patchit Documentation

Release 1.0

eisensheng

January 18, 2013

CONTENTS

exception `patchit.`**`PatchSyntaxError`**
> Thrown whenever the parser chokes on a malformed unified diff.

exception `patchit.`**`PatchConflictError`**
> Thrown by the hunk merge logic whenever the merge failed.

class `patchit.`**`Hunk`**(*source_range*, *target_range*, *comment=None*)
> Collection of operations concerning an isolated file chunk.

> **`merge`**(*lines*)
> > Merge Hunk into *lines*.

> > > **Parameters** **lines** (*generator*) – isolated collection of lines on which the hunk should be applied to.

> > > **Raises** `PatchConflictError`

> **class** **`Operation`**
> > Operation(symbol, text)

> > Describes a single instruction within a hunk.

> > **`symbol`**
> > > Specifies whenever the given text shall match, be deleted from or inserted into the given stream.

> > **`text`**
> > > Specifies the text to insert or match for equality or deletion

class `patchit.`**`Patch`**(*source_filename*, *target_filename*, *source_comment=None*, *target_comment=None*, *metadata=None*)
> Collection of Hunks concerning a single file.

> **`hunks`**
> > Ordered collection of `Hunk` instances.

> **`source_filename`**
> > Old file name

> **`source_comment`**
> > Comment relating old file name (i.e. last time of change)

> **`target_filename`**
> > New file name

> **`target_comment`**
> > Comment relating new file name (i.e. last time of change)

> **`metadata`**
> > VCS related metadata (like diff parameters or commit id)

> **`merge`**(*lines*)
> > Merges entire hunk collection into *lines*.

> > > **Parameters** **lines** (*generator*) – collection of lines on which the patch should be applied.

> > > **Raises** `PatchConflictError`

class `patchit.`**`PatchSet`**
> Collection of Patches.

> **`patches`**
> > Collection of `Patch` instances.

> **classmethod** **`from_stream`**(*in_stream*)
> > Reads from *in_stream* and return the parsed patch set.

> > > **Parameters in_stream** (*file*) – stream containing a patch set.

class `patchit.`**`PatchSetReader`**

> Stateful reader parsing arbitrary patches and hunks.

> **`feed`**(*lines*)

> > Parses set of by newline separated lines describing a patch set.

> > > **Parameters lines** (*generator*) – collection of newline terminated lines

> > > **Raises** `PatchSyntaxError`

> **`patch_set`**

> > `PatchSet` containing parsed content.

> > > **Return type** `PatchSet`

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX