Vecto 3.x Release Notes

DI Dr. Markus Quaritsch, DI Dr. Christian Kreiner, Michael Krisper

November 12, 2015

1 Vecto 3.0.1 Release Notes

1.1 General Notes

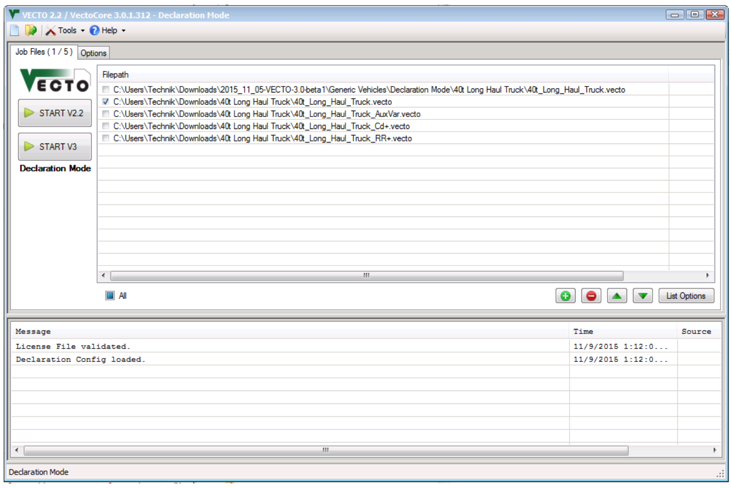
Vecto 3.0 has been rewritten from scratch. It is now a dynamic program library that contains the core simulation and can be embedded in other applications.

Vecto 3.0.1 has been integrated in the graphical user interface of Vecto 2.2 via an additional "Start" button on the main screen (cf. Figure 1). Additionally, a command-line program is provided to run multiple Vecto jobs (cf. Figure 2). For more information how to use the command-line program please see "vectocmd.exe - h".

In case you find a bug or Vecto 3.0 does not behave as expected **please follow the instructions** given in Section 2.

1.2 Changes in Vecto 3.0.1

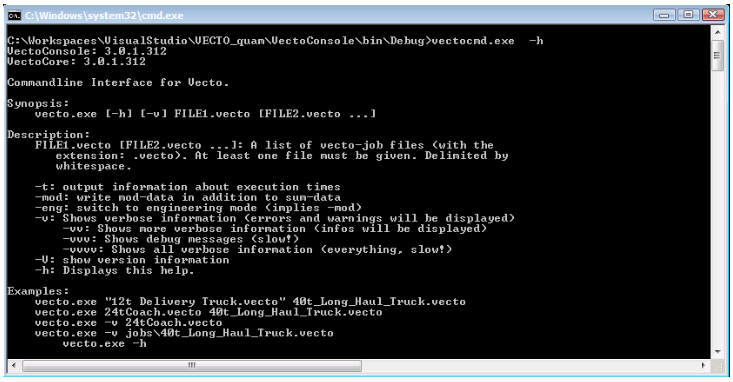
- New distance-based simulation model
- Simulation with variable steps. The simulation distance is adapted such that the respective time is about 0.5 s (parameter TargetTimeInterval)
- Accuracy of simulated distance vs. driving cycle distance $< 1\,\mu\text{m}$
- New component-based software architecture



Filepath
□ C:\Users\Technik\Downloads\2015_11_05-VECTO-3.0-beta1\Generic Vehicles\Declaration Mode\40t Long Haul Truck\40t_Long_Haul_Truck.vecto
☑ C:\Users\Technik\Downloads\40t Long Haul Truck\40t_Long_Haul_Truck.vecto
□ C:\Users\Technik\Downloads\40t Long Haul Truck\40t_Long_Haul_Truck_AuxVar.vecto
□ C:\Users\Technik\Downloads\40t Long Haul Truck\40t_Long_Haul_Truck_Cd+.vecto
□ C:\Users\Technik\Downloads\40t Long Haul Truck\40t_Long_Haul_Truck_RR+.vecto

Message	Time	Source
License File validated.	11/9/2015 1:12:0...	
Declaration Config loaded.	11/9/2015 1:12:0...	

