

University of New South Wales Law Research Series

**BUILDING DATALEX DECISION SUPPORT
SYSTEMS A TUTORIAL ON RULE---BASED
REASONING IN LAW**

**GRAHAM GREENLEAF, PHILIP CHUNG AND ANDREW
MOWBRAY**

[2017] *UNSWLRS* 68

UNSW Law
UNSW Sydney NSW 2052 Australia

E: unswlrs@unsw.edu.au
W: <http://www.law.unsw.edu.au/research/faculty-publications>
AustLII: <http://www.austlii.edu.au/au/journals/UNSWLRS/>
SSRN: <http://www.ssrn.com/link/UNSW-LEG.html>



[Australasian Legal Information Institute \(AustLII\)](#)

Building *DataLex* decision support systems *A tutorial on rule-based reasoning in law*

Graham Greenleaf, Philip Chung & Andrew Mowbray

4 September 2017*

1.	Aims of the tutorial -----	3
1.1.	Purposes -----	3
1.2.	The Developer's Manual -----	3
1.3.	Where to start: The DataLex Community page -----	3
2.	Demonstrations of DataLex inferencing applications -----	4
2.1.	The user interface - elements to test -----	4
3.	Simple backward & forward chaining rules -----	5
3.1.	A set of rules ... -----	6
3.2.	... and a problem -----	6
3.3.	Testing forward and backward chaining -----	6
4.	Writing a simple legal rule-base -----	7
4.1.	Choose an area of law with some legislation that interest you -----	7
4.2.	Creating a new topic in the DataLex Community -----	7
4.3.	Some starting points in writing rules -----	8
4.4.	Start with the simplest rule representing a section -----	10
5.	Running rule-bases -----	10
6.	Extending your rule-base -----	10
6.1.	Add some more rules to deepen the section structure -----	10
6.2.	Personalise your dialogues - named subjects -----	11
7.	Error checking in your rule-base -----	12
7.1.	Check Fact Cross References -----	12
7.2.	Check Fact Translations -----	13
8.	Expanding your rule-base -----	14
9.	Hypertext links from rule-bases -----	16
9.1.	Automatic links to AustLII legislation -----	16
9.2.	Explicit links in a rule-base (the LINK ... TO ... keywords) -----	16
9.3.	Searches from rule-bases -----	16
10.	'Co-operative inferencing' - rule-bases on multiple pages -----	17
10.1.	The INCLUDE keyword -----	17
10.2.	At least one GOAL rule must be specified -----	17
10.3.	LINK ... TO... and 'co-operative' rulebases -----	17
11.	Document generation -----	17
11.1.	DOCUMENT rules -----	18
11.2.	Using RULES with DOCUMENTS -----	18
11.3.	Hypertext links from documents -----	18
11.4.	Example - a simple will generator -----	18
12.	Case-based inferencing using PANNDATA -----	19
12.1.	Example -----	20

* Please check online (AustLII or SSRN) for updated versions of this Tutorial.

1. Aims of the tutorial

This tutorial* aims to provide a practical introduction to rule-based inferencing (reasoning) systems in law, developed using AustLII's DataLex software, and utilising the world-wide-web as a platform. It covers the development of a simple backward and forward chaining rule-base; publishing and running a rule-base on AustLII's DataLex community website; integration of rule-bases with hypertext and search resources on AustLII and elsewhere; and 'co-operative inferencing' (rule-bases by different authors distributed across web sites but operating in tandem). Document generation and case-based inferencing using the DataLex software is also covered in brief.

1.1. Purposes

From mid-2017, AustLII is integrating and further developing a set of tools, and development approaches, for the purpose of building legal decision support systems. These include the DataLex inferencing software, the AustLII Communities framework for development of both inferencing applications and supporting interpretative materials, and methods of integration with AustLII's databases. Together, these make up the DataLex legal decision support system.

The primary purpose for which this system is being developed is to assist providers of free legal advice to develop sustainable free legal advisory systems to assist them (and their clients) in their work. The DataLex software is not at this stage being provided as open source software, but rather as 'free use for free legal advice providers'.

