



**GHENT
UNIVERSITY**



Plone

UGent University website migration to Plone 5.2
Plone Conference 2019 Ferrara

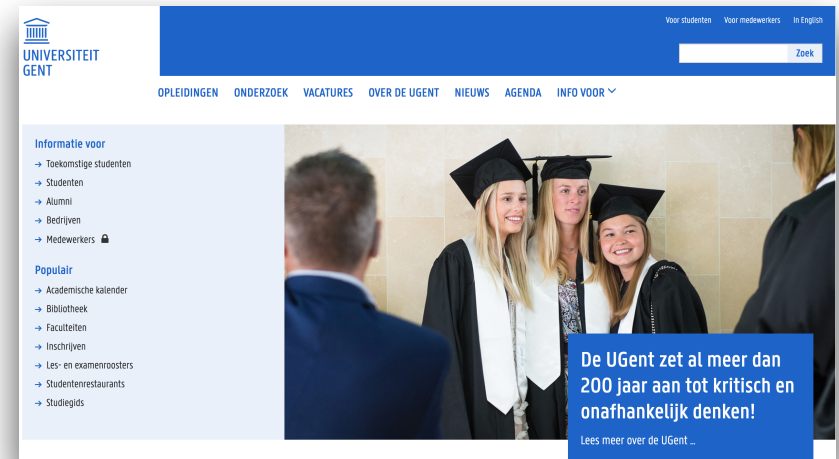
Andreas Jung
www.zopyx.com • info@zopyx.com

About Me

- **Python** since 1993
- **Plone** since 2002
- **Publishing** since 1995
- Plone migrations this year:
 -  2 larger custom Plone migrations to Plone 5.2, Python 3
 -  7 smaller/medium migrations to Plone 5.1
 -  4 other migrations to Plone 5.2, Python 3 planned
- contractor in current UGent Plone 5.2 migration project

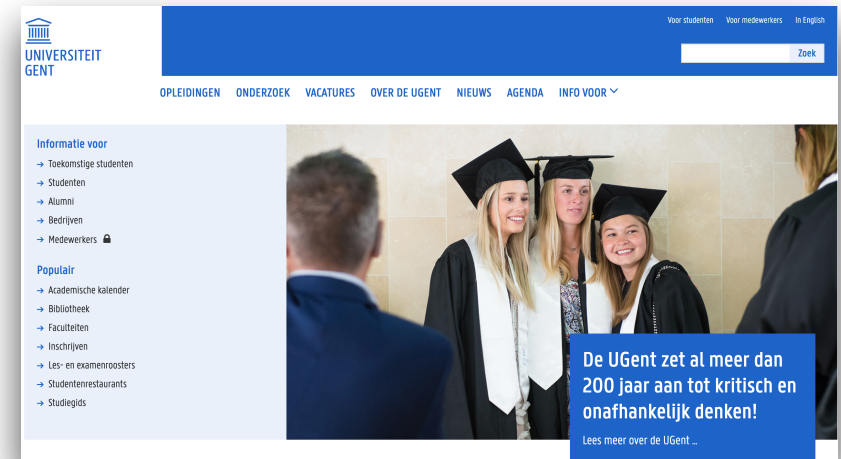
Plone @ugent.be

- Ghent University, Belgium
- 44.000 students
- 9.000 staff members
- 500 active editors
- one large Plone site for all departments
- current state: Plone 4.3
- about 83 add-ons
- 69 subsites
- Size: about 90.000 content objects
- Languages: English, Dutch, (Korean) (no translation tool)
- Plone migration team: 6 persons (3 internal, 3 external)



Plone @ugent.be

- 2002: **Zope/CMF**
- 2007: **Plone 2.5**
- 2008: new theme
- 2010: **Plone 3.3**
- 2012: **Plone 4.1**
- 2015: **Plone 4.3**
- 2016: new theme
- 2019/2020: **Plone 5.2/Python 3**



Plone migration approaches

- traditional Plone **in-place** migration
 - not applicable
 - too many old add-ons, no 1:1 mapping possible
- Full export, full import using **Transmogrifier**
 - not ready for Python 3 (6 months ago)
 - too much magic hidden in too many places (blueprints)
- Full export, full import using a **custom migration solution**
 - `collective.jsonify` → ArangoDB → `plone.restapi`