Figure 1: Vecto 3.0.1 Graphical User Interface



```
C:\Workspaces\VisualStudio\VECTO_quam\VectoConsole\bin\Debug>vectocmd.exe  -h
VectoConsole: 3.0.1.312
VectoCore: 3.0.1.312

Commandline Interface for Vecto.

Synopsis:
    vecto.exe [-h] [-v] FILE1.vecto [FILE2.vecto ...]

Description:
    FILE1.vecto [FILE2.vecto ...]: A list of vecto-job files (with the
       extension: .vecto). At least one file must be given. Delimited by
       whitespace.

    -t: output information about execution times
    -mod: write mod-data in addition to sum-data
    -eng: switch to engineering mode (implies -mod)
    -v: Shows verbose information (errors and warnings will be displayed)
        -vv: Shows more verbose information (infos will be displayed)
        -vvv: Shows debug messages (slow!)
        -vvvv: Shows all verbose information (everything, slow!)
    -V: show version information
    -h: Displays this help.

Examples:
    vecto.exe "12t Delivery Truck.vecto" 40t_Long_Haul_Truck.vecto
    vecto.exe 24tCoach.vecto 40t_Long_Haul_Truck.vecto
    vecto.exe -v 24tCoach.vecto
    vecto.exe -v jobs\40t_Long_Haul_Truck.vecto
        vecto.exe -h
```

Figure 2: Vecto 3.0.1 Command Line Program

  - Separate software component for every part of the power-train: Combustion Engine, Clutch, Gearbox, Retarder, Axle Gear Vehicle, Wheels, Driver, Driving Cycle

    The models of every component are taken from Vecto 2.2
  - Interfaces for components reflecting physical quantities: force/velocity, angular speed/torque, . . .
  - Power-train components modularly usable, custom power-train configurations possible

- DriverStrategy as separate component with well-defined interface

- GearshiftStrategy as separate component with well-defined interface

- All computations are done in SI units; data-types reflect concrete SI units (i.e., kg, m, s, Nm, . . .)

- Run-time conformity check of SI units on all computations and SI conversions

- Traction interruption of arbitrary time possible (also $< 1\,\mathrm{s}$), accurate simulation of traction interruption time

- New Idle-Controller to simulate engine speed during traction interruption periods

- Intelligent search of operating point during coasting, braking, and in engine-overload situations, deviation from FLD $< 0.5\,\mathrm{W}$ (parameter EnginePowerSearchTolerance).

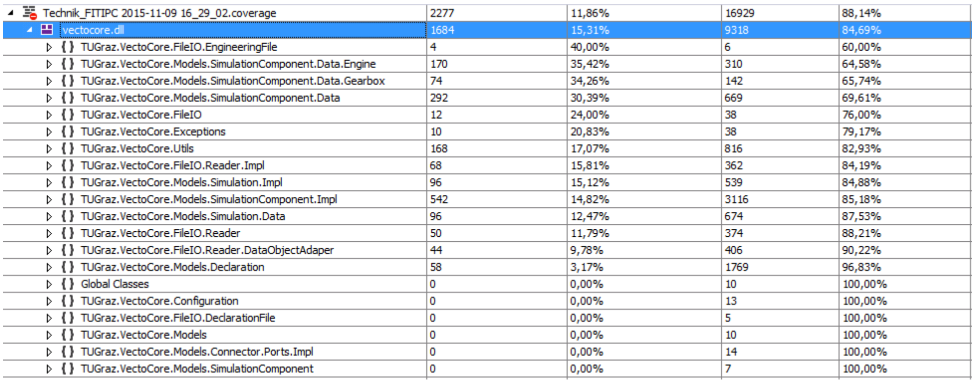
- Exact analytic solutions instead of exhaustive search where possible, e.g., computing engine's preferred speed, computing time interval required for driving a certain distance, compute distance required to decelerate to a given target speed depending on the driver's deceleration curve, computation of distance to drive when accelerating before braking, . . .