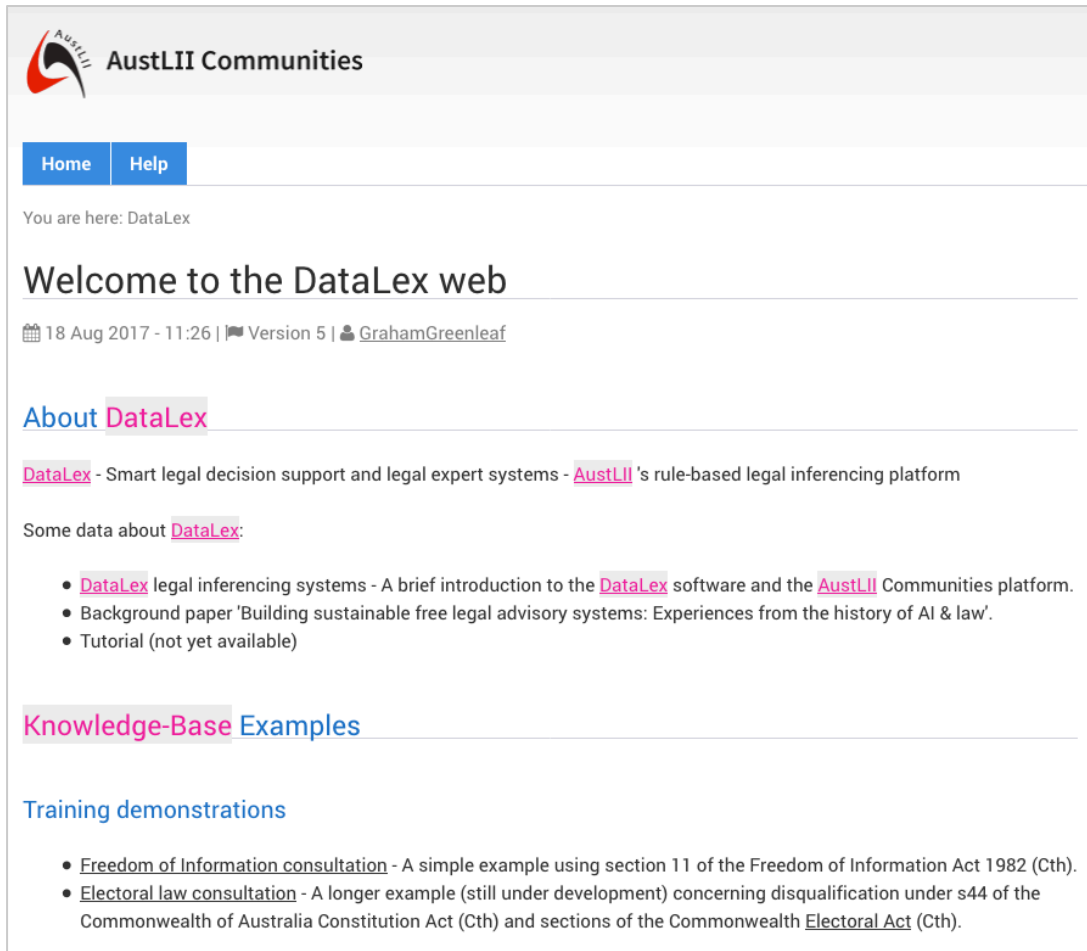
1.2. The Developer's Manual

These tutorial exercises are designed to be read with the *DataLex Developer's Manual* (currently under revision). Key parts of the Manuals are summarised in this Tutorial. A previous edition, *The Wysh Developer's Manual* is available at <<https://papers.ssrn.com/abstract=2986144>>, and is accurate in relation to rule-base development but out-of-date in relation to interfaces.

1.3. Where to start: The DataLex Community page

Most of the resources needed for the tutorial can be located from the DataLex Community page located on the AustLII Communities website: <<http://austlii.community/wiki/DataLex/>>.

* Geoffrey King and Simon Cant contributed to an earlier version of this tutorial, 'Rule-based inferencing over the world-wide-web'.



AustLII Communities

[Home](#) [Help](#)

You are here: DataLex

Welcome to the DataLex web

📅 18 Aug 2017 - 11:26 | 🗨️ Version 5 | 👤 [GrahamGreenleaf](#)

About DataLex

[DataLex](#) - Smart legal decision support and legal expert systems - [AustLII](#)'s rule-based legal inferencing platform

Some data about [DataLex](#):

- [DataLex](#) legal inferencing systems - A brief introduction to the [DataLex](#) software and the [AustLII](#) Communities platform.
- Background paper 'Building sustainable free legal advisory systems: Experiences from the history of AI & law'.
- Tutorial (not yet available)

Knowledge-Base Examples

Training demonstrations

- [Freedom of Information consultation](#) - A simple example using section 11 of the Freedom of Information Act 1982 (Cth).
- [Electoral law consultation](#) - A longer example (still under development) concerning disqualification under s44 of the Commonwealth of Australia Constitution Act (Cth) and sections of the Commonwealth [Electoral Act](#) (Cth).

2. Demonstrations of DataLex inferencing applications

The purpose of this exercise is to familiarise you with the DataLex software to be used in the tutorial. Examples on freedom of information and electoral law are used to illustrate some rule-based applications that can be developed.

2.1. *The user interface - elements to test*

When testing examples, please test the use of all the system functions, including the buttons for 'Facts' (what have you told the system), 'Conclusions' (what conclusions has the system derived), 'Why' (why is the current question being asked), 'What if' (what conclusions will be derived from a hypothetical fact) and 'Uncertain' (if a fact is not essential, inferencing will continue).

DataLex session

This is a demonstration of rule-based inferencing and must not be relied upon or used for the purposes of legal advice.

1) What is the name of the nominee ?

What If?

Things to note about the interface:

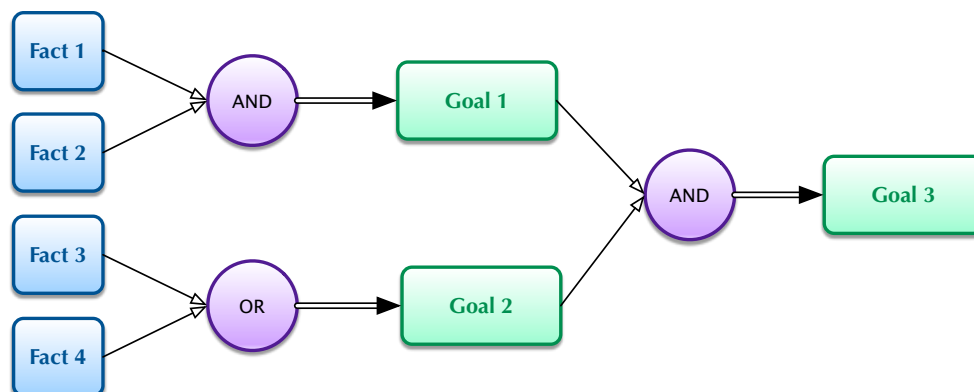
- To choose a goal to be evaluated, type the number of that goal in the window;
- To forget a single fact, first display the facts (using the 'Facts' button), then enter the number to be forgotten and click on 'Forget';
- Do not use the browser's 'back' button in order to change facts - it won't work and may give bizarre results

