

[PrestaShopException]

Property FeatureValue->value is not valid
at line 954 in file classes/ObjectModel.php

```

949.         }
950.
951.         $message = $this->validateField($field, $value, $id_lang);
952.         if ($message !== true) {
953.             if ($die) {
954.                 throw new PrestaShopException($message);
955.             }
956.             return $error_return ? $message : false;
957.         }
958.     }
959. }
```

- **ObjectModelCore->validateFieldsLang** - [\[line 295 - classes/ObjectModel.php\]](#)

```

290.         // Backward compatibility
291.         if (method_exists($this, 'getTranslationsFieldsChild')) {
292.             return $this->getTranslationsFieldsChild();
293.         }
294.
295.         $this->validateFieldsLang();
296.         $is_lang_multishop = $this->isLangMultishop();
297.
298.         $fields = array();
299.         if ($this->id_lang === null) {
300.             foreach (Language::getIDs(false) as $id_lang) {
```

- **ObjectModelCore->getFieldsLang** - [\[line 511 - classes/ObjectModel.php\]](#)

```

506.         return false;
507.     }
508.
509.     // Database insertion for multilingual fields related to the obj
510.     if (!empty($this->def['multilang'])) {
511.         $fields = $this->getFieldsLang();
512.         if ($fields && is_array($fields)) {
513.             $shops = Shop::getCompleteListOfShopsID();
514.             $asso = Shop::getAssoTable($this->def['table'].'_lang');
515.             foreach ($fields as $field) {
516.                 foreach (array_keys($field) as $key) {
```

- **ObjectModelCore->add** - [\[line 178 - classes/FeatureValue.php\]](#) - [\[2 Arguments\]](#)

```

Argument [0]
1
Argument [1]
```

- **FeatureValueCore->add** - [line 171 - classes/FeatureValue.php]

```

166.         // Feature doesn't exist, create it
167.         $feature_value = new FeatureValue();
168.         $feature_value->id_feature = (int)$id_feature;
169.         $feature_value->custom = (bool)$custom;
170.         $feature_value->value = array_fill_keys(Language::getIDs(false),
171.         $feature_value->add();
172.
173.         return (int)$feature_value->id;
174.     }
175.
176.     public function add($autodate = true, $nullValues = false)

```

- **FeatureValueCore::addFeatureValueImport** - [line 1864 - controllers/admin/AdminImportController.php] - [5 Arguments]

```

1859.             $id_feature = (int)Feature::addFeatureImport($id_feature);
1860.             $id_product = null;
1861.             if ($force_ids || $match_ref) {
1862.                 $id_product = (int)$product->id;
1863.             }
1864.             $id_feature_value = (int)FeatureValue::addFeatureValueImport($id_feature, $id_product, $value, $autodate, $nullValues);
1865.             Product::addFeatureProductImport($product->id, $id_feature_value);
1866.         }
1867.     }
1868. }
1869. // clean feature positions to avoid conflict

```

- **AdminImportControllerCore->productImport** - [line 3493 - controllers/admin/AdminImportController.php]

```

3488.         case $this->entities[$import_type = $this->l('Category')] :
3489.             $this->categoryImport();
3490.             $this->clearSmartyCache();
3491.             break;
3492.         case $this->entities[$import_type = $this->l('Product')] :
3493.             $this->productImport();
3494.             $this->clearSmartyCache();
3495.             break;
3496.         case $this->entities[$import_type = $this->l('Customer')] :
3497.             $this->customerImport();
3498.             break;

```

- **AdminImportControllerCore->postProcess** - [line 178 - classes/controller/Controller.php]

```

173.         if (!$this->content_only && ($this->display_header || (isset($this->media) && $this->media))) {
174.             $this->setMedia();
175.         }
176.

```

```
177.          // postProcess handles ajaxProcess
178.          $this->postProcess();
179.
180.          if (!empty($this->redirect_after)) {
181.              $this->redirect();
182.          }
183.
```

- **ControllerCore->run** - [\[line 367 - classes/Dispatcher.php\]](#)

```
362.          if (isset($params_hook_action_dispatcher)) {
363.              Hook::exec('actionDispatcher', $params_hook_action_dispa
364.          }
365.
366.          // Running controller
367.          $controller->run();
368.      } catch (PrestaShopException $e) {
369.          $e->displayMessage();
370.      }
371.  }
372.
```

- **DispatcherCore->dispatch** - [\[line 58 - admin/index.php\]](#)

```
53. if (!isset($_REQUEST['controller']) && isset($_REQUEST['tab'])) {
54.     $_REQUEST['controller'] = strtolower($_REQUEST['tab']);
55. }
56.
57. // Prepare and trigger admin dispatcher
58. Dispatcher::getInstance()->dispatch();
```