Australasian Legal Information Institute (AustLII)

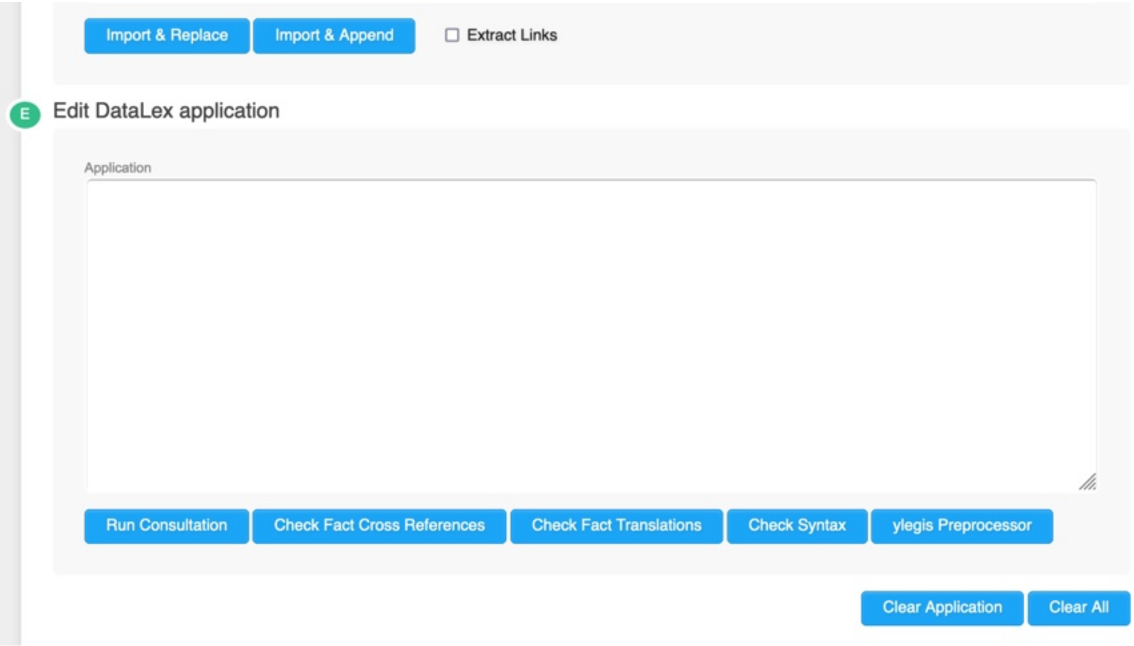
Quick Tutorial – Building *DataLex* 'Rules as Code' apps

Graham Greenleaf, Philip Chung & Andrew Mowbray, 26 November 2021

1 *Creating a new code-base*

Creating a new code-base using the free-standing Development Tools

- Go to http://www.datalex.org/dev/tools/ (*DataLex Application Development Tools* page*)*.
- No login is necessary. This page enables creation and test-runs of small applications using the DataLex software. However, it does not allow test apps to be saved.
- There are two ways to start writing an app:
  - (i) Simply start writing in the 'Edit DataLex application' editing screen, following the instructions in parts (2)-(7) of this Tutorial.
  - (ii) If you know what section of an Australian Act you would like to start with, go to the 'Import legislative section (available on AustLII)' and enter the Act name, jurisdiction and section, then start editing as in (i) above.
- The rest of this Tutorial explains the purpose of each selectable button below.



Testing the yscript code-base

- Select the 'Run Consultation' button when you are ready to test the yscript code-base.
- After running your app (successfully or unsuccessfully) you can use your browser to come back to the Tools page to make further edits to your code-base.
- If your app does not run as intended, use the 'Check Fact Cross References' and 'Check Fact Translations' and 'Check syntax' buttons to run diagnostics (see 5 below) to identify problems. Edit and run again.

- If you have spent any significant time developing a test code-base, you might want to save a copy in a word processor or other file, in case the browser malfunctions, or if you want to use the code-base again after you have quit the browser.

Refinements

- To add another section of an Act from AustLII to an existing code-base, use the 'Import and Append' button after specifying the additional section.
- To over-write and erase an existing KB with a new section from AustLII, use the 'Import and Replace' button.
- To clear an existing code-base and start again, use 'Clear Application.

The ylegis pre-processor [Optional]

- The newest software in the DataLex tools is the ylegis pre-processor, by which conversion of legislative provisions directly to yscript code can be undertaken.
- Simply import one or more sections of an Act into the application window and select the ylegis button.
- Read the two-page explanation 'The DataLex legislation pre-processor for rules as code'. Decide whether the yscript code produced by ylegis needs modification.