3. Simple backward & forward chaining rules

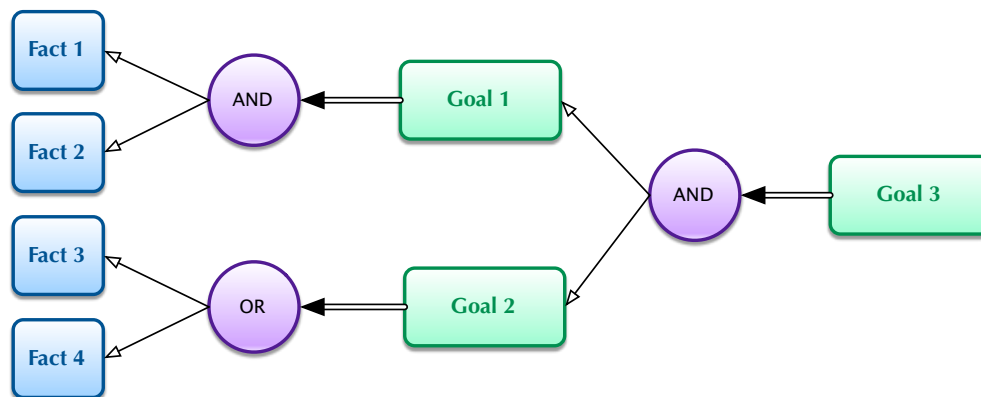
'Rule-based reasoning' is a method of declaring knowledge about a particular subject, without having to encode the order in which that knowledge is used to produce a result. We therefore refer to a 'declarative knowledge-base' rather than procedural programming.

Some simple symbolic examples are used to make it easier to see the logic behind backward and forward chaining without the distraction of real facts.

Forward chaining or data-driven reasoning uses available data (or 'facts') and inferencing rules to derive further 'facts'.



Backward chaining or goal-driven reasoning looks at the result and attempts to figure out the reasons or conditions on which a particular result has been derived. It addresses the question: “why has this happened?”



3.1. A set of rules ...

Assume that you have 5 rules which allow you to reach conclusions about whether 8 propositions (represented as A to F) are true or not:

Rule	Conditions	Conclusion
Rule 1	E and F	A
Rule 2	A and (B or C)	D
Rule 3	E or G	C
Rule 4	H and G and C and not F	A
Rule 5	F and not E	not D

3.2. ... and a problem

If you were asked whether D is true or not, given that you know that E, H and G are true, but F and B are false, how would you go about working out an answer. **Write down each step in your reasoning.**

3.3. Testing forward and backward chaining

There are 3 DataLex knowledge-bases (small applications) that you can use, accessible from the DataLex home page, entitled ‘Goal Tests’:

- (i) *goaltest* (forward and backward chaining)
- (ii) *goalbackward* (backward chaining only)
- (iii) *goalforward* (forward chaining only)

You can now use these files (applications) to test how DataLex’s inferencing mechanisms work. For example, assume the following:

- the objective (goal) is D;
- E, H and G are correct (true);
- F and B are not correct (false).

First, try to work out on paper what you think the DataLex applications should do, then run each of the three versions of the application and watch to see if each one does what you expect. (If you wish, you can then try out objective and fact combinations of your own.)

Note the order in which rules and attributes are evaluated in DataLex's backward-chaining mode:

- (i) rules are evaluated top-down from the start of the rule-base; and
- (ii) attributes are evaluated left to right within a rule

4. Writing a simple legal rule-base

The purpose of this exercise is for you to work out how to convert a section or two of an Act into a small rule-base which *DataLex* can run to infer conclusions.

4.1. Choose an area of law with some legislation that interest you

For the rest of this Tutorial, you need to choose an area of law on which to develop a small DataLex application. If it is an area of law with some Australian legislation, that will make it possible to use the automated links from inferencing dialogues to AustLII databases that are covered later.

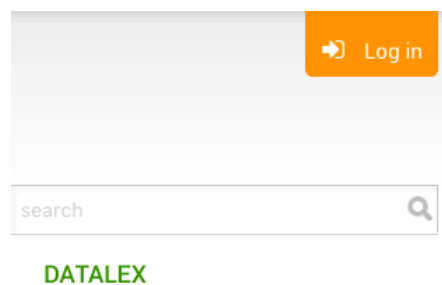
Find on the web any Act that interests you, choose a couple of fairly simple and inter-related sections, preferably sections that create obligations or offences. Read the rest of this part, and then convert the section(s) into a rule or set of rules that will run under the DataLex software.

The easiest way to proceed may be to save (or cut and paste) the section on which you wish to start into a new topic in the DataLex Community, and then edit it.

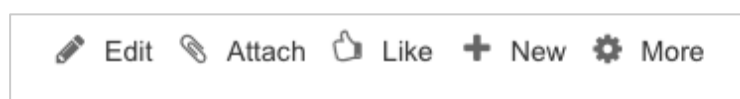
4.2. Creating a new topic in the DataLex Community

The DataLex Community (part of the AustLII Communities) is a collaborative closed wiki-like platform used in the DataLex system for creating and editing knowledge-bases or rule-bases.

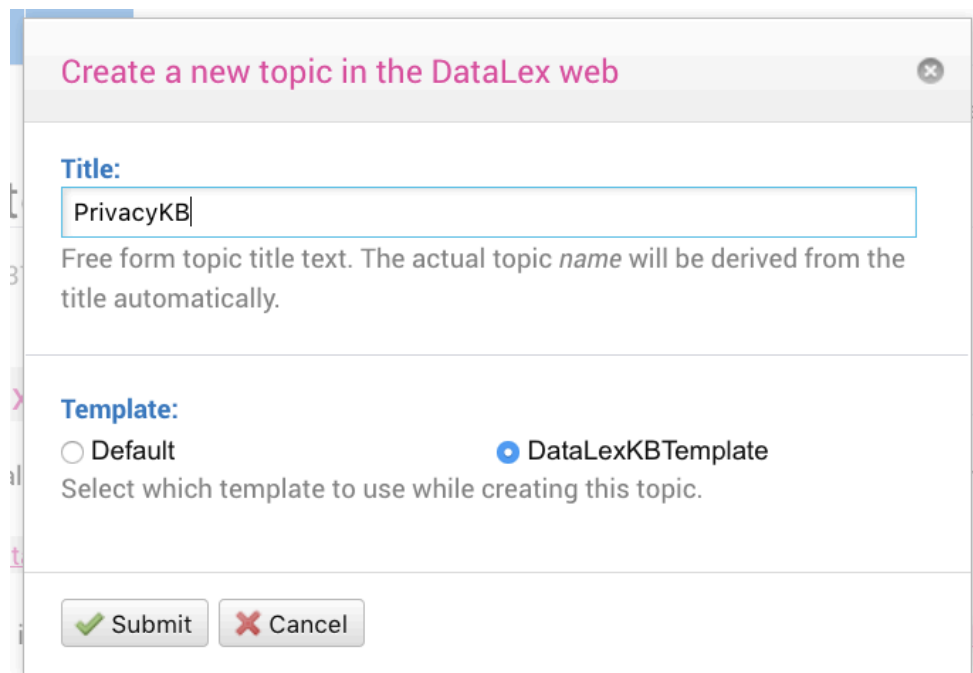
Log into the DataLex Community site by clicking on the 'Log in' button on the top right hand corner.



