---+ Package `TWiki::Meta`

All TWiki topics have **data** (text) and **meta-data** (information about the topic). Meta-data includes information such as file attachments, form fields, topic parentage etc. When TWiki loads a topic from the store, it represents the meta-data in the topic using an object of this class.

A meta-data object is a hash of different types of meta-data (keyed on the type, such as 'FIELD' and 'TOPICINFO').

Each entry in the hash is an array, where each entry in the array contains another hash of the key=value pairs, corresponding to a single meta-datum.

If there may be multiple entries of the same top-level type (i.e. for FIELD and FILEATTACHMENT) then the array has multiple entries. These types are referred to as "keyed" types. The array entries are keyed with the attribute 'name' which must be in each entry in the array.

For unkeyed types, the array has only one entry.

Pictorially,

- TOPICINFO
    - author => '...'
    - date => '...'
    - ...
- FILEATTACHMENT
    - [0] -> { name => '...' ... }
    - [1] -> { name => '...' ... }
- FIELD
    - [0] -> { name => '...' ... }
    - [1] -> { name => '...' ... }

As well as the meta-data, the object also stores the web name, topic name and remaining text after meta-data extraction.

ClassMethod new (`$session,$web,$topic,$text`)

- `$session` - a TWiki object (e.g. =$TWiki::Plugins::SESSION)
- `$web, $topic` - the topic that the metadata relates to

Construct a new, empty object to contain meta-data for the given topic.

- $text - optional raw text to convert to meta-data form

ObjectMethod finish `()`

Break circular references.

ObjectMethod session `()`

Get the session

ObjectMethod web `()`

Get the web name

ObjectMethod topic `()`

Get the topic name

ObjectMethod text `([$text]) -> $text`

Get/set the topic body text. If $text is undef, gets the value, if it is defined, sets the value to that and returns the new text.

ObjectMethod put `($type,\%args)`

Put a hash of key=value pairs into the given type set in this meta. This will **not** replace another value with the same name (for that see `putKeyed`)

For example,

```
$meta->put( 'FIELD', { name => 'MaxAge', title => 'Max Age', value =>'103' } );
```

ObjectMethod putKeyed `($type,\%args)`

Put a hash of key=value pairs into the given type set in this meta, replacing any existing value with the same key.

For example,

```
$meta->putKeyed( 'FIELD', { name => 'MaxAge', title => 'Max Age', value =>'103' } );
```

ObjectMethod putAll

Replaces all the items of a given key with a new array.

For example,

```
$meta->putAll( 'FIELD',
     { name => 'MinAge', title => 'Min Age', value =>'50' },
     { name => 'MaxAge', title => 'Max Age', value =>'103' },
     { name => 'HairColour', title => 'Hair Colour', value =>'white' }
 );
```

ObjectMethod get `($type,$key) -> \%hash`

Find the value of a meta-datum in the map. If the type is keyed (idenitifed by a `name`), the `$key` parameter is required to say *which* entry you want. Otherwise you will just get the first value.

If you want all the keys of a given type use the 'find' method.

The result is a reference to the hash for the item.

For example,

```
my $ma = $meta->get( 'FIELD', 'MinAge' );
my $topicinfo = $meta->get( 'TOPICINFO' ); # get the TOPICINFO hash
```

ObjectMethod find (`$type`) -> `@values`

Get all meta data for a specific type. Returns the array stored for the type. This will be zero length if there are no entries.

For example,

```
my $attachments = $meta->find( 'FILEATTACHMENT' );
```

ObjectMethod remove (`$type,$key`)

With no type, will remove all the contents of the object.

With a $type but no $key, will remove *all* items of that type (so for example if $type were FILEATTACHMENT it would remove all of them)

With a $type and a $key it will remove only the specific item.

ObjectMethod copyFrom (`$otherMeta,$type,$nameFilter`)

Copy all entries of a type from another meta data set. This will destroy the old values for that type, unless the copied object doesn't contain entries for that type, in which case it will retain the old values.

If $type is undef, will copy ALL TYPES.

If $nameFilter is defined (a perl refular expression), it will copy only data where `{name}` matches $nameFilter.

Does **not** copy web, topic or text.

ObjectMethod count (`$type`) -> `$integer`

Return the number of entries of the given type

ObjectMethod getRevisionInfo (`$fromrev`) -> (`$date,$author,$rev,$comment`)

Try and get revision info from the meta information, or, if it is not present, kick down to the Store module for the same information.

Returns ($revDate, $author, $rev, $comment)

$rev is an integer revision number.

ObjectMethod merge (`$otherMeta,$formDef`)

- `$otherMeta` - a block of meta-data to merge with $this
- `$formDef` reference to a TWiki::Form that gives the types of the fields in $this

Merge the data in the other meta block.

- File attachments that only appear in one set are preserved.
- Form fields that only appear in one set are preserved.
- Form field values that are different in each set are text-merged
- We don't merge for field attributes or title
- Topic info is not touched
- The `mergeable` method on the form def is used to determine if that fields is mergeable. if it isn't, the value currently in meta will *not* be changed.

ObjectMethod stringify (`$types`) -> `$string`

Return a string version of the meta object. Uses \n to separate lines. If `$types` is specified, return only types that match it. Types should be a perl regular expression.

ObjectMethod forEachSelectedValue (`$types,$keys,\&fn,\%options`)

Iterate over the values selected by the regular expressions in $types and $keys.

- `$types` - regular expression matching the names of fields to be processed. Will default to qr/^[A-Z]+$/ if undef.
- `$keys` - regular expression matching the names of keys to be processed. Will default to qr/^[a-z]+$/ if undef.

Iterates over each value, calling `\&fn` on each, and replacing the value with the result of \&fn.

\%options will be passed on to $fn, with the following additions:

- `_type` => the type name (e.g. "FILEATTACHMENT")
- `_key` => the key name (e.g. "user")

ObjectMethod getParent () -> `$parent`

Gets the TOPICPARENT name.

ObjectMethod getFormName () -> `$formname`

Returns the name of the FORM, or " if none.

ObjectMethod renderFormForDisplay () -> `$html`

Render the form contained in the meta for display.

ObjectMethod renderFormFieldForDisplay `($name,$format,$attrs) -> $text`

Render a single formfield, using the $format. See TWiki::Form::FormField::renderForDisplay for a description of how the value is rendered.

ObjectMethod getEmbeddedStoreForm `() -> $text`

Generate the embedded store form of the topic. The embedded store form has meta-data values embedded using %META: lines. The text stored in the meta is taken as the topic text.

ObjectMethod getMetaFor `() -> $meta`

This method will load (or otherwise fetch) the meta-data for a named web/topic. The request might be satisfied by a read from the store, or it might be satisfied from a cache. The caller doesn't care.

This is an object method rather than a static method because it depends on the implementation of Meta - it might be this base class, or it might be a caching subclass, for example.

This topic: TWiki > TWikiMetaDotPm
Topic revision: r4 - 2008-01-22 - TWikiContributor