- Support sparse representation of driving cycles. Instead of specifying the driving cycle on a per-meter basis only distance entries with speed or slope changes need to be specified. Reduction of 96 % in disk space

- Multi-threaded execution when multiple driving cycles resp. multiple Vecto jobs are executed

- Vecto 3.0.1 only supports the latest version of the input file formats (JSON), i.e., .vecto: v 2; .vveh: v 7; .vgbx: v 5; .veng: v 3

- Support for arbitrary order of columns in CSV files if the header is given as specified in the Manual. If the header is not recognized: fall-back to column-based parsing



Technik_FITIPC 2015-11-09 16_29_02.coverage	2277	11,86%	16929	88,14%
vectocore.dll	1684	15,31%	9318	84,69%
{ } TUGraz.VectoCore.FileIO.EngineeringFile	4	40,00%	6	60,00%
{ } TUGraz.VectoCore.Models.SimulationComponent.Data.Engine	170	35,42%	310	64,58%
{ } TUGraz.VectoCore.Models.SimulationComponent.Data.Gearbox	74	34,26%	142	65,74%
{ } TUGraz.VectoCore.Models.SimulationComponent.Data	292	30,39%	669	69,61%
{ } TUGraz.VectoCore.FileIO	12	24,00%	38	76,00%
{ } TUGraz.VectoCore.Exceptions	10	20,83%	38	79,17%
{ } TUGraz.VectoCore.Utils	168	17,07%	816	82,93%
{ } TUGraz.VectoCore.FileIO.Reader.Impl	68	15,81%	362	84,19%
{ } TUGraz.VectoCore.Models.Simulation.Impl	96	15,12%	539	84,88%
{ } TUGraz.VectoCore.Models.SimulationComponent.Impl	542	14,82%	3116	85,18%
{ } TUGraz.VectoCore.Models.Simulation.Data	96	12,47%	674	87,53%
{ } TUGraz.VectoCore.FileIO.Reader	50	11,79%	374	88,21%
{ } TUGraz.VectoCore.FileIO.Reader.DataObjectAdaper	44	9,78%	406	90,22%
{ } TUGraz.VectoCore.Models.Declaration	58	3,17%	1769	96,83%
{ } Global Classes	0	0,00%	10	100,00%
{ } TUGraz.VectoCore.Configuration	0	0,00%	13	100,00%
{ } TUGraz.VectoCore.FileIO.DeclarationFile	0	0,00%	5	100,00%
{ } TUGraz.VectoCore.Models	0	0,00%	10	100,00%
{ } TUGraz.VectoCore.Models.Connector.Ports.Impl	0	0,00%	14	100,00%
{ } TUGraz.VectoCore.Models.SimulationComponent	0	0,00%	7	100,00%

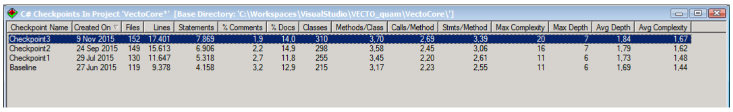
Figure 3: Screenshot of Test-coverage Report (Microsoft Visual Studio)

- more energy-efficient implementation of *Overspeed*: vehicle acceleration only due to road's slope, engine is on drag-curve
- Use of well-known logging framework (NLog) allows to configure logging output based on severity and namespace. Supports different logging targets
- Implemented in C#, requires .net Framework version 4.5
- Use of latest libraries for logging, parsing JSON files, and processing of PDF files
- Software Tests: 258 successful unit- and integration tests
  - Code coverage: 84 %
    The main parts of the code are covered by tests, parts with low coverage are mainly auto-generated equality methods and parsing of input data (cf. Figure 3)
- Code complexity[1]: currently the max. complexity is 20 (single method) and 10 methods have a complexity >10. The most complex methods are in the ShiftStrategy, the DriverStrategy, and the Powertrain builder; see Figure 4.

1.3 Open Issues

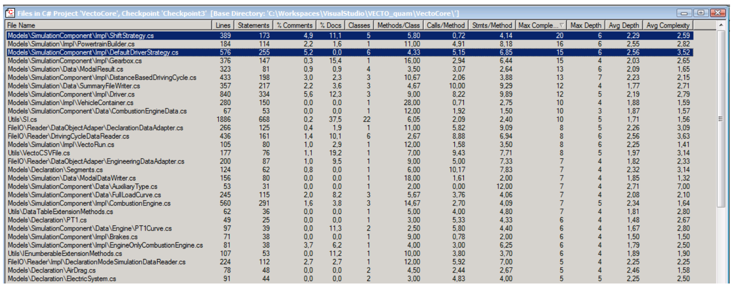
- Sanity checks on input data
- Refactoring of DefaultDriverStrategy to handle certain situations when braking is required during coasting to meet small target-speed changes during coasting.
- Status output to GUI/Console during simulation
- Advanced Driver Assist Systems
- Adaptation of component models to most recent ACEA Whitebook