Once the login process is verified, an extra row of buttons for editing and creating new rule-bases will appear on the page.



Click on 'Edit' to edit existing rule-bases. To start a new rule-base (or topic), click on the 'New' button. The following window will appear:



Enter the 'Title' of the rule-base to be constructed. The example here is 'PrivacyKB' for a privacy knowledge-base. In the 'Template' section, select 'DataLexKBTemplate' (not 'Default'). Then, click on 'Submit'.

4.3. Some starting points in writing rules

These notes are a simplified version of what is in the *Developer's Manual*.

Form of a simple rule

In its simplest form, a rule contains four elements:

- (i) **the keyword 'RULE'**, indicating the start of a new rule (in default of any specification, the type of this rule is that it will be both backward chaining and forward chaining);
- (ii) **the name of the rule** (usually just the name of the Act and section that it paraphrases); The name of a rule should differ from that of any other rule in the rule-base.;
- (iii) **the keyword 'PROVIDES'**, indicating the start of the body of the rule; and
- (iv) **the statement(s)** which make up the inferencing content of the rule (one or more statements). Statements may be thought of as the content of rules. A rule contains one or more statements. Statements consist of declarations. One of the simplest forms of a statement is 'IF condition THEN conclusion'.

The simplest syntax for a rule is therefore as follows:

RULE name **PROVIDES** *statements*

The example below shows three rules, each with one moderately complex statement.

Declarations contain keywords and descriptors

Statements, the content of rules, are made up of declarations. A declaration consists of various *keywords* which are used to join together, in a logical form, a number of *descriptors*, which are simply terms or phrases used to describe some object, event etc.

A descriptor must be a proposition which is capable of being true or false (ie 'truth bearing propositions'). DataLex rule-bases enable reasoning by propositional logic.

Keywords

Keywords give rules the logical structure used by DataLex to draw inferences. They are written in FULL UPPER CASE so DataLex can distinguish them from their equivalents in ordinary words (which may occur in descriptors). Examples of important keywords, or sets of keywords are:

ONLY IF
 IF THEN
 IF ... THEN ELSE
 IS
 AND
 OR
 PLUS
 MINUS
 PERSON
 THING.

These and other keywords have functions in a DataLex knowledge-base which is very similar to their normal linguistic function as words. This correspondence is a large part of what gives DataLex applications a 'quasi natural language' or 'English like' syntax.

Descriptors

Descriptors may be any sequence of words or symbols but must not contain keywords (they can contain the lower case versions of them). Descriptors are generally written in lower case, with normal capitalisation. In the example below, some descriptors used are 'a person has a legally enforceable right under s11 to obtain access to a document', 's11(a) applies' and 'the document is not an exempt document'. These are all attributes, but other descriptors include constants and named subjects.

Example of a rule-base of 3 rules - FOI Act s11

The *Freedom of Information Act 1982* (Cth) s11 reads:

11. Subject to this Act, every person has a legally enforceable right to obtain access in accordance with this Act to -
 (a) a document of an agency, other than an exempt document; or
 (b) an official document of a Minister, other than an exempt document.

A rule-base of 3 rules consisting solely of this section could read as follows. The rules have been (over-)simplified, for demonstration purposes, by ignoring the words 'subject to this Act' in s11. Note that you could alternatively have written one much more complex rule.

RULE Freedom of Information Act 1982 (Cth) s11 PROVIDES
 a person has a legally enforceable right under s11 to obtain access to a document ONLY IF
 s11(a) applies OR
 s11(b) applies

RULE Freedom of Information Act 1982 (Cth) s11(a) PROVIDES

s11(a) applies ONLY IF
 the document is a document of an agency AND
 the document is not an exempt document

RULE Freedom of Information Act 1982 (Cth) s11(b) PROVIDES
 s11(b) applies ONLY IF
 the document is an official document of a Minister AND
 the document is not an exempt document

4.4. *Start with the simplest rule representing a section*

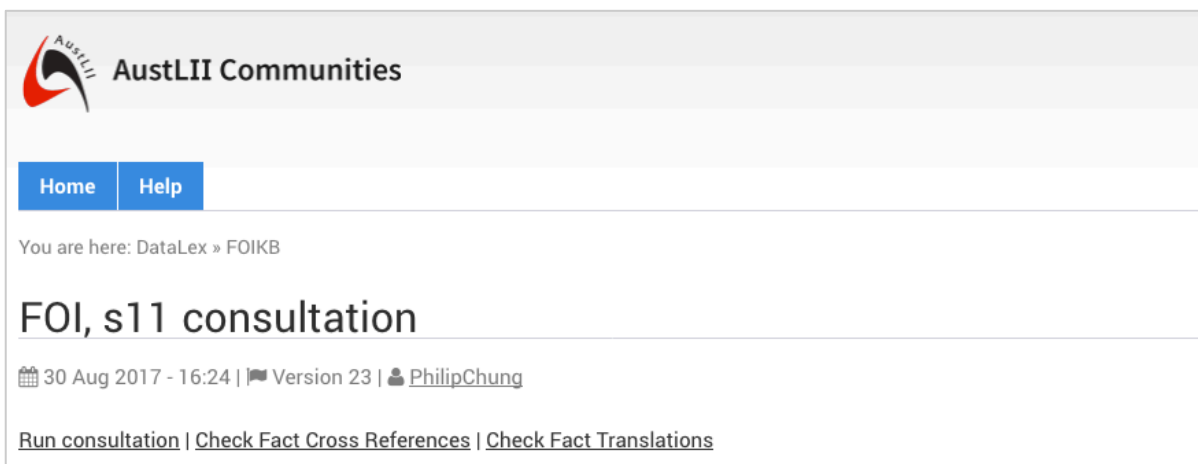
The rules above have been (over-)simplified, for demonstration purposes, by ignoring the words 'subject to this Act' in s11 (To include that phrase requires a rule involving interpretation of the whole FOI Act.).

