Installation in progress. Please wait.

Installation Output:
Info - Auto build of User done.
Info - Auto build of Role done.
Info - Auto build of Group done.
Info - Auto build of Task done.
Info - (Added)
Info - Item (subst)
Info - MANY_MANY
Info - Setting Task->activityItems.
Info - (Added)
Info - Permission
Info - HAS_MANY
Info - Setting Task->permissions.
Info - Setting Task->name.
Info - Setting Task->dueDateTime.
Info - Setting Task->description.
Info - Setting Task->completed.
Info - Setting Task->completedDateTime.
Info - Task Being Created.
Info - Auto build of Note done.
Info - (Added)
Info - (Set)
Info - FileContent
Info - HAS_ONE
Info - Setting FileModel->fileContent.
Info - Setting FileModel->type.
Info - Setting FileModel->size.
Info - Setting FileModel->name.

Copyright © Zurmo Inc., 2012. All Rights reserved.

PHP warning

Invalid argument supplied for foreach()

/var/www/zurmo/app/protected/modules/zurmo/utils/ReadPermissionsOptimizationUtil.php(911)

```
899             {
900                 return;
901             }
902             if ($role->role->id > 0)
903             {
904                 self::decrementCountForAllSecurableItems            ($mungeTableName, $role->role);
905                 self::decrementParentRolesCountsForAllSecurableItems($mungeTableName, $role->role);
906             }
907         }
908
909         protected static function bulkIncrementParentRolesCounts($mungeTableName, $securableItemIds, Role $role)
910         {
911             foreach ($securableItemIds as $securableItemId)
912             {
913                 self::incrementParentRolesCounts($mungeTableName, intval($securableItemId), $role);
914             }
915         }
916
917         protected static function bulkDecrementParentRolesCounts($mungeTableName, $securableItemIds, Role $role)
918         {
919             foreach ($securableItemIds as $securableItemId)
920             {
921                 self::decrementParentRolesCounts($mungeTableName, intval($securableItemId), $role);
922             }
923         }
```

Stack Trace

#0 /var/www/zurmo/app/protected/modules/zurmo/utils/ReadPermissionsOptimizationUtil.php(662): **ReadPermissionsOptimizationUtil::bulkIncrementParentRolesCounts**("account_read", null, Role)

```
657                 $sql = "select securableitem_id
658                         from   ownedsecurableitem
659                         where  owner__user_id = $userId";
660                 $securableItemIds = R::getCol($sql);
661                 //Increment the parent roles for securableItems that the user is the owner on.
662                 self::bulkIncrementParentRolesCounts($mungeTableName, $securableItemIds, $user->role);
663
664                 //Get all downstream groups the user is in including any groups that are in those groups recursively.
665                 //Then for each group found, add weight for the user's upstream roles.
666                 $groupMungeIds = array();
667                 foreach ($user->groups as $group)
```

#1 /var/www/zurmo/app/protected/modules/users/models/User.php(205): **ReadPermissionsOptimizationUtil::userAddedToRole**(User)

```
200         protected function afterSave()
201         {
202             if (((isset($this->originalAttributeValues['role'])) || $this->isNewModel) &&
203                 $this->role != null && $this->role->id > 0)
204             {
205                 ReadPermissionsOptimizationUtil::userAddedToRole($this);
206                 $this->onChangeRights();
207                 $this->onChangePolicies();
208             }
209             if (isset($this->originalAttributeValues['language']) && Yii::app()->user->userModel != null &&
210                 Yii::app()->user->userModel == $this)
```

#2 /var/www/zurmo/app/protected/extensions/zurmoinc/framework/models/RedBeanModel.php(1887): **User->afterSave**()

```
1882                                                                                    $modelClassName);
1883                                 $baseModelClassName = $modelClassName;
1884                             }
1885                         }
1886                         $this->modified = false;
1887                         $this->afterSave();
1888                         RedBeanModelsCache::cacheModel($this);
1889                         $this->isSaving = false;
1890                         return true;
1891                     }
1892                 }
```