Add-ons @ugent.be

archetypes.markerfield
archetypes.schematuning
atreal.massloader
atreal.monkeyplone
collective.blog.portlets
collective.blog.view
collective.contentstats
collective.indexing
collective.keywordwidgetreplacer
collective.portlet.localsearch
collective.revisionmanager
collective.searchform
collective.setdefaulteditor
collective.tinymce templates
collective.topicitemsevent
collective.ZipFileTransport
p4a.plonecalendar
plone.app.changeownership
Products.CAS4PAS
Products.membrane
Products.OpenXml
Products.PloneFormGen

Products.RedirectionTool
munin.zope
Products.ZNagios
collective.portlet.localsearch
pyugent
ugent.api
ugent.authorship
ugent.cas_config
ugent.clearhistory
ugent.collectiondelete
ugent.configuration
ugent.contentfixer
ugent.customsearch
ugent.defaultpagechanged
ugent.editmode
ugent.folderdeleteconfirmation
ugent.foldersorting
ugent.forms
ugent.frameworkcontracts
ugent.helper
ugent.homepages
ugent.imagelibrary

ugent.librarywidgets
ugent ldap
ugent.linkchecker
ugent.membrane
ugent.metadata
ugent.pas ldap
ugent.patches
ugent.phdddefense
ugent.phonebook
ugent.plone4theme
ugent.plone4translations
ugent.policy
ugent.portlet.banner
ugent.portlet.collection
ugent.portlet.infolinks
ugent.portlet.links
ugent.portlet.slideshow
ugent.portlet.socialbanner
ugent.portlet.twitter
ugent.portlets
ugent.provisionalcontent
ugent.remove markers

ugent.reporting
ugent.review
ugent.rss
ugent.sharing
ugent.stalecontent
ugent.subsites
ugent.supergroup
ugent.textindex
ugent.tinymce
ugent.unavailable
ugent.vacancy
ugent.vocabularies
mailinglogger
collective.z3c.keywordwidget
qi.portlet.TagClouds
eea.facetednavigation
raven

Content types @ugent.be

- standard Plone types
- custom content-types
 - PloneFormGen
 - PhdDefense
 - Vacancy
 - LibraryDocument
- extensive usage of schema extenders

Analyze and investigate your dependencies

- check your packages for Python 3 compatibility (setup.py)
- check for Python 3/Plone 5.2 related branches
- drop any package related to Archetypes or other Python 3 incompatible dependencies
- drop packages are not no longer needed or are obsolete
- check for package alternatives
(PloneFormGen → collective.easyform)