One recommended way to start is to create a rule which explicitly models the structure of the section being represented by the rule, such as:

RULE Freedom of Information Act 1982 (Cth) s11 PROVIDES
 a person has a legally enforceable right under s11 to obtain access to a document ONLY IF
 s11(a) applies OR
 s11(b) applies

'Applies' is a good verb to use with sections. It is asserted as '... does apply', negated as '... does not apply', and made interrogative as 'Does apply?'

5. Running rule-bases



The screenshot shows the AustLII Communities website interface. At the top left is the AustLII logo. Below it are 'Home' and 'Help' buttons. A breadcrumb trail reads 'You are here: DataLex » FOIKB'. The main heading is 'FOI, s11 consultation'. Below the heading is a timestamp '30 Aug 2017 - 16:24 | Version 23 | PhilipChung'. At the bottom of the page, there is a 'Run consultation' button and two links: 'Check Fact Cross References' and 'Check Fact Translations'.

Click 'Run consultation' to start the consultation using the rule-base found on the page.

6. Extending your rule-base

This section provides some basic steps toward a more sophisticated rule-base.

6.1. *Add some more rules to deepen the section structure*

To make backward chaining work, create some additional rules which have as their conclusions the structural elements of the section that are represented as attributes in your first rule (eg 's11(a) applies' and 's11(b) applies'. For example, these rules can be added to the above rule:

RULE Freedom of Information Act 1982 (Cth) s11(a) PROVIDES
 s11(a) applies ONLY IF
 the document is a document of an agency AND
 the document is not an exempt document

RULE Freedom of Information Act 1982 (Cth) s11(b) PROVIDES
 s11(b) applies ONLY IF
 the document is an official document of a Minister AND
 the document is not an exempt document

So far, the only keywords used have been RULE, PROVIDES, ONLY IF, OR and AND. There are a lot more that can be used, but these are probably the most important ones.

Run, test and edit again - experiment!

Obviously, you do this every time you make a change to the rules. So, you should experiment with changes to rules to see what effect they have (eg what is the difference between IF and ONLY IF?)

AustLII Communities

Home Admin Help

You are here: DataLex » FOIKB [Edit](#) [Attach](#) [Like](#) [New](#) [More](#)

FOI, s11 consultation

📅 30 Aug 2017 - 16:24 | 🗨️ Version 23 | 👤 PhilipChung

[Run consultation](#) | [Check Fact Cross References](#) | [Check Fact Translations](#)

```

THING the document requested
PERSON the person applying
LINK document of an agency TO http://www.austlii.edu.au/au/legis/cth/consol_act/foia1982222/s4.html
LINK official document of a Minister TO http://www.austlii.edu.au/cgi-bin/sinosrch.cgi?query=%22official+document+of+a+minister%22&results=50&submit=Search&mask_world=&mask_path=&callback=on&method=auto&meta=%2Fau

GOAL RULE Freedom of Information Act 1982 (Cth) s11 PROVIDES
  the person applying does have a legally enforceable right under s11 of
  the Freedom of Information Act 1982 to obtain access to the document
  requested ONLY IF
    Freedom of Information Act 1982 s11 (1)(a) applies OR
    Freedom of Information Act 1982 s11 (1)(b) applies
  
```

6.2. Personalise your dialogues - named subjects

If you wish, you can have references to people and things in your rules automatically replaced, when the rules are used to run dialogues, with actual names provided by the user, and correct pronouns and possessives resulting. Subjects which are to be treated in this way are referred to as *named subjects*. The use of named subjects allows you to instantiate the dialogues that DataLex generates, making them appear much more responsive to the answers you have already given.

Use named subjects wherever possible, as they improve communication.

Named subject declarations

Named subjects must be declared in a rule-base, and given one of the types PERSON, THING or PERSONTHING. When an attribute containing a defined subject is first evaluated, automatic prompts for a subject name and, in the case of persons, the subject's sex, will be issued. (An alternative to gendered subjects, based on preferred titles, is being developed.) Where the type is PERSONTHING, the subject may be either a person or a thing (eg where either a natural person or a company may be a subject), and a prompt ('Is x a natural person?') will be issued to determine this.

Named subjects are declared simply by inserting a line or lines anywhere in the rule-base such as in these examples:

```
PERSON the claimant
THING the agreement
PERSONTHING the first party
PERSON the intestate
```

In the FOI example being used, two useful named subjects are:

```
PERSON a person
THING the document
```

Once a named subject is declared, DataLex will recognise it as a named subject in any subsequent part of the rule-base, without need for any further identification of it as such. Named subjects referred to in other attributes are recognised automatically, and their values are substituted in the other attributes.

Run, test and edit yet again

It is important to test any named subjects thoroughly, as they can sometimes affect dialogues in unexpected ways. You may need to make subtle distinctions between, for example 'a person' and 'the person' (with only one of these being a named subject, for the dialogues to work as you intend).

7. Error checking in your rule-base

In addition to the usual 'Run consultation' link to start a consultation with your rule-base, there are on the page of your rule-base, two additional links which allow you to check for some types of errors in your rule-base before you try to run it: 'Check Fact Cross References' and 'Check Fact Translations'.

