

Redmine - Feature #33784

Update Mercurial repository support to be compatible with Python 3 and remove support for Mercurial < 5.1

2020-07-29 16:26 - Harald Klimach

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	SCM	Estimated time:	0.00 hour
Target version:	6.1.0		
Resolution:	Fixed		

Description

As Mercurial is [transitioning to Python 3](#), I updated my Mercurial version now to make use of Python 3 and ran into trouble with the RedmineHelper. Mainly due the string handling, as the Mercurial API requires bytes objects instead of strings.

Here is a patch, that changes the helper to work with Mercurial on Python 3, though it is not compatible with Python2 then anymore. There probably are better ways to implement this, but it works for me in this form.

```
Index: lib/redmine/scm/adapters/mercurial/redminehelper.py
=====
--- lib/redmine/scm/adapters/mercurial/redminehelper.py      (revision 19937)
+++ lib/redmine/scm/adapters/mercurial/redminehelper.py      (working copy)
@@ -45,14 +45,14 @@
     </repository>
     </rhmanifest>
     """
-import re, time, cgi, urllib
+import re, time, html, urllib
     from mercurial import cmdutil, commands, node, error, hg, registrar

     cmdtable = {}
     command = registrar.command(cmdtable) if hasattr(registrar, 'command') else cmdutil.command(cmdtable)

     _x = cgi.escape
     _u = lambda s: cgi.escape(urllib.quote(s))
+    _x = lambda s: html.escape(s.decode('utf-8')).encode('utf-8')
+    _u = lambda s: html.escape(urllib.parse.quote(s)).encode('utf-8')

     def _changectx(repo, rev):
         if isinstance(rev, str):
@@ -70,10 +70,10 @@
             except TypeError: # Mercurial < 1.1
                 return repo.changelog.count() - 1
             tipctx = _changectx(repo, tiprev())
-            ui.write('<tip revision="%d" node="%s"/>\n'
+            ui.write(b'<tip revision="%d" node="%s"/>\n'
                 % (tipctx.rev(), _x(node.hex(tipctx.node()))))

     _SPECIAL_TAGS = ('tip',)
+    _SPECIAL_TAGS = (b'tip',)

     def _tags(ui, repo):
         # see mercurial/commands.py:tags
@@ -84,7 +84,7 @@
         r = repo.changelog.rev(n)
         except error.LookupError:
             continue
-            ui.write('<tag revision="%d" node="%s" name="%s"/>\n'
+            ui.write(b'<tag revision="%d" node="%s" name="%s"/>\n'
                 % (r, _x(node.hex(n)), _x(t)))
```

```

def _branches(ui, repo):
@@ -104,66 +104,67 @@
    return repo.branchheads(branch)
    def lookup(rev, n):
        try:
-            return repo.lookup(rev)
+            return repo.lookup(str(rev).encode('utf-8'))
        except RuntimeError:
            return n
    for t, n, r in sorted(iterbranches(), key=lambda e: e[2], reverse=True):
        if lookup(r, n) in branchheads(t):
-            ui.write('<branch revision="%d" node="%s" name="%s"/>\n'
+            ui.write(b'<branch revision="%d" node="%s" name="%s"/>\n'
                        % (r, _x(node.hex(n)), _x(t)))

def _manifest(ui, repo, path, rev):
    ctx = _changectx(repo, rev)
-    ui.write('<manifest revision="%d" path="%s">\n'
+    ui.write(b'<manifest revision="%d" path="%s">\n'
              % (ctx.rev(), _u(path)))

    known = set()
-    pathprefix = (path.rstrip('/') + '/').lstrip('/')
+    pathprefix = (path.decode('utf-8').rstrip('/') + '/').lstrip('/')
    for f, n in sorted(ctx.manifest().iteritems(), key=lambda e: e[0]):
-        if not f.startswith(pathprefix):
+        fstr = f.decode('utf-8')
+        if not fstr.startswith(pathprefix):
            continue
-        name = re.sub(r'/.*', '/', f[len(pathprefix):])
+        name = re.sub(r'/.*', '/', fstr[len(pathprefix):])
        if name in known:
            continue
        known.add(name)

        if name.endswith('/'):
-            ui.write('<dir name="%s"/>\n'
-                    % _x(urllib.quote(name[:-1])))
+            ui.write(b'<dir name="%s"/>\n'
                      % _x(urllib.parse.quote(name[:-1]).encode('utf-8')))
        else:
            fctx = repo.filectx(f, fileid=n)
            tm, tzoffset = fctx.date()
            ui.write('<file name="%s" revision="%d" node="%s" '
                    'time="%d" size="%d"/>\n'
+            ui.write(b'<file name="%s" revision="%d" node="%s" '
                      b'time="%d" size="%d"/>\n'
                      % (_u(name), fctx.rev(), _x(node.hex(fctx.node())),
                        tm, fctx.size(), ))

-    ui.write('</manifest>\n')
+    ui.write(b'</manifest>\n')

-@command('rhannotate',
-        [ ('r', 'rev', '', 'revision'),
-        ('u', 'user', None, 'list the author (long with -v)'),
-        ('n', 'number', None, 'list the revision number (default)'),
-        ('c', 'changeset', None, 'list the changeset'),
+@command(b'rhannotate',
+        [ (b'r', b'rev', b'', b'revision'),
+        (b'u', b'user', None, b'list the author (long with -v)'),
+        (b'n', b'number', None, b'list the revision number (default)'),
+        (b'c', b'changeset', None, b'list the changeset'),
+        ],
-        'hg rhannotate [-r REV] [-u] [-n] [-c] FILE...')
+        b'hg rhannotate [-r REV] [-u] [-n] [-c] FILE...')