Document your insights

Third-party packages inventory

Created by Dimitri Boone, last modified on Oct 10, 2019

	Package	Description	Source code	Is dependency of:	Plone 5.2 compatible?	Python 3 compatible?	Notes	Must have
1	Products.CAS4PAS	PAS plugin that authenticates users against a CAS-server.	https://github.com/collective/Products.CAS4PAS	ugent.cas_config	?		Possibly easy to port and to fix since the code only depends on PAS which has not changed much since a decade. And the code base is not that big. But no experience with this specific module nor with CAS itself.	Y
2	collective.castle	Plone user interface for Products.CAS4PAS. CAS4PAS enables CAS-based authentication.	https://github.com/collective/collective.castle	- ugent.cas_config - ugent.editmode	?		Same as with Products.CAS4PAS ☒ PCM-1167 - collective.castle niet compatibel met Plone 5 RESOLVED	Y
3	Products.membrane dexterity.membrane	Content-based users and groups.	https://github.com/collective/Products.membrane https://github.com/collective/dexterity.membrane	ugent.membrane	?		under investigation: https://community.plone.org/t/state-and-plans-of-membrane-on-plone-5-2-python-3/8282	Y
4	borg.supergroup	PAS plugin to create groups "on the fly" for users via subscription adapters and utilities.		ugent.supergroup	N	N	old-style imports, not Python 3 compatible but possibly easy to port	Y
5	munin.zope	Munin plugins for Zope/Plone.	https://github.com/RedTurtle/munin.zope		N	N	Depends on Zope2's ZServer which is gone in Zope 4 under Python 3	
6	Products.ZNagios	Allows Nagios and munin to tap into the Zope2 server and retrieve status and performance data.	https://github.com/zopefoundation/Products.ZNagios		N	N	Like the same as with munin.zope	
7	raven Replaced with collective.sentry/zope2instance extension	Client for Sentry (error tracker)	https://github.com/getsentry/raven-python		?		<i>Raven is the legacy client for Sentry, replaced by sentry-python ?</i> <i>Low prio, only relevant for production.</i> <i>Replaced with collective.sentry and a pending PR for zope2instance recipe</i>	Y
8	RedirectionTool						<i>Will be moved in Plone core as of 5.2</i>	Y
9	collective.easyform	Plone form builder (replacement for Products.PloneFormGen)	https://github.com/collective/collective.easyform		Y	Y	collective.easyform 2.1 compatible with 5.2/Python3	Y
10	eea.facetednavigation	Enables faceted navigation on collections.	https://github.com/eea/eea.facetednavigation		Y	Y	compatible with Python 3/Plone 5.2 in version >= 13	Y
11	collective.siteimprove	Siteimprove.com integration	https://github.com/collective/collective.siteimprove		Y	Y	Ported to Python 3, Plone 5.2 ("plone5" branch). Controlpanel working, overlay showing up, no further tests possible without Siteimprove credentials.	Y

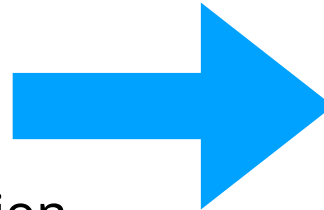
Create your target setup

- start with a minimal Plone 5.2 buildout
- add one verified Python 3 compatible add-on at a time
- test, test, test `_manually_`
- focus on content-types **first**
- configurations, decorations, extra functionality **next**

@ugent.be migration export

- customized version of **collective.jsonify**:

- full-export of all content objects and their properties
- export of portlet assignments
- default pages, layout information
- workflow states
- local roles (@@sharing)
- pre-computed values for further efficient processing



- 90.000 content objects
- 90.000 JSON files
- 55 GB data
- 90 minutes
- binaries b64 encoded

ArangoDB migration DB

Why a migration database?

- we want to run partial imports (e.g. importing a particular subtree of the original site)
- we want to test a particular migration feature e.g. a complex migration step (PloneFormGen to EasyForm)
- Easy query and check the original data (just a query by path)

Why ArangoDB?

- 🐼 because MongoDB sucks (16 MB BSON limit) 🐼
- Multi-model database: key-value, document, graph
- dedicated DSL AQL (Arango Query Language) vs. JSON-based queries (ES, MongoDB...)
- easy to install and use
- JSON files can be dumped into ArangoDB without modifications
- import time: about 45 minutes
- successfully used in previous Python projects



ArangoDB migration DB

The screenshot displays the ArangoDB web interface. At the top, there are buttons for 'Queries', 'New', and 'Save as'. A dropdown menu shows '1000 results'. The query editor contains the following code:

```
1 for doc in import
2   filter '/plone_portal/nl' in doc._paths_all && doc._type == 'Event'
3   return {'path': doc._path, 'type': doc._type}
```

Below the query editor, there are buttons for 'Remove all results', 'Create Debug Package', 'Profile', 'Explain', and 'Execute'. The results panel shows '133 elements' and a execution time of '29.851 s'. The results are displayed in JSON format, showing a list of documents with 'path' and 'type' fields.

Key Value JSON

No bind parameters defined.

