pass</th>
        <th>VPN Host</th>
    </tr>
</thead>
<tbody>
<tr>
    <td>{service.fields.vpnuser}</td>
    <td>{service.fields.vpnpass}</td>
    <td>Login at <a href="https://{service.fields.vpnhost}">{service.fields.vpnhost}</a><
/td>
    </tr>
</tbody>
</table>
</div>
```

Admin Service Info

Admin Service Info renders the information shown in the expandable area within the client profile when clicking on the service row in the admin area.

Example Code (Replace tags with appropriate service or package tags. In this example, we have service fields called vpnuser, vpnpass, vpnhost)

```
<table class="table">
    <tbody><tr class="heading_row">
        <td class="border_left">VPN USer</td>
        <td>VPN Pass</td>
        <td>VPN Host</td>
    </tr>
    <tr>
        <td>{service.fields.vpnuser}</td>
        <td>{service.fields.vpnpass}</td>
        <td>Login at <a href="https://{service.fields.vpnhost}">{service.fields.vpnhost}</a><
/td>
    </tr>
</tbody>
</table>
```

Creating Packages



Universal Module product service field names are used for the email tag. For example, a service field with the name "hostname" would result in a tag available to the welcome email of {service.hostname}. Universal module products are unique in this regard because each product may have different tags, defined by the staff member who created them.