```

```

def rhannotate(ui, repo, *pats, **opts):
-     rev = urllib.unquote_plus(opts.pop('rev', None))
+     rev = urllib.parse.unquote_plus(opts.pop('rev', None))
    opts['rev'] = rev
-     return commands.annotate(ui, repo, *map(urllib.unquote_plus, pats), **opts)
+     return commands.annotate(ui, repo, *map(urllib.parse.unquote_plus, pats), **opts)

-@command('rhcat',
-         [ ('r', 'rev', '', 'revision') ],
-         'hg rhcat ([-r REV] ...) FILE...')
+@command(b'rhcat',
+         [ (b'r', b'rev', b'', b'revision') ],
+         b'hg rhcat ([-r REV] ...) FILE...')
    def rhcat(ui, repo, file1, *pats, **opts):
-     rev = urllib.unquote_plus(opts.pop('rev', None))
+     rev = urllib.parse.unquote_plus(opts.pop('rev', None))
    opts['rev'] = rev
-     return commands.cat(ui, repo, urllib.unquote_plus(file1), *map(urllib.unquote_plus, pats), **
opts)
+     return commands.cat(ui, repo, urllib.parse.unquote_plus(file1), *map(urllib.parse.unquote_plus, pats), **opts)

-@command('rhdiff',
-         [ ('r', 'rev', [], 'revision'),
-           ('c', 'change', '', 'change made by revision') ],
-         'hg rhdiff ([-c REV] | [-r REV] ...) [FILE]...')
+@command(b'rhdiff',
+         [ (b'r', b'rev', [], b'revision'),
+           (b'c', b'change', b'', b'change made by revision') ],
+         b'hg rhdiff ([-c REV] | [-r REV] ...) [FILE]...')
    def rhdiff(ui, repo, *pats, **opts):
        """diff repository (or selected files)"""
        change = opts.pop('change', None)
@@ -171,34 +172,34 @@
        base = _changectx(repo, change).parents()[0].rev()
        opts['rev'] = [str(base), change]
        opts['nodates'] = True
-     return commands.diff(ui, repo, *map(urllib.unquote_plus, pats), **opts)
+     return commands.diff(ui, repo, *map(urllib.parse.unquote_plus, pats), **opts)

-@command('rhlog',
+@command(b'rhlog',
+         [
-             ('r', 'rev', [], 'show the specified revision'),
-             ('b', 'branch', [],
-              'show changesets within the given named branch'),
-             ('l', 'limit', '',
-              'limit number of changes displayed'),
-             ('d', 'date', '',
-              'show revisions matching date spec'),
-             ('u', 'user', [],
-              'revisions committed by user'),
-             ('', 'from', '',
-              ''),
-             ('', 'to', '',
-              ''),
-             ('', 'rhbranch', '',
-              ''),
-             ('', 'template', '',
-              'display with template') ],
-         'hg rhlog [OPTION]... [FILE]')
+             (b'r', b'rev', [], b'show the specified revision'),
+             (b'b', b'branch', [],
+              b'show changesets within the given named branch'),
+             (b'l', b'limit', b'',
+              b'limit number of changes displayed'),
+             (b'd', b'date', b'',
+              b'show revisions matching date spec'),
+             (b'u', b'user', [],
+              b'revisions committed by user'),
+             (b'', b'from', b'',
+              b''),
+             (b'', b'to', b'',
+              b''),
+             (b'', b'rhbranch', b'',
+              b''),
+             (b'', b'template', b'',
+              b'display with template') ],
+         b'hg rhlog [OPTION]... [FILE]')

```

```

+             b'show revisions matching date spec'),
+             (b'u', b'user', [],
+              b'revisions committed by user'),
+             (b'', b'from', b'',
+              b''),
+             (b'', b'to', b'',
+              b''),
+             (b'', b'rhbranch', b'',
+              b''),
+             (b'', b'template', b'',
+              b'display with template')),
+             b'hg rhlog [OPTION]... [FILE]')
def rhlog(ui, repo, *pats, **opts):
    rev = opts.pop('rev')
    bra0 = opts.pop('branch')
-   from_rev = urllib.unquote_plus(opts.pop('from', None))
-   to_rev = urllib.unquote_plus(opts.pop('to', None))
-   bra = urllib.unquote_plus(opts.pop('rhbranch', None))
+   from_rev = urllib.parse.unquote_plus(opts.pop('from', None))
+   to_rev = urllib.parse.unquote_plus(opts.pop('to', None))
+   bra = urllib.parse.unquote_plus(opts.pop('rhbranch', None))
    from_rev = from_rev.replace('"', '\\')
    to_rev = to_rev.replace('"', '\\')
    if hg.util.version() >= '1.6':
@@ -206,28 +207,30 @@
        else:
            opts['rev'] = ['%s:%s' % (from_rev, to_rev)]
            opts['branch'] = [bra]
-   return commands.log(ui, repo, *map(urllib.unquote_plus, pats), **opts)
+   return commands.log(ui, repo, *map(urllib.parse.unquote_plus, pats), **opts)

-@command('rhmanifest',
-         [(('r', 'rev', '', 'show the specified revision')],
-          'hg rhmanifest [-r REV] [PATH]')
-def rhmanifest(ui, repo, path='', **opts):
+@command(b'rhmanifest',
+         [(b'r', b'rev', b'', b'show the specified revision')],
+         b'hg rhmanifest [-r REV] [PATH]')
+def rhmanifest(ui, repo, path=b'', **opts):
    """output the sub-manifest of the specified directory"""
-   ui.write('<?xml version="1.0"?>\n')
-   ui.write('<rhmanifest>\n')
-   ui.write('<repository root="%s">\n' % _u(repo.root))
+   ui.write(b'<?xml version="1.0"?>\n')
+   ui.write(b'<rhmanifest>\n')
+   ui.write(b'<repository root="%s">\n' % _u(repo.root))
    try:
-   _manifest(ui, repo, urllib.unquote_plus(path), urllib.unquote_plus(opts.get('rev')))
+   _manifest(ui, repo,
+             urllib.parse.unquote_plus(path.decode('utf-8')).encode('utf-8'),
+             urllib.parse.unquote_plus(opts.get('rev').decode('utf-8')).encode('utf-8'))
    finally:
-   ui.write('</repository>\n')
-   ui.write('</rhmanifest>\n')
+   ui.write(b'</repository>\n')
+   ui.write(b'</rhmanifest>\n')

-@command('rhsummary', [], 'hg rhsummary')
+@command(b'rhsummary', [], b'hg rhsummary')
def rhsummary(ui, repo, **opts):
    """output the summary of the repository"""
-   ui.write('<?xml version="1.0"?>\n')
-   ui.write('<rhsummary>\n')
-   ui.write('<repository root="%s">\n' % _u(repo.root))
+   ui.write(b'<?xml version="1.0"?>\n')
+   ui.write(b'<rhsummary>\n')
+   ui.write(b'<repository root="%s">\n' % _u(repo.root))