2 The structure of rules in yscript

- Simplest syntax: 'RULE name of rule PROVIDES' statements
- The **keyword 'RULE'** indicates the start of a new rule.
  - Use GOAL RULE (at least twice) to give users a choice of starting goals
- The **name of the rule**: must be unique; often name of Act and section
  - Names of rules are used to give users a choice of GOALs
- The **keyword 'PROVIDES'**, indicates the start of the body of the rule
- The **statement(s)** (one or more) are the content of the rule, and consist of declarations; a simple and common statement is in the form 'IF condition THEN conclusion'.
- A **declaration** contains keywords and descriptors; Keywords are logical connectors between descriptors; declarations assert that these connections are true.
- A **keyword** is an English-like expression which asserts some logical connection between two descriptors, or asserts some characteristic about one descriptor. Keywords are written in FULL CAPITALS.
- A **descriptor** is a term or phrase which describes some object, event etc. A descriptor is a proposition which is capable of being true or false (ie 'truth bearing propositions'). Write them in lower case with normal capitalisation, and be consistent in multiple uses

3 Commonly used keywords (full list is attached)

Keywords	***Meaning***	***Types of uses***
ONLY IF		use for 'means' definitions
IF .... THEN		use for 'includes' definitions
IF ... THEN .... ELSE		use for alternative conclusions
AND	normal 'and'	
OR	normal 'or'	
AND/OR	OR with higher binding strength (like parentheses)	use where an OR follows an AND in a declaration

4 Named subject keywords

- Use named subject as much as possible to make consultation dialogues more realistic and customised to a particular problem.
- Subject declarations can appear anywhere in a rule-base, but usually go at the start.
- Make sure the named subject is referred to consistently (use 'Check Fact Cross References' to be sure).
    - PERSON – where the subject is a human person, not a legal person
        - Example: PERSON the intestate
    - THING – where the subject is not a human person
        - Example: THING the agreement
    - PERSONTHING – where the subject can be either human person or a legal person
        - PERSONTHING the first party

5 Running and checking rule-bases

- To run a code- base so as to test how it is working, select 'Run Consultation'.
- Use 'Check Fact Cross References' to check whether descriptors have been used consistently across rules. Similarly named but not identical descriptors are one of the main reasons why rules do not run as expected.
- Use 'Check Fact Translations' to ensure that descriptors used in your rule-base are being translated into readily understood forms as questions, positive assertions and negative assertions. If this is not occurring, consider modifying the descriptor. ('applies' is a useful term to use in descriptors concerning legislation).
    - 'Verbs' declarations can be used to correct mis-behaving verbs
    - If necessary, PROMPT and TRANSLATE can provide alternatives
- Use 'Check Syntax' to ensure that the many keywords available in yscript (see 8. Below) have been used correctly. It will give alert that a keyword is missing.
- In the Datalex Manual, see *2.5 Running a DataLex application* for debugging details.

6 Hypertext links from rule-bases

- Links should be created automatically to references to Acts or sections, or to citations of cases or articles, if they appear in the statements contained in a RULE (not its name).
- Keywords LINK and TO are used to specify in a rule-base that a particular word or phrase is always to appear as a hypertext link to a particular URL
    - Example: LINK document of an agency TO http://www.austlii.edu.au/au/legis/cth/consol_act/foia1982222/s4.html
- LINK ...TO ... can be used to create links from a rule-base to the results of any search (not only over AustLII)
    - Example: LINK official document of a Minister TO http://www.austlii.edu.au/cgi-bin/sinosrch.cgi?query=%20official+document+of+a+Minister%20

7 Linking rule-bases on multiple pages ('cooperative inferencing')

- The use of the keyword INCLUDE in a rule-base, followed by the URL of another page containing a DataLex code-base, will cause the second rule-base to be loaded with the first code-base, and the two run together.
- Multiple rule-bases can be INCLUDEd.
- The same named subject declarations can occur in two INCLUDEd rule-bases
- INCLUDE can link to rule-bases located in DataLex Communities, or at another URL
    - Example: INCLUDE http://austlii.community/wiki/DataLex/FOI4KB

8 *List of Main Keywords*

The most detailed explanation of all keywords is in the *DataLex Developer's Manual* <http://austlii.community/foswiki/pub/DataLex/WebHome/datalex-developers-manual.pdf>. This is a selection of the keywords explained there. -.

8.1. Rule types

RULE
GOAL
PROCEDURE
BACKWARD
FORWARD
DAEMON
DOCUMENT
EXAMPLE

8.1.1. Attribute types

BOOLEAN
INTEGER
REAL
DOLLAR
SEX
STRING
DATE

8.1.2. Document types

(Documents only)
PARAGRAPH
LINE
TEXT

8.1.3. Named subject types

THING
PERSONTHING
PERSON

8.1.4. Translation operators

TRANSLATE - AS
PROMPT
VERBS

8.1.5. Statement operators

ASSERT
DETERMINE
IF - THEN – ELSE
IF - THEN
ONLY IF
IS
WHILE - DO
REPEAT - UNTIL
CALL
BEGIN - END

8.1.6. Expression operators

(Pre) Unary Operators

NOT
MINUS
PLUS
DAY
MONTH
YEAR

(Post) Unary Operators

DAYS
WEEKS
MONTHS
YEARS

Binary Operators

DIVIDED BY
TIMES
PLUS
MINUS

IN
EQUALS
NOT EQUALS
IS GREATER THAN
IS LESS THAN
IS GREATEREQUAL THAN
IS LESSEQUAL THAN

AND/OR
AND/OR/WITH
AND
AND/WITH
OR
OR/WITH

8.1.7. Layout operators

(Documents only)
NUMBERED
LEVEL

8.1.8. Miscellaneous keywords

RANGE - TO
VERBS
PROVIDES
ORDER