[Run consultation](#) | [Check Fact Cross References](#) | [Check Fact Translations](#)

7.1. Check Fact Cross References

Use of similarly named but not identically named attributes is one of the main causes of errors in DataLex rule-bases, particularly where rules which are supposed to chain do not do so. The 'Check Fact Cross References' button allows you to check for such errors. This button is very useful because it allows you to check whether the same attribute has been mistakenly described in two similar but not identical ways.

The example given below is for the 'FOI s11' rule-base used in this Tutorial.

Every attribute is printed (in alphabetical order) showing the names of rules setting (*) and using (-) the attribute. In the example below, the attribute 's11(a) applies' appears in two rules:

```
*Freedom of Information Act 1982 (Cth) s11(a)
-Freedom of Information Act 1982 (Cth) s11
```

The rule(s) 'setting' an attribute (marked *) are those where the attribute is a conclusion of the rule. Rules 'using' the attribute (marked ~) are those where it is one of the premises. Named subjects are also listed in alphabetical order (eg 'a person'), but the names of the rules in which they are used are not listed.

Example - FOI s11 rule-base

```
FACT X-REF
=====
```

```
a person
```

```
a person has a legally enforceable right under s11 to obtain access to a
document
```

```
*Freedom of Information Act 1982 (Cth) s11
```

```
s11(a) applies
```

```
*Freedom of Information Act 1982 (Cth) s11(a)
-Freedom of Information Act 1982 (Cth) s11
```

```
s11(b) applies
```

```
*Freedom of Information Act 1982 (Cth) s11(b)
-Freedom of Information Act 1982 (Cth) s11
```

```
the document
```

```
the document is a document of an agency
```

```
-Freedom of Information Act 1982 (Cth) s11(a)
```

```
the document is an exempt document
```

```
-Freedom of Information Act 1982 (Cth) s11(a)
-Freedom of Information Act 1982 (Cth) s11(b)
```

```
the document is an official document of a Minister
```

```
-Freedom of Information Act 1982 (Cth) s11(b)
```

```
the name of a person
```

```
the name of the document
```

```
the sex of a person
```

7.2. Check Fact Translations

Use the 'Check Fact Translations' button to check that your attributes are expressed correctly.

For each attribute in the rule-base, in the order in which they occur, 'Check Fact Translations' shows: (i) prompts (questions); (ii) a translation in positive form; and (iii) a translation in negative form. For example, the interrogative, positive and negative translations of the attribute 's11(a) applies' are as follows:

- Does s11(a) apply ?
- S11(a) applies.
- S11(a) does not apply.

Named subjects are also listed, but since named subjects do not have 'true' or 'false' values, the translations given are only (i) the question to obtain the name; and (ii) the way in which that name would be reported back to the user (where the name give is substituted for '<>').

This allows you to check that DataLex will correctly process the verbs that you have used.

Example - FOI s11 rule-base

DEFAULT FACT TRANSLATIONS =====

- What is a person ?
- A person is <>.

- Has a person a legally enforceable right under s11 to obtain access to a document ?
- A person has a legally enforceable right under s11 to obtain access to a document.
- A person has not a legally enforceable right under s11 to obtain access to a document.

- Does s11(a) apply ?
- S11(a) applies.
- S11(a) does not apply.

- Does s11(b) apply ?
- S11(b) applies.
- S11(b) does not apply.

- What is the document ?
- The document is <>.

- Is the document a document of an agency ?
- The document is a document of an agency.
- The document is not a document of an agency.

- Is the document an exempt document ?
- The document is an exempt document.
- The document is not an exempt document.

- Is the document an official document of a Minister ?
- The document is an official document of a Minister.
- The document is not an official document of a Minister.

- What is the name of a person ?
- The name of a person is <>.

- What is the name of the document ?
- The name of the document is <>.

- What is the sex of a person ?
- The sex of a person is <>.

8. Expanding your rule-base

This exercise aims to guide you to some of the more important parts of the Developer's Manual (The Wysh Developer's Manual can be used until the DataLex Developer's Manual is updated). Use the list below as a check-list of important things to understand about the DataLex software.

Attributes

Understand how to name boolean (true/false) attributes, and the importance of consistency.

Understand how to name a non-boolean attribute type, and how to declare one (if you need to).

Constants

Understand what constants DataLex can recognise automatically - and how they need to be named.

Q&A dialogues

Understand how DataLex generates dialogues, and how you may need to occasionally teach it about a new verb using the VERBS declaration. For extensive examples of the use of VERBS declarations, see the Copyright and Privacy rule-bases in 1. above.

Don't bother with your own attribute translations unless you have to.

Named subjects

This has been covered adequately above, but it is important that named subject be used well and extensively.

Goals

Understand what goal rules are (they appear as choices at the start of a DataLex consultation), and why you need to give them understandable names.

Types of rules

You will usually only need RULE and GOAL rules.

Statements

You need to know how to use IF-THEN, IF-THEN-ELSE and ONLY IF statements, and the differences between them (eg the difference between how to represent types of legal definitions - inclusive and exclusive).

Assertions and assignments (ASSERT and IS statements)(p36) are sometimes useful.

DETERMINE statements should be avoided, as they destroy the declarative nature of the rule-base. Ditto for CALL statements.

Date and number operators

These are only needed if you need to compare dates or amounts, or perform operations on them.

AND, OR and WITH operators

Understand the limits on how AND and OR operate, and how the more complex operators can overcome these limits if needed.

Use the AND/OR operator instead of BEGIN-END operators.

Read the style guidelines!

You will save yourself a lot of problems if you understand these guidelines - their overall message is 'keep it simple and modular'.

9. Hypertext links from rule-bases

Hypertext links may be explicitly created from a knowledge-base, so that whenever a linked word or phrase appears in an inferencing dialogue, explanation or report, it will appear as a hypertext link. Hypertext links to Acts and sections of Acts will be created automatically.