```

```

    try:
        _tip(ui, repo)
        _tags(ui, repo)
@@ -234,6 +237,6 @@
        _branches(ui, repo)
        # TODO: bookmarks in core (Mercurial>=1.8)
    finally:
-        ui.write('</repository>\n')
-        ui.write('</rhsummary>\n')
+        ui.write(b'</repository>\n')
+        ui.write(b'</rhsummary>\n')

```

It would be great, if Redmine could support Mercurial on Python 3. Maybe the [mercurial.pycompat](#) module could be of help to create a patch that does not break backwards compatibility.

Related issues:

Related to Redmine - Patch #42500: Skip repository tests when the SCM client ...	Closed
Related to Redmine - Patch #42597: Skip some Mercurial tests when using Mercu...	Closed
Related to Redmine - Defect #33108: Mercurial support is broken for Mercurial...	Closed

Associated revisions

Revision 23513 - 2025-02-16 09:43 - Go MAEDA

Update Mercurial repository support to be compatible with Python 3 and remove support for Mercurial < 5.1 (#33784).

Patch by Harald Klimach (user:haraldkl), Olivier Houdas (user:olivier.houdas@geoconcept.com), Jakob Haufe (user:sur5r), and Sean Baggaley (user:NotTheActualSean).

Revision 23687 - 2025-04-20 04:22 - Go MAEDA

Ensure Mercurial uses Python >= 3.5 (#33784).

Redmine no longer supports Python 2.7.
Mercurial does not support Python 3.0 to 3.4.

Patch by Go MAEDA (user:maeda).

History

#1 - 2020-07-31 16:52 - Harald Klimach

The patch above does not work properly for everything (file views and diffs of files), so I had to revise it a little bit. Still this changed version only supports Mercurial on Python 3:

```

Index: lib/redmine/scm/adapters/mercurial/redminehelper.py
=====
--- lib/redmine/scm/adapters/mercurial/redminehelper.py      (revision 19937)
+++ lib/redmine/scm/adapters/mercurial/redminehelper.py      (working copy)
@@ -45,17 +45,20 @@
     </repository>
     </rhmanifest>
 """
-import re, time, cgi, urllib
+import re, time, cgi, urllib
+import re, time, html, urllib
+from mercurial import cmdutil, commands, node, error, hg, registrar

cmdtable = {}
command = registrar.command(cmdtable) if hasattr(registrar, 'command') else cmdutil.command(cmdtable)

-_x = cgi.escape
-_u = lambda s: cgi.escape(urllib.quote(s))
+_x = lambda s: html.escape(s.decode('utf-8')).encode('utf-8')
+_u = lambda s: html.escape(urllib.parse.quote(s)).encode('utf-8')

+def unquoteplus(*args, **kwargs):
+    return urllib.parse.unquote_to_bytes(*args, **kwargs).replace(b'+', b' ')
+
+def _changectx(repo, rev):
-    if isinstance(rev, str):
+    if isinstance(rev, bytes):
+        rev = repo.lookup(rev)
+    if hasattr(repo, 'changectx'):

```

```

        return repo.changectx(rev)
@@ -70,10 +73,10 @@
        except TypeError: # Mercurial < 1.1
            return repo.changelog.count() - 1
        tipctx = _changectx(repo, tiprev())
- ui.write('<tip revision="%d" node="%s"/>\n'
+ ui.write(b'<tip revision="%d" node="%s"/>\n'
            % (tipctx.rev(), _x(node.hex(tipctx.node()))))

-_SPECIAL_TAGS = ('tip',)
+_SPECIAL_TAGS = (b'tip',)

def _tags(ui, repo):
    # see mercurial/commands.py:tags
@@ -84,7 +87,7 @@
        r = repo.changelog.rev(n)
        except error.LookupError:
            continue
- ui.write('<tag revision="%d" node="%s" name="%s"/>\n'
+ ui.write(b'<tag revision="%d" node="%s" name="%s"/>\n'
            % (r, _x(node.hex(n)), _x(t)))

def _branches(ui, repo):
@@ -104,130 +107,140 @@
        return repo.branchheads(branch)
    def lookup(rev, n):
        try:
-         return repo.lookup(rev)
+         return repo.lookup(str(rev).encode('utf-8'))
        except RuntimeError:
            return n
    for t, n, r in sorted(iterbranches(), key=lambda e: e[2], reverse=True):
        if lookup(r, n) in branchheads(t):
-         ui.write('<branch revision="%d" node="%s" name="%s"/>\n'
+         ui.write(b'<branch revision="%d" node="%s" name="%s"/>\n'
                    % (r, _x(node.hex(n)), _x(t)))

def _manifest(ui, repo, path, rev):
    ctx = _changectx(repo, rev)
- ui.write('<manifest revision="%d" path="%s">\n'
+ ui.write(b'<manifest revision="%d" path="%s">\n'
            % (ctx.rev(), _u(path)))

    known = set()
- pathprefix = (path.rstrip('/') + '/').lstrip('/')
+ pathprefix = (path.decode('utf-8').rstrip('/') + '/').lstrip('/')
    for f, n in sorted(ctx.manifest().iteritems(), key=lambda e: e[0]):
        if not f.startswith(pathprefix):
+         fstr = f.decode('utf-8')
+         if not fstr.startswith(pathprefix):
            continue
-         name = re.sub(r'/.*', '/', f[len(pathprefix):])
+         name = re.sub(r'/.*', '/', fstr[len(pathprefix):])
        if name in known:
            continue
        known.add(name)

        if name.endswith('/'):
-         ui.write('<dir name="%s"/>\n'
+         ui.write(b'<dir name="%s"/>\n'
                    % _x(urllib.quote(name[:-1])))
+         ui.write(b'<dir name="%s"/>\n'
                    % _x(urllib.parse.quote(name[:-1]).encode('utf-8')))
        else:
            fctx = repo.filectx(f, fileid=n)
            tm, tzoffset = fctx.date()
-         ui.write('<file name="%s" revision="%d" node="%s" '
+         ui.write(b'<file name="%s" revision="%d" node="%s" '
                    'time="%d" size="%d"/>\n'
+         ui.write(b'<file name="%s" revision="%d" node="%s" '
                    b'time="%d" size="%d"/>\n'
                    % (_u(name), fctx.rev(), _x(node.hex(fctx.node())),
                      tm, fctx.size(), ))

- ui.write('</manifest>\n')
+ ui.write(b'</manifest>\n')

