

Genny How-To

The QuickStart Genny How-To is as follows

Genny requires Python v2.2 or higher.

Unzip the archive wherever you like, all paths are relative to the installation directory.

the directory structure is as follows

- [./docs](#) - genny documentation (although incomplete)
- [./docs/dtd](#) - genny LDD dtd
- [./docs/srcdoc](#) - source documentation
- [./imports](#) - genny python modules
- [./projects](#) - a place for project directories

Each project has a name and that name will be a subdirectory under the [./projects](#) directory. When you want to install a new version of Genny you'll just be able to copy your projects from the old Genny installation directory to the new one.

Under each projects directory are the following directories

- [./projects/project-name/input](#) - all of the input files that Genny uses (Tab delim, XML LDD, boilerplate text, etc.)
- [./projects/project-name/input/docbook-components](#) - individual files with the same name as the CONTAINERS in the LDD are placed here with docbook formatted xml that gets merged with the LDD information to create the IGuide chapter in docbook format
- [./projects/project-name/output](#) - the primary output of Genny goes here (DTDs, HTML LDD, schemas, etc.)
- [./projects/project-name/output/ele](#) - MISMO ele style DTD fragments will be created here for each container
- [./projects/project-name/output/logs](#) - as Genny does its work it creates a set of logs
- [./projects/project-name/output/templates](#)
- [./projects/project-name/output/templates/python](#) - some initial work to auto generate a python class that knows how to produce and consume XML based on the LDD
- [./projects/project-name/output/templates/cobol](#) - some incomplete work on auto generating cobol FD lineups from the LDD

So after you've unzipped the archive, execute the following command

```
x:\ python genny.py PROJECT-NAME --new
```

where PROJECT-NAME is the name for your project and Genny will create the correct directory and file templates under the [./projects](#) directory.

Take your completed MS-Excel spreadsheet (from the template that I sent) and save it as a tab-delimited text file as
FILENAME.txt in the [./projects/PROJECT-NAME/input](#) directory

where FILENAME is the name you want for your output files (e.g. InvestorReporting-v2-2-DRAFT)

Then execute the following command

```
x:\ python genny.py PROJECT-NAME FILENAME
```

where PROJECT-NAME and FILENAME are the names you used above.

Genny How-To

If all goes well you'll see a bunch of messages scroll by about all the wonderful things that Genny is generating.

When you enter the above command Genny is doing the following:

Converting the tab-delimited LDD into an XML version of the LDD

Parsing the XML LDD

Processing the LDD to produce the commented and uncommented DTDs and ELE's

Processing the LDD to produce the SQL DDL

Processing the LDD to produce the commented and uncommented schemas

Processing the LDD to produce the python template code

Processing the LDD to produce the cobol template code

Processing the LDD to produce the HTML version of the LDD

Processing the LDD and the [./projects/PROJECT-NAME/input/docbook-components](#) to produce the Chapter3.xml docbook IGuide documentation

If for any reason you want to delete all of the output for a given project you can

```
x:\ python genny.py PROJECT-NAME --cleanup
```

and all the files in the output directories for that project will be deleted.

I've included the Servicing Transfer project directory in the archive so you can test your install by

```
x:\ python genny.py SERVICING-TRANSFER servicing-transfer-v2-2-DRAFT
```

or by just running the ST.bat file.

Please look at the Servicing Transfer project as an example of how to do things.

The most common thing that can go wrong if the tab-delimited input is broken in some way (e.g. tabs or hard returns in the cells) .

Other things you should be aware of regarding the spreadsheet are

- * all rows must be sorted by CONTAINER (all data items for a given container must be consecutive in the spreadsheet)

- * all rows must have a CONTAINER and a DATA ITEM name (even rows for enumerated data values)

- * the order of the spreadsheet will be the order of the output

- * the valid data types are

(enumerated,Alphanumeric,DateTime,Date,Money,Numeric,xsd:Boolean,Fixed,ID,IDREF,IDREFS)

- * the parent and child container lists are what provides the document structure,

they are comma separated lists of the form

container:x:y,container2:x:y

where x = minoccurs (0) and y = maxoccurs (1 or u) u is unbounded

Genny How-To

Acknowledgements:

- This product includes software developed by Laurence Tratt. His *ASV.py* python module parses the tab-delimited input files.
- Hamish B Lawson's *TextFormatter.py* python module provides some of the formatting capability for the DTD output.