9.1. Automatic links to legislation, cases and articles

Links to names of Acts (and sections within Acts), or citation of cases, or citations of articles will be added automatically to your knowledge-base (ie you do not have to create tags for each link in your knowledge-base), if those Acts/sections/cases/articles are located on AustLII, LawCite or any LII collaborating in CommonLII (BAILII, PacLII, HKLII etc). Automatic links to these cited items will also appear from dialogues generated by the knowledge-base. Automatic links are not created to words defined in Acts.

9.2. Explicit links in a rule-base (the LINK ... TO ... keywords)

In addition to automatic links to AustLII, specific links can be specified in the rule-base. The 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. LINK ...TO ... can be used to create links from a rule-base to any URL, not just to AustLII.

Some useful types of links that can be created are where, in order to answer a question being asked by a DataLex dialog, it is valuable for the user to have access to definitions in Acts, to a case that must be considered, or to authoritative commentary that assists answering the question. In many cases (as above) these links will be created automatically, but if they are not, then LINK...TO may be used.

Example

```
LINK document of an agency TO
http://www.austlii.edu.au/au/legis/cth/consol_act/foia1982222/s4.html
```

```
RULE Freedom of Information Act 1982 (Cth) s11(a) PROVIDES
s11(a) applies ONLY IF
    the document is a document of an agency AND
    the document is not an exempt document
```

(At present, links to Commonwealth legislation can only go to the definitions section (ie s. 4 here), and not to the precise location of the definition. Links to NSW legislation can go to precise definitions, by linking to the anchors to those definitions.)

9.3. Searches from rule-bases

It is also possible to use LINK ...TO ... to create links from a rule-base to a stored search anywhere on the world-wide-web, not just to 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
```

```
RULE Freedom of Information Act 1982 (Cth) s11(b) PROVIDES
s11(b) applies ONLY IF
    the document is an official document of a Minister AND
    the document is not an exempt document
```

10. 'Co-operative inferencing' - rule-bases on multiple pages

[This is being re-developed for the AustLII Communities environment. An example will be added in the next update to this Tutorial.] 'Co-operative inferencing', as we have called it, allows different rule-base developers to place rule-bases on other web pages (outside or inside AustLII Communities) and to declare that consultations (located within the DataLex Community) using their rulebase will also use those other rulebases located elsewhere that they specify are to be 'included'. Rule-base development becomes a 'co-operative' activity where developers can contribute their rule-bases to a larger enterprise.

10.1. The INCLUDE keyword

The use of the keyword INCLUDE in a rulebase, followed by the URL of another page containing a DataLex rulebase, will cause the second rulebase to be loaded with the first rulebase, and the two run together. More than two rulebases can be declared to be INCLUDED. There is no limit on the number.

It is useful to make the URLs of INCLUDED rulebases live links from the starting consultation (using LINK...TO, so that users of a knowledge-base can conveniently view all knowledge-bases which are to be included in a consultation.

It does not matter if an INCLUDED rulebase INCLUDES the rulebase that INCLUDED it - ie DataLex ignores recursion and does not go into some endless loop of loading the same rulebases.

Example

[Example is hypothetical: no 'sandpit.austlii' server exists. See next update for live version.]

```
INCLUDE http://sandpit.austlii.edu.au/~aial/foitest/foi_s11_b.html
INCLUDE http://sandpit.austlii.edu.au/~aial/foitest/foi_s11_a.html
PERSON a person
THING the document
GOAL RULE Freedom of Information Act 1982 (Cth) s11 PROVIDES
a person has a legally enforceable right under s11 to obtain access to a document ONLY IF
    s11(a) applies OR
    s11(b) applies
```

10.2. At least one GOAL rule must be specified

As in the example above, you must specify which rule is the GOAL RULE that is to start the consultation, because the operation of INCLUDE means that you cannot be certain which rule DataLex will consider is the first one appearing in your knowledge-base. If more than one GOAL RULE is specified in a set of 'co-operative' rulebases, the user will be given a choice of which rule is to start the consultation. GOAL RULES may be declared in any rulebase.

10.3. LINK ... TO... and 'co-operative' rulebases

LINK ... TO ... declarations may not yet result in 'document' being hypertext linked from all linked knowledgebases, but only from the one in which it occurs. This is being developed further so that LINK ... TO ... declarations will be 'global' unless you declare them to be local.

11. Document generation

This exercise takes you through constructing simple automated legal document generators using DataLex, and integrating them with rule-bases and hypertext.

11.1. DOCUMENT rules

The DataLex Developer's Manual sets out the special rules relating to document generation (currently covered in the Wysh Developer's Manual, Chapter 6, particularly 6.2 -6.8.) but this short introduction covers key aspects.

Pay particular attention to:

- (i) You need to use both named subject and angle brackets ('< attribute >') in PARAGRAPH, LINE and TEXT statements in order to personalise your documents. A named subject is needed wherever an attribute must have a type - at least where 'PERSON' would be appropriate. (see 6.4)
- (ii) You may need to use BEGIN ... END delimiters if you use a number of consecutive PARAGRAPH statements within an IF ... THEN ... ELSE construct, as otherwise the scope of these keywords may be ambiguous.
- (iii) You need to specify the rule which starts each separate document with 'GAOL DOCUMENT' so that the user can choose which document to generate each time. The system will only generate one document or report each time a consultation is run.

11.2. Using RULES with DOCUMENTS

See a simple example concerning capacity to make a will in the simple will generator example below.

11.3. Hypertext links from documents

Hypertext links may be created from documents and DOCUMENT rules.

11.4. Example - a simple will generator

See <<http://austlii.community/wiki/DataLex/WillGeneratorKB>> for the simple will generator reproduced below.