```

```

-@command('rhannotate',
-        [('r', 'rev', '', 'revision'),
-         ('u', 'user', None, 'list the author (long with -v)'),
-         ('n', 'number', None, 'list the revision number (default)'),
-         ('c', 'changeset', None, 'list the changeset'),
+@command(b'rhannotate',
+        [(b'r', b'rev', b'', b'revision'),
+         (b'u', b'user', None, b'list the author (long with -v)'),
+         (b'n', b'number', None, b'list the revision number (default)'),
+         (b'c', b'changeset', None, b'list the changeset'),
+        ],
-        'hg rhannotate [-r REV] [-u] [-n] [-c] FILE...')
+        b'hg rhannotate [-r REV] [-u] [-n] [-c] FILE...')
def rhannotate(ui, repo, *pats, **opts):
-    rev = urllib.unquote_plus(opts.pop('rev', None))
+    rev = unquoteplus(opts.pop('rev', b''))
    opts['rev'] = rev
-    return commands.annotate(ui, repo, *map(urllib.unquote_plus, pats), **opts)
+    return commands.annotate(ui, repo, *map(unquoteplus, pats), **opts)

-@command('rhcat',
-        [('r', 'rev', '', 'revision')],
-        'hg rhcat ([-r REV] ...) FILE...')
+@command(b'rhcat',
+        [(b'r', b'rev', b'', b'revision')],
+        b'hg rhcat ([-r REV] ...) FILE...')
def rhcat(ui, repo, file1, *pats, **opts):
-    rev = urllib.unquote_plus(opts.pop('rev', None))
+    rev = unquoteplus(opts.pop('rev', b''))
    opts['rev'] = rev
-    return commands.cat(ui, repo, urllib.unquote_plus(file1), *map(urllib.unquote_plus, pats), **opts)
+    return commands.cat(ui, repo, unquoteplus(file1), *map(unquoteplus, pats), **opts)

-@command('rhdiff',
-        [('r', 'rev', [], 'revision'),
-         ('c', 'change', '', 'change made by revision')],
-        'hg rhdiff ([-c REV] | [-r REV] ...) [FILE]...')
+@command(b'rhdiff',
+        [(b'r', b'rev', [], b'revision'),
+         (b'c', b'change', b'', b'change made by revision')],
+        b'hg rhdiff ([-c REV] | [-r REV] ...) [FILE]...')
def rhdiff(ui, repo, *pats, **opts):
    """diff repository (or selected files)"""
    change = opts.pop('change', None)
    if change: # add -c option for Mercurial<1.1
        base = _changectx(repo, change).parents()[0].rev()
        opts['rev'] = [str(base), change]
+    opts['rev'] = [base, change]
    opts['nodates'] = True
-    return commands.diff(ui, repo, *map(urllib.unquote_plus, pats), **opts)
+    return commands.diff(ui, repo, *map(unquoteplus, pats), **opts)

-@command('rhlog',
+@command(b'rhlog',
+        [
-            ('r', 'rev', [], 'show the specified revision'),
-            ('b', 'branch', [],
-             'show changesets within the given named branch'),
-            ('l', 'limit', '',
-             'limit number of changes displayed'),
-            ('d', 'date', '',
-             'show revisions matching date spec'),
-            ('u', 'user', [],
-             'revisions committed by user'),
-            ('', 'from', '',
-             ''),
-            ('', 'to', '',
-             ''),
-            ('', 'rhbranch', '',
-             ''),
-            ('', 'template', '',
-             'display with template')],
-        'hg rhlog [OPTION]... [FILE]')
+            (b'r', b'rev', [], b'show the specified revision'),
+            (b'b', b'branch', [],
+             b'show changesets within the given named branch'),
+            (b'l', b'limit', b'',
+             b'limit number of changes displayed'),
+            (b'd', b'date', b'',
+             b'show revisions matching date spec'),
+            (b'u', b'user', [],
+             b'revisions committed by user'),
+            (b'', b'from', b'',
+             b''),
+            (b'', b'to', b'',
+             b''),
+            (b'', b'rhbranch', b'',
+             b''),
+            (b'', b'template', b'',
+             b'display with template')],
+            b'hg rhlog [OPTION]... [FILE]')
+        (b'r', b'rev', [], b'show the specified revision'),
+        (b'b', b'branch', [],