Query 133 elements 29.851 s JSON Table

```
1 {
2   {
3     "path": "/plone_portal/nl/agenda/1567755861528",
4     "type": "Event"
5   },
6   {
7     "path": "/plone_portal/nl/agenda/1565352862012",
8     "type": "Event"
9   },
10  {
11    "path": "/plone_portal/nl/agenda/1567079431754",
12    "type": "Event"
13  },
14  {
15    "path": "/plone_portal/nl/agenda/1567497735079",
```

Import infrastructure

- clean Python 3.7, Python 5.2 buildout
- import via plone.restapi
- dedicated **migration package** with specific views for functionality that is not provided by plone.restapi or specific to the migration project
 - path to UID in RichText fields
 - setting INavigationRoot
 - setting position in parent
 - allowed/addable types per folder (folder restrictions)
 - setting related items (as post migration operation)
 - setting permissions and roles
 - setting marker interfaces
 - adding portlets

The "magic" migration script (1/3)

- YAML based configuration

```
1  arango:
2    url: http://localhost:8529
3    username: root
4    password: ugent
5    database: ugent
6    collection: import
7
8  plone:
9    # remote Plone site running plone.restapi
10   url: http://localhost:20081
11
12   # HTTP auth credentials
13   username: admin
14   password: admin
15
16  site:
17    # id of new site
18    id: plone_portal
19
20    # add-on profiles to be imported (':default' is implied unless specified)
21    extension_ids:
22      - ugent.policy
23
24    languages:
25      - nl
26      - en
27
28    default language: nl
```

```
30  migration: Andreas JungA, 4 months ago • moved migration configuration
31
32   # content types explicitly excluded from the migration (should not conflict
33   # with 'content_types'
34   excluded_content_types:
35
36   # import only the given content types. No content_types specified -> full import,
37   # no filtering
38   content_types:
39     - Topic
40     - Folder
41     - Link
42     - News Item
43     - Event
44     - Document
45     - FormFolder
46     - PhdDefense
47     - Vacancy
48     - LibraryDocument
49     - Topic
50     - Image
51     - File
52
53   # folders to be imported by absolute path (not relative path to the Plone site)
54   folders:
55
56     - /plone_portal/200
57     - /plone_portal/alumnus
58     - /plone_portal/anser
59     - /plone_portal/aseanplus
60     - /plone_portal/bw
61     - /plone_portal/campus-kortrijk
62     - /plone_portal/cesam
63     - /plone_portal/cessmir
64     - /plone_portal/chinaplatform
```

The "magic" migration script (2/3)

- **Phase 1 (migration setup & preparation)**

- migration pre-check
- removal of target site (if existing)
- creating a new fresh Plone 5.2 site with a given number of extension profiles

- **Phase 2 (folders)**

- query ArangoDB for all folderish types
- rebuild complete folder structure with full metadata, workflow information

The "magic" migration script (3/3)

- **Phase 3 (non folders)**

- query ArangoDB for all non-folderish types
- add non-folderish content as leafs to existing folder structure

- **Phase 4 (global actions)**

- path to UID check/migration for RichText fields
- assignment of portlets
- some specific fixup operations

PloneFormGen to EasyForm

- **PloneFormGen** no longer available under Plone 5.2/Python 3 (Archetypes dependency)
- **collective.easyform** as replacement (Dexterity-based clone of PFG)
- PFG export: one JSON for FormFolder and 1..N JSON for all fields and mail, save and script adapter
- fields and actions are defined in **collective.easyform** as a schema
- extract relevant metadata for **EasyForm** from all JSON files
- build supermodel XML files for fields and actions from all JSON files
- instantiate **EasyForm** instance and assign fields + actions supermodel

Topics to Collections

- Topics export: one JSON for 'Topic' and 1..N JSON file for each criteria
- Conversion of old-style topic criteria to `plone.app.querystring`
- basically based on original migration code of Plone
- instantiate **Collection** and assign query strings

From schema extenders to behaviors

1. documentation and analysis of all schema extenders, their usage
2. check with Dexterity replacements
3. reimplementation as behaviors

Issues

- customizing of existing behaviors painful
- bunch of monkey-patches or code duplication

Migrating packages to Python 3

- mostly covered by talks of Philipp Bauer and David Glick
- most common problems:
 - fixing UTF-8 vs. unicode related code
 - import fixes
 - @implements -> @implementer
- rarely used 2to3 or moderinzr
- no need to maintain both Python 2 and Python 3 compatibility
- usually created a dedicated *Plone52Python3* branch from the current master
- unit tests if available
- basic manually testing