[1] According to *SourceMonitor*



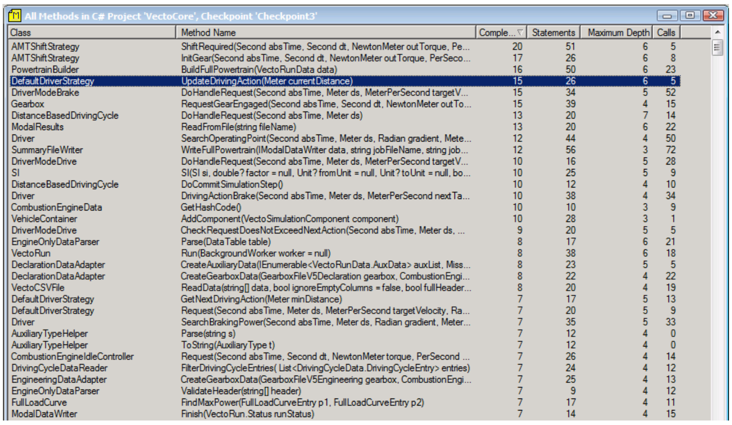
C# Checkpoints In Project 'VectoCore*' [Base Directory: 'C:\Workspaces\VisualStudio\VECTO_quam\VectoCore\']

Checkpoint Name	Created On ▽	Files	Lines	Statements	% Comments	% Docs	Classes	Methods/Class	Calls/Method	Stmts/Method	Max Complexity	Max Depth	Avg Depth	Avg Complexity
Checkpoint3	9 Nov 2015	152	17.401	7.869	1,9	14,0	310	3,70	2,69	3,39	20	7	1,84	1,67
Checkpoint2	24 Sep 2015	149	15.613	6.906	2,2	14,9	298	3,58	2,45	3,06	16	7	1,79	1,62
Checkpoint1	29 Jul 2015	130	11.647	5.318	2,7	11,8	255	3,45	2,20	2,61	11	6	1,73	1,48
Baseline	27 Jun 2015	119	9.378	4.158	3,2	12,9	215	3,17	2,23	2,55	11	6	1,69	1,44



Files in C# Project 'VectoCore', Checkpoint 'Checkpoint3' [Base Directory: 'C:\Workspaces\VisualStudio\VECTO_quam\VectoCore\']