```

```

+         b'show changesets within the given named branch'),
+         (b'l', b'limit', b'',
+         b'limit number of changes displayed'),
+         (b'd', b'date', b'',
+         b'show revisions matching date spec'),
+         (b'u', b'user', [],
+         b'revisions committed by user'),
+         (b'', b'from', b'',
+         b''),
+         (b'', b'to', b'',
+         b''),
+         (b'', b'rhbranch', b'',
+         b''),
+         (b'', b'template', b'',
+         b'display with template')],
+         b'hg rhlog [OPTION]... [FILE]')
def rhlog(ui, repo, *pats, **opts):
    rev = opts.pop('rev')
    bra0 = opts.pop('branch')
    from_rev = urllib.unquote_plus(opts.pop('from', None))
    to_rev = urllib.unquote_plus(opts.pop('to', None))
    bra = urllib.unquote_plus(opts.pop('rhbranch', None))
    from_rev = from_rev.replace("'", '\\\\')
    to_rev = to_rev.replace("'", '\\\\')
    if hg.util.version() >= '1.6':
        opts['rev'] = ['"%s":"%s"' % (from_rev, to_rev)]
    from_rev = unquoteplus(opts.pop('from', b''))
    to_rev = unquoteplus(opts.pop('to', b''))
    bra = unquoteplus(opts.pop('rhbranch', b''))
    from_rev = from_rev.replace(b'\\', b'\\')
    to_rev = to_rev.replace(b'\\', b'\\')
    if (from_rev != b'') or (to_rev != b''):
        if from_rev != b'':
            quotefrom = b'"%s"' % (from_rev)
        else:
            quotefrom = from_rev
        if to_rev != b'':
            quoteto = b'"%s"' % (to_rev)
        else:
            quoteto = to_rev
        opts['rev'] = [b'%s:%s' % (quotefrom, quoteto)]
    else:
        opts['rev'] = ['%s:%s' % (from_rev, to_rev)]
    opts['branch'] = [bra]
    return commands.log(ui, repo, *map(urllib.unquote_plus, pats), **opts)
    opts['rev'] = rev
    if (bra != b''):
        opts['branch'] = [bra]
    return commands.log(ui, repo, *map(unquoteplus, pats), **opts)

-@command('rhmanifest',
-        [ ('r', 'rev', '', 'show the specified revision') ],
-        'hg rhmanifest [-r REV] [PATH]')
-def rhmanifest(ui, repo, path='', **opts):
+@command(b'rhmanifest',
+        [ (b'r', b'rev', b'', b'show the specified revision') ],
+        b'hg rhmanifest -r REV [PATH]')
+def rhmanifest(ui, repo, path=b'', **opts):
    """output the sub-manifest of the specified directory"""
    ui.write('<?xml version="1.0"?>\n')
    ui.write('<rhmanifest>\n')
    ui.write('<repository root="%s">\n' % _u(repo.root))
    ui.write(b'<?xml version="1.0"?>\n')
    ui.write(b'<rhmanifest>\n')
    ui.write(b'<repository root="%s">\n' % _u(repo.root))
    try:
        _manifest(ui, repo, urllib.unquote_plus(path), urllib.unquote_plus(opts.get('rev')))
    +    _manifest(ui, repo, unquoteplus(path), unquoteplus(opts.get('rev')))
    finally:
        ui.write('</repository>\n')
        ui.write('</rhmanifest>\n')
        ui.write(b'</repository>\n')
        ui.write(b'</rhmanifest>\n')

-@command('rhsummary', [], 'hg rhsummary')

```



```

+@command(b'rhsummary', [], b'hg rhsummary')
def rhsummary(ui, repo, **opts):
    """output the summary of the repository"""
-   ui.write('<?xml version="1.0"?>\n')
-   ui.write('<rhsummary>\n')
-   ui.write('<repository root="%s">\n' % _u(repo.root))
+   ui.write(b'<?xml version="1.0"?>\n')
+   ui.write(b'<rhsummary>\n')
+   ui.write(b'<repository root="%s">\n' % _u(repo.root))
    try:
        _tip(ui, repo)
        _tags(ui, repo)
@@ -234,6 +247,6 @@
        _branches(ui, repo)
        # TODO: bookmarks in core (Mercurial>=1.8)
    finally:
-        ui.write('</repository>\n')
-        ui.write('</rhsummary>\n')
+        ui.write(b'</repository>\n')
+        ui.write(b'</rhsummary>\n')

```

#2 - 2020-08-08 23:42 - S DW

Mercurial integration with redmine on the NixOS linux distribution is broken, and I think it could be related to this issue?
The error we are getting is either:

```

Aug 08 18:07:33 nixos bundle[871]: *** failed to import extension redminehelper from /nix/store/c2za42x86iqk8w
rxphl4pc2rkb5kbwxq-redmine-4.1.1/share/redmine/lib/redmine/scm/adapters/mercurial/redminehelper.py: unicode 'r
hsummary' found in cmdtable
Aug 08 18:07:33 nixos bundle[871]: *** (use b'' to make it byte string)
Aug 08 18:07:33 nixos bundle[871]: hg: unknown command 'rhsummary'
Aug 08 18:07:33 nixos bundle[871]: (did you mean summary?)

```

Or, depending on redmine/mercurial versions used:

```

Aug 08 18:30:38 nixos bundle[1133]: *** failed to import extension redminehelper from /nix/store/3sdqxliq5qd15
wfyic1qs5ax1b6ggnc-redmine-4.1.1/share/redmine/lib/redmine/scm/adapters/mercurial/redminehelper.py: module 'c
gi' has no attribute 'escape'
Aug 08 18:30:38 nixos bundle[1133]: hg: unknown command 'rhsummary'
Aug 08 18:30:38 nixos bundle[1133]: (did you mean summary?)

```

Link to the NixOS issue: <https://github.com/NixOS/nixpkgs/issues/94949>

#3 - 2022-02-07 21:48 - François Cerbelle

Mercurial switched to python 3, python 2.x is EOL/abandonned.
Debian switched to Rail 6.x in all versions. Thus, it is impossible to use Redmine later than 4.0.7.
It became a real challenge to keep a Redmine system uptodate with dependency versions and security.