#3 /var/www/zurmo/app/protected/modules/zurmo/models/Item.php(98): **RedBeanModel->save**(false, null)

```
093
094         // Makes Item appear on the stack so that auditing can ensure
095         // that things owned by Item are only saved by Item and not directly.
096         public function save($runValidation = true, array $attributeNames = null)
097         {
098             return parent::save($runValidation, $attributeNames);
099         }
100
101         /**
102          * Special handling of the import scenario. When you are importing a model, you can potentially set the
103          * created/modified user/datetime which is normally not allowed since they are read-only attributes.  This
```

#4 /var/www/zurmo/app/protected/modules/users/models/User.php(186): **Item->save**(false, null)

```
181         public function save($runValidation = true, array $attributeNames = null)
182         {
183             $passwordChanged = array_key_exists('hash', $this->originalAttributeValues);
```

```
184             unset($this->originalAttributeValues['hash']);
185             assert('!isset($this->originalAttributeValues["hash"])');
186             $saved = parent::save($runValidation, $attributeNames);
187             if ($saved && $passwordChanged)
188             {
189                 AuditEvent::
190                 logAuditEvent('UsersModule', UsersModule::AUDIT_EVENT_USER_PASSWORD_CHANGED, $this->username, $this);
191             }
```

#5 /var/www/zurmo/app/protected/extensions/zurmoinc/framework/models/RedBeanModel.php(1825): **User->save**(false)

```
1820                                    if ($this->isSavableFromRelation &&
1821                                       !($this->isAttributeRequired($relationName) &&
1822                                         !$relatedModel->isModified() &&
1823                                         $relatedModel->id > 0))
1824                                    {
1825                                        if (!$relatedModel->save(false))
1826                                        {
1827                                            $this->isSaving = false;
1828                                            return false;
1829                                        }
1830                                    }
```

#6 /var/www/zurmo/app/protected/extensions/zurmoinc/framework/utils/RedBeanDatabaseBuilderUtil.php(87): **RedBeanModel->save**()

```
82                 if (!$model instanceof OwnedModel && !$model instanceof OwnedCustomField && !$model instanceof OwnedMultipleValuesCustomField)
83                 {
84                     try
85                     {
86                         $model->setScenario('autoBuildDatabase');
87                         $saved = $model->save();
88
89                         if ($saved)
90                         {
91                             $metadata = $model->getMetadata();
92                             foreach ($metadata as $unused => $classMetadata)
```

#7 /var/www/zurmo/app/protected/modules/install/utils/InstallUtil.php(594): **RedBeanDatabaseBuilderUtil::autoBuildModels**(array("AuditEvent", "Account", "AccountsFilteredList", "NamedSecurableItem", ...), MessageLogger)

```
589             foreach (Module::getModuleObjects() as $module)
590             {
591                 $moduleAndDependenciesRootModelNames = $module->getRootModelNamesIncludingDependencies();
592                 $rootModels = array_merge($rootModels, array_diff($moduleAndDependenciesRootModelNames, $rootModels));
593             }
594             RedBeanDatabaseBuilderUtil::autoBuildModels($rootModels, $messageLogger);
595         }
596
597         /**
598          * Freezes the database.
599          */
```

#8 /var/www/zurmo/app/protected/modules/install/utils/InstallUtil.php(839): **InstallUtil::autoBuildDatabase**("MessageLogger")

```
834             InstallUtil::createSuperUser(   'super',
835                                             $form->superUserPassword);
836             $messageLogger = new MessageLogger($messageStreamer);
837             Yii::app()->custom->runBeforeInstallationAutoBuildDatabase($messageLogger);
838             $messageStreamer->add(Yii::t('Default', 'Starting database schema creation.'));
839             InstallUtil::autoBuildDatabase($messageLogger);
840             $messageStreamer->add(Yii::t('Default', 'Database schema creation complete.'));
841             $messageStreamer->add(Yii::t('Default', 'Rebuilding Permissions.'));
842             ReadPermissionsOptimizationUtil::rebuild();
843             $messageStreamer->add(Yii::t('Default', 'Freezing database.'));
844             InstallUtil::freezeDatabase();
```

