

Test Case: no-orphans

FAIL **PASS**

Test Scenario

Tagging three paragraphs, changing orders, and finally deleting one paragraph works as expecting: keeping tag/paragraph associations correct and removing the field tag entities as appropriate.

Pre-Conditions

1. An entity type with a paragraphs field.
2. The paragraphs field has field tagging enabled.
3. The paragraphs field allows unlimited items.

Test Data

tag1	alpha
tag2	bravo
tag3	charlie

Test Execution

Test Steps	Test Results
<ol style="list-style-type: none">1. Create a copy paragraph with alpha as the value and tag it with alpha.2. Create another copy paragraph with bravo as the value and tag it with bravo.3. Save entity.4. Reload the entity edit page.	FAIL PASS Assert the tags match their content.
<ol style="list-style-type: none">5. Drag bravo before alpha6. Save entity.7. Reload the entity edit page.	FAIL PASS Assert the tags match their content.
<ol style="list-style-type: none">8. Create a third copy paragraph with charlie as the value and tag it with charlie.9. Drag it to the top position.10. Save entity.11. Reload the entity edit page.	FAIL PASS Assert all three tags match their content. FAIL PASS Assert the order is charlie , bravo , alpha .
<ol style="list-style-type: none">12. Drag them in the original order: alpha, bravo, charlie.13. Remove the paragraph with charlie.14. Save entity.15. Reload the entity edit page.	FAIL PASS Assert the edit form loads. FAIL PASS Assert all two tags match their content.

Test Steps	Test Results
16. Add a new third paragraph.	FAIL PASS Assert the tag field is empty, does not contain charlie

Comments

Please mention browser, device type, where you logged in, and anything else that can help diagnose failure.

Test completed by: Aaron Klump

Test Case: tags-in-code

FAIL **PASS**

Test Scenario

It is possible to tag a field when creating and updating an entity programmatically; tags get attached correctly on read; field tag entities are deleted when the parent entity is deleted.

Pre-Conditions

1. An entity type with an image field called `field_images`
2. The image field has field tagging enabled.
3. The image field allows unlimited images.
4. An entity type with a paragraphs field called `field_main_copy`
5. The paragraphs field has field tagging enabled.
6. The paragraphs field allows unlimited items.

Test Data

tag1	<code>banner,</code> <code>vimeo</code>
tag2	<code>thumb image</code>
tag3	<code>foo bara</code>
ptag1	<code>apple</code>
ptag2	<code>banana</code>

Test Execution

Test Steps	Test Results
<ol style="list-style-type: none">1. Begin adding new entity.2. Upload your first image and tag it <code>banner, vimeo</code>3. Upload your second image and tag it <code>thumb image</code>4. Create a copy paragraph with <code>apple</code> as the value and tag it with <code>apple</code>.5. Create another copy paragraph with <code>banana</code> as the value and tag it with <code>banana</code>.6. Save the entity.7. Run <i>Snippet A</i> replacing <code>\$nid</code> with the node ID of the entity you created.	FAIL PASS Assert the output is 'passed'
<ol style="list-style-type: none">8. Run <i>Snippet B</i> replacing <code>\$nid</code> with the node ID of the entity you created.	FAIL PASS Assert the output is 'passed'

Test Steps	Test Results
9. Reload the edit form for the entity.	<div>FAIL PASS Assert the image is tagged with foo bara .</div> <div>FAIL PASS Assert field_main_copy has the correct tags in place.</div>

Code Snippets

Snippet A

Make sure we can programmatically access our tagged items.

```
<?php
$node = \Drupal\node\Entity\Node::load(18733);
$service = \Drupal::service('field_tag')->attachTags($node);

// Assert we can access a CSV tag.
$vimeos = $service->getItemsTaggedBy('vimeo', 'field_images');
1 === count($vimeos) || die('vimeo failed');

// Assert we can access another CSV tag.
$banners = $service->getItemsTaggedBy('banner', 'field_images');
1 === count($banners) || die('banner failed');

// Assert we can access a single tag with a space.
$thumbs = $service->getItemsTaggedBy('thumb image', 'field_images');
1 === count($thumbs) || die('thumb image failed');

// Assert the fieldTag instance.
// Assert the key is 1, which matches the delta of the field item.
$result = $thumbs[1]->fieldTag instanceof \Drupal\field_tag\Entity\FieldTag;
$result || die('no fieldTag instance');

die('passed');
```

Snippet B

```

<?php
$node = \Drupal\node\Entity\Node::load( 18733 );
$service = \Drupal::service('field_tag')->attachTags($node);
$i = $service->getItemsTaggedBy('thumb image', 'field_images');
$item = array_first($i)->getValue();
$item['field_tag'] = 'foo bara';
$node->field_images->filter(function () {
    return FALSE;
})->appendItem($item);
$node->save();
$service = \Drupal::service('field_tag')->attachTags($node);

$thumbs = $service->getItemsTaggedBy('thumb image', 'field_images');
0 === count($thumbs) || die('thumb image failed');

$foobars = $service->getItemsTaggedBy('foo bara', 'field_images');
1 === count($foobars) || die('foo bara failed');

die('passed');

```

Snippet C

```

<?php
$node = \Drupal\node\Entity\Node::load( 18733 );
$service = \Drupal::service('field_tag')->attachTags($node);
$item = array_first($service->getItemsTaggedBy('thumb image', 'field_images'))->getValue

// Assert 'field_tag' replaces when an array.
$item = $alphas[0]->getValue();
$item['field_tag'] = ['tag' => 'bravo charlie, delta'];
$node->field_images->filter(function () {
    return FALSE;
})->appendItem($item);
$node->save();
$service = \Drupal::service('field_tag')->attachTags($node);

$thumb_images = $service->getItemsTaggedBy('thumb image', 'field_images');
0 === count($thumb_images) || die('thumb image failed');
$bravos = $service->getItemsTaggedBy('bravo charlie', 'field_images');
1 === count($bravos) || die('bravo charlie failed');

```

Comments

Please mention browser, device type, where you logged in, and anything else that can help diagnose failure.

Test completed by: Aaron Klump

Test Case: unsaved-multi-image

FAILPASS

Test Scenario

Tagging more than one image in a filefield works.

Pre-Conditions

- 1. An entity type with an image field.
- 2. The image field has field tagging enabled.
- 3. The image field allows unlimited images.

Test Data

tag1	alpha
------	-------

Test Execution

Test Steps	Test Results
<div>1. Begin adding new entity.</div> <div>2. Upload your first image and tag it alpha</div> <div>3. Upload your second image</div>	<div>FAILPASS</div> Assert alpha is still present in the first image's text field.

Comments

Please mention browser, device type, where you logged in, and anything else that can help diagnose failure.

Test completed by: Aaron Klump