Any chance to have this bug/script fixed in the 4.0.x branch ? at least included in a roadmapped version ?
The provided patch merged ? It is better to have something partially working than no mercurial at all.

Not complaining, just asking

#4 - 2022-02-08 08:36 - Harald Klimach

François Cerbelle wrote:

Any chance to have this bug/script fixed in the 4.0.x branch ? at least included in a roadmapped version ?
The provided patch merged ? It is better to have something partially working than no mercurial at all.

Can't say anything about roadmaps or integration into the project, but did you try to apply the patch above? Though I didn't test it, I think it should work for that branch aswell.

#5 - 2022-02-17 10:07 - salman mp

+1

same problem

#6 - 2022-03-22 07:40 - Go MAEDA

- Target version set to Candidate for next major release

#7 - 2022-08-09 22:35 - Scott Cunningham

I manually added the patch above ([thank you Harald Klimach](#)) with Redmine 5.0.2 and Mercurial 6.2.1. I installed the patch and it seems to be working (can view history, files, diffs, branches) based on some quick testing.

My understanding is that Mercurial 6.2.1 not only dropped Python 2 but also support up to Python 3.5 - it is Python 3.6+ now (<https://www.mercurial-scm.org/wiki/SupportedPythonVersions>).

- Windows server 2019
- Bitnami Redmine app for windows:

```
Redmine version      5.0.2.stable
Ruby version         2.6.10-p210 (2022-04-12) [x64-mingw32]
Rails version        6.1.6
Environment          production
Database adapter      Mysql2
Mailer queue         ActiveJob::QueueAdapters::AsyncAdapter
Mailer delivery       smtp
SCM:
Mercurial             6.2.1
```

#8 - 2022-09-09 09:14 - Michael Sssssno

Hello,

I have the same problem, but unfortunately applying the patch does not work for me.

As described here <https://www.redmine.org/projects/redmine/wiki/patch> I go into the redmine directory and execute the patch with "patch -p0 < PATCH.diff or "sudo patch -p0 < PATCH.diff".

Unfortunately, I get the following error messages

patching file lib/redmine/scm/adapters/mercurial/redminehelper.py

Hunk #1 FAILED at 45.

Hunk #3 FAILED at 84.

Hunk #4 FAILED at 104.

3 out of 5 hunks FAILED -- saving rejects to file lib/redmine/scm/adapters/mercurial/redminehelper.py.rej

On my system (ubuntu 22.04 server) I installed redmine 5.0.2 with mercurial 6.1.1.

Redmine according to this manual https://www.redmine.org/projects/redmine/wiki/HowTo_Install_Redmine_50x_on_Ubuntu_2004_with_Apache2 (but with "sudo svn co <https://svn.redmine.org/redmine/branches/5.0-stable> redmine-5.0") and mercurial with " sudo apt install mercurial".

The system runs so far, only the mercurial connection does not work.

What am I doing wrong?

#9 - 2022-11-09 12:43 - François Cerbelle

Harald Klimach wrote:

François Cerbelle wrote:

Any chance to have this bug/script fixed in the 4.0.x branch ? at least included in a roadmapped version ?
The provided patch merged ? It is better to have something partially working than no mercurial at all.

Can't say anything about roadmaps or integration into the project, but did you try to apply the patch above? Though I didn't test it, I think it should work for that branch aswell.

I did, but it does not apply :

```
/usr/share/redmine # patch --verbose -F3 -p0 --dry-run <hgpatch.patch
Hmm... Looks like a unified diff to me...
The text leading up to this was:
-----
|Index: lib/redmine/scm/adapters/mercurial/redminehelper.py
|=====
|--- lib/redmine/scm/adapters/mercurial/redminehelper.py      (revision 19937)
|+++ lib/redmine/scm/adapters/mercurial/redminehelper.py      (working copy)
```

```
-----
checking file lib/redmine/scm/adapters/mercurial/redminehelper.py
Using Plan A...
Hunk #1 succeeded at 45 with fuzz 3.
Hunk #2 succeeded at 73.
Hunk #3 succeeded at 87.
Hunk #4 FAILED at 107.
Hunk #5 succeeded at 237.
1 out of 5 hunks FAILED
done
```

#10 - 2022-12-14 09:28 - Olivier Houdas

- File `33784_Redmine_5.0.3.patch` added

Here are my 2 cents to this issue:

I have attached a patch for Redmine 5.0.3 for those interested.

HOWEVER, in this patch, I have modified line 129 from

```
fstr = f.decode('utf-8')
```

to

```
fstr = f.decode('cp1252')
```

as we had got an error

```
redmine_1 | File "/usr/src/redmine/lib/redmine/scm/adapters/mercurial/redminehelper.py", line 129, in _manifest
redmine_1 |     fstr = f.decode('utf-8')
redmine_1 | UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe9 in position 37: invalid continuation byte
```

So it seems that we need to use Mercurial's codepage for this line. Unfortunately, I don't see how to make this transparent for the user, it would depend on Mercurial's installation.

#11 - 2023-02-23 12:59 - Jakob Haufe

```
redmine_1 | File "/usr/src/redmine/lib/redmine/scm/adapters/mercurial/redminehelper.py", line 129, in _manifest
redmine_1 |     fstr = f.decode('utf-8')
redmine_1 | UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe9 in position 37: invalid continuation byte
```

So it seems that we need to use Mercurial's codepage for this line. Unfortunately, I don't see how to make this transparent for the user, it would depend on Mercurial's installation.

While trying to reproduce this problem I found out that this is not "Mercurials codepage". It's the encoding of filenames stored in the repository in question. Redmine already has a per-repo setting for this, it needs to be passed to the helper.

#12 - 2023-02-24 14:03 - Jakob Haufe

- File `mercurial-py3-fix` added

I've enhanced the patch to pass the configured file name encoding to the helper.

Olivier: Could you please test whether this works correctly in your installation?