Reimplementations

- historic code in portal_skins replaced with browser views → reimplementation
- core Archetypes-based content-types replaced with Dexterity → reimplementation

Other common problems

- improper image/file metadata in blobs
(TIFF stored as some.jpg, image/jpeg)
- migration on vocabulary values
(e.g. old to new departments)
- there is always one more thing missing in your export
(repetitive export/import cycles)
- there is always one more bug in your migration script
(portlets, collections)

Quality control

- aspects
 - completeness of content and configurations
 - consistency of migration (systematic errors)
 - too many tiny aspects slipping out of our focus
- unit tests if available
- **manual testing** by different persons
- "works for me" does not mean that it works for others

Add-ons @ugent.be (old)

archetypes.markerfield
archetypes.schematuning
atreal.massloader
atreal.monkeyplone
collective.blog.portlets
collective.blog.view
collective.contentstats
collective.indexing
collective.keywordwidgetreplacer
collective.portlet.localsearch
collective.revisionmanager
collective.searchform
collective.setdefaulteditor
collective.tinymce templates
collective.topicitemsevent
collective.ZipFileTransport
p4a.plonecalendar
plone.app.changeownership
Products.CAS4PAS
Products.membrane
Products.OpenXml
Products.PloneFormGen

Products.RedirectionTool
munin.zope
Products.ZNagios
collective.portlet.localsearch
pyugent
ugent.api
ugent.authorship
ugent.cas_config
ugent.clearhistory
ugent.collectiondelete
ugent.configuration
ugent.contentfixer
ugent.customsearch
ugent.defaultpagechanged
ugent.editmode
ugent.folderdeleteconfirmation
ugent.foldersorting
ugent.forms
ugent.frameworkcontracts
ugent.helper
ugent.homepages
ugent.imagelibrary

ugent.librarywidgets
ugent ldap
ugent.linkchecker
ugent.membrane
ugent.metadata
ugent.pas ldap
ugent.patches
ugent.phdddefense
ugent.phonebook
ugent.plone4theme
ugent.plone4translations
ugent.policy
ugent.portlet.banner
ugent.portlet.collection
ugent.portlet.infolinks
ugent.portlet.links
ugent.portlet.slideshow
ugent.portlet.socialbanner
ugent.portlet.twitter
ugent.portlets
ugent.provisionalcontent
ugent.remove markers

ugent.reporting
ugent.review
ugent.rss
ugent.sharing
ugent.stalecontent
ugent.subsites
ugent.supergroup
ugent.textindex
ugent.tinymce
ugent.unavailable
ugent.vacancy
ugent.vocabularies
mailinglogger
collective.z3c.keywordwidget
qi.portlet.TagClouds
eea.facetednavigation
raven

Add-ons @ugent.be (new)

Products.PloneKeywordManager

eea.facetednavigation

collective.siteimprove

collective.blog.portlets

collective.portlet.localsearch

ugent.customaddforms

ugent.forms

ugent.imagelibrary

ugent.librarywidgets

ugent.phddefense

ugent.phonebook

ugent.plone5migration

ugent.policy

ugent.portlet.collection

ugent.portlet.infolinks

ugent.portlet.links

ugent.portlet.slideshow

ugent.portlet.socialbanner

ugent.portlet.twitter

ugent.showlabel

ugent.subsitefolder

ugent.textindex

ugent.vacancy

plone.app.changeownership

Current status/open issues

- Content migration almost complete
- next: detailed testing needed
- next: integration with updated UGent theme and theme testing
- open:
 - replacement of a specific Membrane usage
 - collective.castle, Products.CAS4PAS

Takeaways

- Export Plone → JSON: **2 hours** (fast)
- Import JSON → ArangoDB: **45 mins** (fast)
- Import ArangoDB → plone.restapi: **36-48 hours** (painfully slow)
- 1,5-2,0 seconds per content object in average
- no option for parallelization (risk of write conflicts)
- **CREATE** most expensive operation
- Plone/Zope/ZODB are painfully slow backends for "mass" content

Questions?