```

DATE the date of execution of the Will
DATE the date of the old Will
INTEGER the maximum number of months within which the wedding must take place
PERSON the person making the Will
PERSONTHING the sole beneficiary
PERSON the sole executor
PERSON the testator/testatrix's fiancée
PERSON the joint beneficiaries

GOAL DOCUMENT Last Will & Testament PROVIDES
IF the person making the Will is legally capable of making a Will THEN BEGIN
    CALL Disclaimer
    CALL Preamble
    CALL Revocation
    CALL Contemplation of Marriage
    CALL Sole Beneficiary
    CALL Attestation END
ELSE the person making the Will should not make a Will

RULE Capability PROVIDES
the person making the Will is not legally capable of making a Will ONLY IF the person making the
Will is not of sound mind
OR s6 of the Wills, Probate and Administration Act 1898 applies OR the person making the Will is
subject to some other form of incapacity

DOCUMENT Disclaimer PROVIDES
PARAGRAPH Disclaimer: This is not a real Will and must not be used as such.
```

This will does not purport to accurately represent the law of any jurisdictions.

DOCUMENT Preamble PROVIDES

PARAGRAPH This will dated <the date of execution of the Will> is made by me <the person making the Will>, of <the testator/testatrix's address>, <the testator/testatrix's occupation>.

DOCUMENT Revocation PROVIDES

IF all former testatmentary disposition are to be revoked THEN

NUMBERED PARAGRAPH I revoke all former testatmentary dispositions.

ELSE

NUMBERED PARAGRAPH I revoke all former testamentary dispositions except clause(s) <list of clauses from the old will which are to be saved> of my testamentary disposition dated <the date of the old Will> which clause(s) I hereby confirm.

DOCUMENT Contemplation of Marriage PROVIDES

IF this Will is to be made in contemplation of marriage THEN

IF the Will is to be conditional on the marriage actually taking place THEN

IF the person making the Will is domiciled in Western Australia AND the person making the Will does not own immovables in other States THEN

NUMBERED PARAGRAPH This will is made in contemplation of my marriage with <the testator/testatrix's fiancée>.

ELSE

NUMBERED PARAGRAPH This will is made in contemplation of my marriage with <the testator/testatrix's fiancée> and is conditional on the marriage taking place within <the maximum number of months within which the wedding must take place> months.

ELSE IF the testator/testatrix is domiciled in Western Australia THEN

NUMBERED PARAGRAPH This will is made in contemplation of my marriage with <the testator/testatrix's fiancée> but shall not be void if the marriage does not take place.

ELSE

NUMBERED PARAGRAPH This will is made in contemplation of my marriage with <the testator/testatrix's fiancée> but is not conditional on the marriage taking place.

DOCUMENT Sole Beneficiary PROVIDES

IF everything disposed of under the Will is to be left one person THEN BEGIN

IF the sole beneficiary is over 18 THEN

NUMBERED PARAGRAPH I give the whole of my estate to <the sole beneficiary> whom I appoint my sole executor.

ELSE BEGIN

NUMBERED PARAGRAPH I give the whole of my estate to <the sole beneficiary>

NUMBERED PARAGRAPH I appoint the <the sole executor> as my sole executor. END

END

ELSE BEGIN

NUMBERED PARAGRAPH I give the whole of my estate in equal shares to <the joint beneficiaries>

NUMBERED PARAGRAPH I appoint the <the sole executor> as my sole executor. END

DOCUMENT Attestation PROVIDES

PARAGRAPH Signed by the testator in our presence and attested by us in the presence him and each other.

12. Case-based inferencing using PANNDA

This example shows that it is possible to create case-based inferencing applications within the DataLex structure using Alan Tyree's PANNDA ("Precedent Analysis by Nearest Neighbour

Discriminant Analysis”) approach, and to integrate them with rule-bases and hypertext. Case-based reasoning for law is based on the legal doctrine of precedent and that “like cases” should be given a similar treatment. DataLex has limited support for case-based inferencing. When constructing a DataLex knowledge-base in this context, the domain expert first identifies a number of attributes (‘factual variables’ found in each case) and then the database of relevant case law is coded according to the value of these attributes. This example is only included here to illustrate methods of integrating rule-based and case-based representations. Links may be made from EXAMPLE rules to hypertext resources, as with other types of rules.

12.1. Example

The example on the Finder’s Cases is accessible from the DataLex Community page or at <http://austlii.community/wiki/DataLex/FinderKB/> >. It has an extra rule concerning trespassers which is not an Example rule, showing that RULE and EXAMPLE rules can be combined. The first two EXAMPLE rules in FINDER (The Finder’s Cases) knowledge-base are:

```
PERSON the finder
PERSON the non-finder

GOAL RULE the finder wins PROVIDES
DETERMINE the finder wins

RULE trespasser rule PROVIDES
IF the finder is a trespasser THEN the finder does not win

EXAMPLE Armory v Delamirie \[1722\] EWHC KB J94 PROVIDES
  the finder wins ONLY IF
    the finder was not the occupier of the premises AND
    the chattel was not attached AND
    the non-finder was not the owner of the real estate AND
    the non-finder was not the owner of the chattel AND
    there was a bailment of the chattel AND
    there was not a term in a lease which mentioned found items AND
    there was not a master-servant relationship between the parties AND
    the chattel was not hidden AND
    there was not an attempt to find the true owner of the chattel AND
    there was prior knowledge of the existence of the chattel

EXAMPLE Bridges v Hawkesworth \(1851\) 21 LJQB 75 PROVIDES
  the finder wins ONLY IF
    the finder was not the occupier of the premises AND
    the chattel was not attached AND
    the non-finder was the owner of the real estate AND
    the non-finder was not the owner of the chattel AND
    there was a bailment of the chattel AND
    there was not a term in a lease which mentioned found items AND
    there was not a master-servant relationship between the parties AND
    the chattel was not hidden AND
    there was an attempt to find the true owner of the chattel AND
    there was not prior knowledge of the existence of the chattel
```

Note that the rule-base includes automatically inserted links to various UK cases, some on BAILII and others indexed in LawCite. These links are also live when the consultation is run, and the links can be found from the ‘Why?’, Conclusions and Report functions.