#13 - 2023-06-04 14:05 - salman mp

Jakob Haufe wrote in [#note-12](#):

I've enhanced the patch to pass the configured file name encoding to the helper.

Olivier: Could you please test whether this works correctly in your installation?

I tested your patch and works smoothly. Thank you so much. But this below change must be applied to controller patch.

```
< +         full_args << '--config' << "redminehelper.fileencoding=" << @path_encoding
> +         full_args << '--config' << "redminehelper.fileencoding=#{@path_encoding}"
```

Env:

OS: Debian 11
Redmine: 4.2.10

Mercurial: 5.6.1
Python: 3.9

#14 - 2023-06-04 16:03 - Jakob Haufe

Thanks [salman mp](#), you are completely right. Odd that this didn't break in my tests. Will update the patch accordingly. I originally planned to have this included in the redmine package in Debian bookworm as well but forgot to take care of it in time.

#15 - 2023-11-30 10:39 - Jakob Haufe

- File mercurial-py3-fix added

This is the patch as it's currently in Debian unstable.

Source: <https://salsa.debian.org/ruby-team/redmine/-/blob/master/debian/patches/mercurial-py3-fix>

#16 - 2024-01-31 07:53 - Go MAEDA

- Target version changed from Candidate for next major release to 6.0.0

Setting the target version to 6.0.0.

#17 - 2024-02-12 06:17 - Go MAEDA

- Target version changed from 6.0.0 to Candidate for next major release

Thank you for posting the patch. There are still some test failures need to be fixed.

Failure:

RepositoriesMercurialControllerTest#test_show_tag [test/functional/repositories_mercurial_controller_test.rb:278]:

Expected response to be a <2XX: success>, but was a <404: Not Found>

bin/rails test test/functional/repositories_mercurial_controller_test.rb:260

Failure:

RepositoriesMercurialControllerTest#test_annotate_latin_1_path [test/functional/repositories_mercurial_controller_test.rb:550]:

Expected response to be a <2XX: success>, but was a <404: Not Found>

bin/rails test test/functional/repositories_mercurial_controller_test.rb:539

Failure:

RepositoriesMercurialControllerTest#test_entry_show_latin_1_path [test/functional/repositories_mercurial_controller_test.rb:323]:

Expected response to be a <2XX: success>, but was a <404: Not Found>

bin/rails test test/functional/repositories_mercurial_controller_test.rb:312

Failure:

RepositoriesMercurialControllerTest#test_annotate_latin_1_contents [test/functional/repositories_mercurial_controller_test.rb:576]:

Expected at least 1 element matching "tr#L1 td.line-code", found 0.

Expected 0 to be >= 1.

bin/rails test test/functional/repositories_mercurial_controller_test.rb:564

Failure:

RepositoriesMercurialControllerTest#test_show_directory_latin_1_path [test/functional/repositories_mercurial_controller_test.rb:196]:

Expected response to be a <2XX: success>, but was a <404: Not Found>

bin/rails test test/functional/repositories_mercurial_controller_test.rb:181

Failure:

RepositoriesMercurialControllerTest#test_entry_show_latin_1_contents [test/functional/repositories_mercurial_controller_test.rb:340]:

Expected response to be a <2XX: success>, but was a <404: Not Found>

bin/rails test test/functional/repositories_mercurial_controller_test.rb:328

Failure:

RepositoriesMercurialControllerTest#test_show_branch [test/functional/repositories_mercurial_controller_test.rb:253]:

Expected response to be a <2XX: success>, but was a <404: Not Found>

```
bin/rails test test/functional/repositories_mercurial_controller_test.rb:232
```

Failure:

```
RepositoryMercurialTest#test_fetch_changesets_from_scratch [test/unit/repository_mercurial_test.rb:173]:
```

Expected: 53

Actual: 47

```
bin/rails test test/unit/repository_mercurial_test.rb:168
```

Failure:

```
RepositoryMercurialTest#test_latest_changesets [test/unit/repository_mercurial_test.rb:266]:
```

Expected: ["30", "11", "10", "9"]

Actual: ["11", "10", "9"]

```
bin/rails test test/unit/repository_mercurial_test.rb:251
```

#18 - 2025-02-05 02:57 - Sean Baggaley

- File mercurial-py3.patch added

I had a go at tackling this myself. I attached an updated version of the patch (mercurial-py3.patch) that fixes the charset conversation issues and therefore the related tests. Since some of the output from Mercurial is in UTF-8 now, some of the conversions from the path encoding were unnecessary. Additionally, in the previous patches the config parameter that Redmine passed and the helper read differed.

This does not fix the test_fetch_changesets_from_scratch or test_latest_changesets test failures, but those are a separate issue, unrelated to the Python 3 conversion or charset shenanigans. This is an intentional behaviour change in Mercurial; I bisected it to <https://repo.mercurial-scm.org/hg/rev/0c72eddb4be5> - starting with that commit, Mercurial now also considers the second parent of a merge commit when listing added files. Revision 30 in the hg test repository is a merge commit, and its second parent adds files, this means Mercurial versions starting with 5.1 will fail these tests, as the added files are no longer listed. I am not sure how to fix those tests while remaining compatible with older Mercurial versions if desired (I am very much a novice at this), but I hope this investigation helps to solve this.

#19 - 2025-02-13 11:04 - Go MAEDA

Sean Baggaley wrote in [#note-18](#):

I am not sure how to fix those tests while remaining compatible with older Mercurial versions if desired (I am very much a novice at this), but I hope this investigation helps to solve this.