File Name	Lines	Statements	% Comments	% Docs	Classes	Methods/Class	Calls/Method	Stmts/Method	Max Comple... ▽	Max Depth	Avg Depth	Avg Complexity
Models\SimulationComponent\Impl\ShiftStrategy.cs	389	173	4,9	11,1	5	5,80	0,72	4,14	20	6	2,29	2,59
Models\Simulation\Impl\PowertrainBuilder.cs	184	114	2,2	1,6	1	11,00	4,91	8,18	16	6	2,55	2,82
Models\SimulationComponent\Impl\DefaultDriverStrategy.cs	576	255	5,2	0,0	6	4,33	5,15	6,85	15	6	2,56	3,52
Models\SimulationComponent\Impl\Gearbox.cs	376	147	0,3	15,4	1	16,00	2,94	6,44	15	4	2,03	2,65
Models\Simulation\Data\ModalResult.cs	323	81	0,9	0,9	4	3,50	3,07	2,64	13	6	2,09	1,65
Models\SimulationComponent\Impl\DistanceBasedDrivingCycle.cs	433	198	3,0	2,3	3	10,67	2,06	3,88	13	7	2,23	2,15
Models\Simulation\Data\SummaryFileWriter.cs	357	217	2,2	3,6	3	4,67	10,00	9,29	12	4	1,77	2,71
Models\SimulationComponent\Impl\Driver.cs	840	334	5,6	12,3	3	9,00	8,22	9,89	12	5	2,19	2,79
Models\Simulation\Impl\VehicleContainer.cs	280	150	0,0	0,0	1	28,00	0,71	2,75	10	4	1,88	1,59
Models\SimulationComponent\Data\CombustionEngineData.cs	67	53	0,0	0,0	1	12,00	1,92	1,50	10	3	1,87	1,57
Utils\SI.cs	1886	668	0,2	37,5	22	6,05	2,09	2,40	10	5	1,71	1,56
FileIO\Reader\DataObjectAdaper\DeclarationDataAdapter.cs	266	125	0,4	1,9	1	11,00	5,82	9,09	8	5	2,26	3,09
FileIO\Reader\DrivingCycleDataReader.cs	436	161	1,4	10,1	6	2,67	8,88	6,94	8	6	2,56	3,63
Models\Simulation\Impl\VectoRun.cs	105	80	1,0	2,9	1	12,00	1,58	3,50	8	6	2,25	1,41
Utils\VectoCSVFile.cs	177	76	1,1	19,2	1	7,00	9,43	7,71	8	5	1,97	3,14
FileIO\Reader\DataObjectAdaper\EngineeringDataAdapter.cs	200	87	1,0	9,5	1	9,00	5,00	7,33	7	4	1,82	2,33
Models\Declaration\Segments.cs	124	62	0,8	0,0	1	6,00	10,17	7,83	7	4	2,32	3,14
Models\Simulation\Data\ModalDataWriter.cs	156	80	0,0	0,0	1	18,00	1,61	2,00	7	4	1,85	1,32
Models\SimulationComponent\Data\AuxiliaryType.cs	53	31	0,0	0,0	1	2,00	0,00	12,00	7	4	2,71	7,00
Models\SimulationComponent\Data\FullLoadCurve.cs	245	115	2,0	8,2	3	5,67	3,76	4,06	7	4	2,08	2,10
Models\SimulationComponent\Impl\CombustionEngine.cs	560	291	1,6	3,8	3	14,67	2,70	4,09	7	5	2,34	1,64
Utils\DataTableExtensionMethods.cs	62	36	0,0	0,0	1	5,00	4,00	4,80	7	4	1,81	2,80
Models\Declaration\PT1.cs	49	25	0,0	0,0	1	3,00	5,33	4,33	6	4	1,48	2,67
Models\SimulationComponent\Data\Engine\PT1Curve.cs	97	39	0,0	11,3	2	2,50	5,80	4,40	6	4	1,67	2,80
Models\SimulationComponent\Impl\Brakes.cs	71	38	0,0	0,0	1	9,00	0,78	2,00	6	4	1,50	1,50
Models\SimulationComponent\Impl\EngineOnlyCombustionEngine.cs	81	38	3,7	6,2	1	4,00	3,00	6,25	6	4	1,79	2,50
Utils\IEnumberableExtensionMethods.cs	107	53	0,0	11,2	1	10,00	3,80	3,70	6	4	1,89	1,90
FileIO\Reader\Impl\DeclarationModeSimulationDataReader.cs	224	112	2,7	2,7	1	12,00	5,92	7,00	5	4	2,25	2,25
Models\Declaration\AirDrag.cs	78	48	0,0	0,0	2	4,50	2,44	2,67	5	4	2,46	1,58
Models\Declaration\ElectricSystem.cs	91	44	0,0	0,0	2	3,00	4,83	4,00	5	5	2,25	2,50



All Methods in C# Project 'VectoCore', Checkpoint 'Checkpoint3'