#9 /var/www/zurmo/app/protected/modules/install/controllers/DefaultController.php(130): **InstallUtil::runInstallation**(InstallSettingsForm, MessageStreamer)

```
125                 $view = new InstallPageView($nextView);
126             echo $view->render();
127
128             $template = CHtml::script("$('#logging-table').prepend('{message}<br/>');");
129             $messageStreamer = new MessageStreamer($template);
130             InstallUtil::runInstallation($form, $messageStreamer);
131             if ($form->installDemoData)
132             {
133                 echo CHtml::script('$("#progress-table").hide(); $("#demo-data-table").show();');
134             }
135             else
```

#10 /var/www/zurmo/app/protected/modules/install/controllers/DefaultController.php(104): **InstallDefaultController->actionRunInstallation**(InstallSettingsForm)

```
099                                                     $serviceCheckResultsDataForDisplay);
100                 $view = new InstallPageView($checkServicesView);
101                 echo $view->render();
102                 Yii::app()->end(0, false);
103             }
104             $this->actionRunInstallation($form);
105             Yii::app()->end(0, false);
106         }
107
108         protected function actionValidateSettings($model)
109         {
```

#11 /var/www/zurmo/app/protected/modules/install/controllers/DefaultController.php(82): **InstallDefaultController->additionalSystemCheck**(InstallSettingsForm)

```
77             {
78                 if (isset($_POST['InstallSettingsForm']))
79                 {
80                     $form->setAttributes($_POST['InstallSettingsForm']);
81                     //in case if additionalSystemCheck it will render its own screen
82                     $this->additionalSystemCheck($form);
83                     Yii::app()->end(0, false);
84                 }
85             }
86             $settingsView = new InstallSettingsView($this->getId(), $this->getModule()->getId(), $form);
87             $view = new InstallPageView($settingsView);
```

#12 /var/www/zurmo/yii/framework/web/actions/CInlineAction.php(50): **InstallDefaultController->actionSettings**()

```
45         $controller=$this->getController();
46         $method=new ReflectionMethod($controller, $methodName);
47         if($method->getNumberOfParameters()>0)
48             return $this->runWithParamsInternal($controller, $method, $params);
49         else
50             return $controller->$methodName();
51     }
52
53 }
```

#13 /var/www/zurmo/yii/framework/web/CController.php(309): **CInlineAction->runWithParams**(array())

```
304     {
305         $priorAction=$this->_action;
306         $this->_action=$action;
307         if($this->beforeAction($action))
308         {
309             if($action->runWithParams($this->getActionParams())===false)
310                 $this->invalidActionParams($action);
311             else
312                 $this->afterAction($action);
313         }
314         $this->_action=$priorAction;
```

#14 /var/www/zurmo/yii/framework/web/filters/CFilterChain.php(134): **CController->runAction**(CInlineAction)

```
129             $filter=$this->itemAt($this->filterIndex++);
```

```
130             Yii::trace('Running filter '.($filter instanceof CInlineFilter ? get_class($this->controller).'.filter'.$filter->name.'()':get_class($filter).'.filter()'),'system.web.filters.CFilterChain');
131             $filter->filter($this);
132         }
133         else
134             $this->controller->runAction($this->action);
135     }
136 }
```

#15 /var/www/zurmo/yii/framework/web/filters/CFilter.php(41): **CFilterChain->run**()

```
36      */
37      public function filter($filterChain)
38      {
39          if($this->preFilter($filterChain))
40          {
41              $filterChain->run();
42              $this->postFilter($filterChain);
43          }
44      }
45
46      /**
```