I think we can drop support for Mercurial before to 5.1. Mercurial 5.1 was released in 2019, almost 6 years ago.

```
diff --git a/lib/redmine/scm/adapters/mercurial_adapter.rb b/lib/redmine/scm/adapters/mercurial_adapter.rb
index 921c9fcc3..28a1922ca 100644
--- a/lib/redmine/scm/adapters/mercurial_adapter.rb
+++ b/lib/redmine/scm/adapters/mercurial_adapter.rb
@@ -50,7 +50,7 @@ module Redmine
   end
```

```
    def client_available
-      client_version_above?([1, 2])
+      client_version_above?([5, 1])
    end
```

```
    def hgversion
```

#20 - 2025-02-14 01:44 - Sean Baggaley

- File mercurial_py3_and_5.1.patch added

That makes it easy then - attached another patch, with your suggestion, and fixing the remaining tests.

#21 - 2025-02-14 03:36 - Go MAEDA

- Target version changed from Candidate for next major release to 6.1.0

Sean Baggaley wrote in [#note-20](#):

That makes it easy then - attached another patch, with your suggestion, and fixing the remaining tests.

Thank you for updating the patch.

I believe the patch is now ready to be committed. I am setting the target version to 6.1.0.

#22 - 2025-02-16 09:45 - Go MAEDA

- Subject changed from *Updating Mercurial helper to work with Python3 to Update Mercurial repository support to be compatible with Python 3 and remove support for Mercurial < 5.1*
- Status changed from *New* to *Closed*
- Assignee set to *Go MAEDA*
- Resolution set to *Fixed*

Committed the patch in [r23513](#). Thank you for your contribution.

#23 - 2025-03-29 14:07 - Marius BĂLTEANU

- Status changed from *Closed* to *Reopened*

[Go MAEDA](#), can you take a look on the failing tests from Redmine CI? https://www.redmine.org/builds/logs/build_trunk_mysql_ruby-3.3_1431.html

#24 - 2025-03-30 09:51 - Go MAEDA

Marius BĂLTEANU wrote in [#note-23](#):

[Go MAEDA](#), can you take a look on the failing tests from Redmine CI?
https://www.redmine.org/builds/logs/build_trunk_mysql_ruby-3.3_1431.html

Maybe the version of the `hg` command on the CI server is old The current mercurial adapter.

The current Mercurial adapter requires Mercurial 5.1 or later. However, the tests are executed without verifying the result of `Redmine::Scm::Adapters::MercurialAdapter.client_available` method. As a result, they fail when an older version of Mercurial is installed.

I will update the tests to skip execution when a appropriate client is not available.

#25 - 2025-03-30 09:53 - Marius BĂLTEANU

Go MAEDA wrote in [#note-24](#):

Marius BĂLTEANU wrote in [#note-23](#):

[Go MAEDA](#), can you take a look on the failing tests from Redmine CI?
https://www.redmine.org/builds/logs/build_trunk_mysql_ruby-3.3_1431.html

Maybe the version of the `hg` command on the CI server is old The current mercurial adapter.

The current Mercurial adapter requires Mercurial 5.1 or later. However, the tests are executed without verifying the result of `Redmine::Scm::Adapters::MercurialAdapter.client_available` method. As a result, they fail when an older version of Mercurial is installed.

I will update the tests to skip execution when a appropriate client is not available.

Good idea, I will contact Jean-Philippe in the meantime, maybe he can help us updating the Mercurial client.

#26 - 2025-04-01 11:51 - Go MAEDA

Go MAEDA wrote in [#note-24](#):

I will update the tests to skip execution when a appropriate client is not available.

I have submitted a patch in [#42500](#).

#27 - 2025-04-20 06:33 - Go MAEDA

Marius BĂLTEANU wrote in [#note-23](#):

[Go MAEDA](#), can you take a look on the failing tests from Redmine CI?
https://www.redmine.org/builds/logs/build_trunk_mysql_ruby-3.3_1431.html

The problem was fixed in trunk by [r23687](#). The reason for the failure was that the code in the current trunk requires Python 3 but the CI server was using Python 2.7. I updated the MercurialAdapter#client_available method to check whether Mercurial uses Python 3.5 or higher in [r23687](#).

The failures in the 6.0-stable and 5.1-stable branches are caused by a different issue: a behavior change in Mercurial 5.1 described in [#note-18](#). To address this, I have just posted a patch [#42597](#) that skips affected tests. Since those tests are properly run with Mercurial 5.1 and later in the trunk, I

don't think it is necessary to spend extra effort making them work with both older and newer Mercurial versions in the 6.0-stable and 5.1-stable branches.

#28 - 2025-04-20 07:09 - Go MAEDA

- Related to Patch #42500: Skip repository tests when the SCM client command is unavailable added

#29 - 2025-04-20 07:10 - Go MAEDA

- Related to Patch #42597: Skip some Mercurial tests when using Mercurial 5.1 or later in Redmine 6.0 or 5.1 added

#30 - 2025-04-20 07:14 - Go MAEDA

- Related to Defect #33108: Mercurial support is broken for Mercurial 5.1 or later added

#31 - 2025-04-20 09:40 - Marius BĂLTEANU

Thanks for finding the second root cause and for the fixes, we started to have green results [on the CI](#). Do you want to ask Jean-Philippe to update Python to at least 3.5 on the CI server?

#32 - 2025-04-20 11:27 - Jakob Haufe

Please update to a supported version of python, i.e. at least 3.9 which will be EOL in October 2025. 3.5 is EOL for more than 4 years now.

#33 - 2025-04-22 11:17 - Go MAEDA

- Status changed from Reopened to Closed

Marius BĂLTEANU wrote in [#note-31](#):

Do you want to ask Jean-Philippe to update Python to at least 3.5 on the CI server?

I sent an email to Jean-Philippe Lang asking him to update Python to version 3.x.

Files

33784_Redmine_5.0.3.patch	11.5 KB	2022-12-14	Olivier Houdas
mercurial-py3-fix	12.6 KB	2023-02-24	Jakob Haufe
mercurial-py3-fix	12.6 KB	2023-11-30	Jakob Haufe
mercurial-py3.patch	13.1 KB	2025-02-05	Sean Baggaley
mercurial_py3_and_5.1.patch	15.3 KB	2025-02-14	Sean Baggaley