Class	Method Name	Comple... ▽	Statements	Maximum Depth	Calls
AMTShiftStrategy	ShiftRequired(Second absTime, Second dt, NewtonMeter outTorque, Pe...	20	51	6	5
AMTShiftStrategy	InitGear(Second absTime, Second dt, NewtonMeter outTorque, PerSeco...	17	26	6	8
PowertrainBuilder	BuildFullPowertrain(VectoRunData data)	16	50	6	23
DefaultDriverStrategy	UpdateDrivingAction(Meter currentDistance)	15	26	6	5
DriverModeBrake	DoHandleRequest(Second absTime, Meter ds, MeterPerSecond targetV...	15	34	5	52
Gearbox	RequestGearEngaged(Second absTime, Second dt, NewtonMeter outTo...	15	39	4	15
DistanceBasedDrivingCycle	DoHandleRequest(Second absTime, Meter ds)	13	20	7	14
ModalResults	ReadFromFile(string fileName)	13	20	6	22
Driver	SearchOperatingPoint(Second absTime, Meter ds, Radian gradient, Mete...	12	44	4	50
SummaryFileWriter	WriteFullPowertrain(IModalDataWriter data, string jobFileName, string job...	12	56	3	72
DriverModeDrive	DoHandleRequest(Second absTime, Meter ds, MeterPerSecond targetV...	10	16	5	28
SI	SI(SI si, double? factor = null, Unit? fromUnit = null, Unit? toUnit = null, bo...	10	25	5	9
DistanceBasedDrivingCycle	DoCommitSimulationStep()	10	12	4	10
Driver	DrivingActionBrake(Second absTime, Meter ds, MeterPerSecond nextTa...	10	38	4	34
CombustionEngineData	GetHashCode()	10	10	3	9
VehicleContainer	AddComponent(VectoSimulationComponent component)	10	28	3	1
DriverModeDrive	CheckRequestDoesNotExceedNextAction(Second absTime, Meter ds, ...	9	20	5	5
EngineOnlyDataParser	Parse(DataTable table)	8	17	6	21
VectoRun	Run(BackgroundWorker worker = null)	8	38	6	18
DeclarationDataAdapter	CreateAuxiliaryData(IEnumerable<VectoRunData.AuxData> auxList, Miss...	8	23	5	5
DeclarationDataAdapter	CreateGearboxData(GearboxFileV5Declaration gearbox, CombustionEngi...	8	22	4	22
VectoCSVFile	ReadData(string[] data, bool ignoreEmptyColumns = false, bool fullHeader...	8	20	4	19
DefaultDriverStrategy	GetNextDrivingAction(Meter minDistance)	7	17	5	13
DefaultDriverStrategy	Request(Second absTime, Meter ds, MeterPerSecond targetVelocity, Ra...	7	20	5	9
Driver	SearchBrakingPower(Second absTime, Meter ds, Radian gradient, Meter...	7	35	5	33
AuxiliaryTypeHelper	Parse(string s)	7	12	4	0
AuxiliaryTypeHelper	ToString(AuxiliaryType t)	7	12	4	0
CombustionEngineIdleController	Request(Second absTime, Second dt, NewtonMeter torque, PerSecond ...	7	26	4	14
DrivingCycleDataReader	FilterDrivingCycleEntries(List<DrivingCycleData.DrivingCycleEntry> entries)	7	24	4	12
EngineeringDataAdapter	CreateGearboxData(GearboxFileV5Engineering gearbox, CombustionEngi...	7	25	4	13
EngineOnlyDataParser	ValidateHeader(string[] header)	7	9	4	12
FullLoadCurve	FindMaxPower(FullLoadCurveEntry p1, FullLoadCurveEntry p2)	7	17	4	11
ModalDataWriter	Finish(VectoRun.Status runStatus)	7	14	4	15

Figure 4: Screenshots of code metrics (SourceMonitor): Checkpoints over time (top), list of classes sorted by complexity (middle), List of methods sorted by complexity (bottom)

2 How to Submit a Bug Report

In case you encounter a bug or Vecto 3 is not behaving as you would expect it is of utmost importance that the developers can reproduce your results[2]. All bugs should be submitted as Jira Issue (either *Bug* or *Feature/Task*) to maintain traceability.

Every bug-report should contain a detailed description containing:

- What did you do
- What is your expected outcome
- What is the actual outcome
- Which Vecto version did you use? (Version number and Build number)
  The version number is printed when you start vectocmd in verbose mode
- Screenshots if necessary

In addition a detailed log of your simulation is necessary. Please follow these instruction to create a log-file and the mod-files:

- **Start a command-line**
  Press *Windows-r* and type `cmd`, change to the directory where you extracted Vecto 3.0
- Delete old log-files:
  `del logs\*`
- Start the simulation using Vecto-Commandline:
  `vectocmd.exe -t -vvvv -mod <jobfile.vecto>`
  it is important to use the option `-vvvv` and `-mod` which enable verbose logging and output of the mod-data file (`.vmod`.

  If logging is enabled, the simulation runs are executed sequentially. If the error occurs only in some simulation runs (i.e., driving cycle and loading) you can cancel the other simulation runs by pressing *Ctrl-c*. (Note: if verbose logging is enabled the simulation takes much longer)
- File a new Jira issue[3] with a detailed description (see above)
- Attach all data necessary to reproduce your results to the Jira issue, or if the data is confidential you can directly send it to the developers.
  - All input files
  - All Log-files in the directory `logs` (compressed)
  - Mod-files (stored in the directory of the Vecto job)

[2] http://www.chiark.greenend.org.uk/~sgtatham/bugs.html

[3] https://webgate.ec.europa.eu/CITnet/jira/secure/CreateIssue!default.jspa