#16 /var/www/zurmo/yii/framework/web/filters/CFilterChain.php(131): **CFilter->filter**(CFilterChain)

```
126     {
127         if($this->offsetExists($this->filterIndex))
128         {
129             $filter=$this->itemAt($this->filterIndex++);
130             Yii::trace('Running filter '.($filter instanceof CInlineFilter ? get_class($this->controller).'.filter'.$filter->name.'()':get_class($filter).'.filter()'),'system.web.filters.CFilterChain');
131             $filter->filter($this);
132         }
133         else
134             $this->controller->runAction($this->action);
135     }
136 }
```

#17 /var/www/zurmo/yii/framework/web/CController.php(292): **CFilterChain->run**()

```
287             $this->runAction($action);
288         else
289         {
290             $priorAction=$this->_action;
291             $this->_action=$action;
292             CFilterChain::create($this,$action,$filters)->run();
293             $this->_action=$priorAction;
294         }
295     }
296
297     /**
```

#18 /var/www/zurmo/yii/framework/web/CController.php(266): **CController->runActionWithFilters**(CInlineAction, array(array("application.modules.install.controllers.filters.InstallControlle...")))

```
261         {
262             if(($parent=$this->getModule())===null)
263                 $parent=Yii::app();
264             if($parent->beforeControllerAction($this,$action))
265             {
266                 $this->runActionWithFilters($action,$this->filters());
267                 $parent->afterControllerAction($this,$action);
268             }
269         }
270         else
271             $this->missingAction($actionID);
```

#19 /var/www/zurmo/yii/framework/web/CWebApplication.php(276): **CController->run**("settings")

```
271         {
272             list($controller,$actionID)=$ca;
273             $oldController=$this->_controller;
274             $this->_controller=$controller;
275             $controller->init();
276             $controller->run($actionID);
277             $this->_controller=$oldController;
278         }
279         else
280             throw new CHttpException(404,Yii::t('yii','Unable to resolve the request "{route}".',
281                 array('{route}'=>$route===''?$this->defaultController:$route)));
```

#20 /var/www/zurmo/yii/framework/web/CWebApplication.php(135): **CWebApplication->runController**("install/default/settings")

```
130             foreach(array_splice($this->catchAllRequest,1) as $name=>$value)
131                 $_GET[$name]=$value;
132         }
133         else
134             $route=$this->getUrlManager()->parseUrl($this->getRequest());
135         $this->runController($route);
136     }
137
138     /**
139      * Registers the core application components.
140      * This method overrides the parent implementation by registering additional core components.
```

#21 /var/www/zurmo/yii/framework/base/CApplication.php(162): **CWebApplication->processRequest**()

```
157      */
158     public function run()
159     {
160         if($this->hasEventHandler('onBeginRequest'))
161             $this->onBeginRequest(new CEvent($this));
162         $this->processRequest();
163         if($this->hasEventHandler('onEndRequest'))
164             $this->onEndRequest(new CEvent($this));
165     }
166
167     /**
```

#22 /var/www/zurmo/app/protected/extensions/zurmoinc/framework/components/WebApplication.php(42): **CApplication->run**()

```
37          */
38         public function run()
39         {
40             try
41             {
42                 parent::run();
43             }
44             catch (ChecksumMismatchException $e)
45             {
46                 echo 'A checksum mismatch has occurred while retrieving a cached model. ' .
47                      'This is most likely caused by setting debug=true. The cache must be cleared.'; // Not Coding Standard
```

#23 /var/www/zurmo/app/index.php(56): **WebApplication->run**()

```
51      $webApplication = Yii::createApplication('WebApplication', $config);
52      if (file_exists($runtimePath))
53      {
54          $webApplication->setRuntimePath($runtimePath);
55      }
56      $webApplication->run();
57      exit;
58  ?>
```

2012-07-08 23:28:28 Apache/2.2.22 (Ubuntu) Yii Framework/1.1.10