WIN-PROLOG 7.0

The contents of this manual describe the product, BDS-PROLOG for Windows (hereinafter called WIN-PROLOG) and one or more of its LPA Toolkits, and are believed correct at the time of going to press. They do not embody a commitment on the part of Logic Programming Associates (LPA), who may from time to time make changes to the specification of the product, in line with their policy of continual improvement. No part of this manual may be reproduced or transmitted in any form, electronic or mechanical, for any purpose without the prior written agreement of LPA.

Copyright (c) 2019 Logic Programming Associates Ltd. All Rights Reserved.

Authors: Rebecca Shalfield, Clive Spenser, Brian D Steel and Alan Westwood

Logic Programming Associates Ltd
PO Box 226
Cranleigh
Surrey
GU6 9DL
England

phone: +44 (0) 20 8871 2016

web site: http://www.lpa.co.uk


01 March, 2019
# Contents

WebFlex Server User Guide

---

Contents

---

Chapter 1 – WebFlex

---

Introduction

---

Chapter 2 – Installing and Setting-Up WebFlex

---

Introduction

---

The WebFlex Directory

---

HTTP Server

---

Copying The WebFlex Directory

---

Copying The ProWeb Files

---

Copying The FLEX.PC File

---

Setting Up The Aliases

---

Viewing The WebFlex Launch Pages

---

Changing WEBFLEX.INI Settings

---

Creating a Temp Directory

---

Running WebFlex

---

Trouble Shooting

---

Chapter 3 – Using WebFlex

---

Introduction

---

INI File Settings

---

[general]

---

[name]

---

Applicable settings

---

General behaviour

---

Example HEAD.HTM, BODY.HTM and FOOT.HTM

---
Chapter 1 – WebFlex

Introduction

WebFlex is an Internet enabled version of the flex expert system toolkit. The LPA web-based expert system, WebFlex, is basically an instance of a ProWeb application.
Chapter 2 – Installing and Setting-Up WebFlex

Introduction

The LPA Setup program will install the WebFlex files. The WebFlex files have been installed if you have a WebFlex directory within your WIN-PROLOG directory.

The rest of this chapter will go through the manual setting-up process.

The WebFlex Directory

The WebFlex directory consists of two subdirectories - PWS and WWWROOT.
HTTP Server

You will need an HTTP server, such as Microsoft's IIS, installed. The rest of this chapter assumes that you are using IIS and that the main directory is C:\INETPUB and the home directory is C:\INETPUB\WWWROOT.

Copying The WebFlex Directory

You need to copy both the PWS and WWWROOT directories to your C:\INETPUB directory. You may already have a WWWROOT directory.

Copying The ProWeb Files

You now need to copy two files into the C:\INETPUB\PWS\EXEC\FLX directory.

Firstly, you need to copy PROWEB.OVL from your C:\PROGRAM FILES\WIN-PROLOG 4420\PROWEB directory and place it into the C:\INETPUB\PWS\EXEC\FLX directory.
Secondly, you need to copy PRO386W.EXE from your C:\PROGRAM FILES\WINPROLOG 4420 directory and place this too into the C:\INETPUB\PWS\EXEC\FLX directory.

Your C:\INETPUB\PWS\EXEC\FLX directory should now look like this:

Thirdly, you need to rename PRO386W.EXE in the C:\INETPUB\PWS\EXEC\FLX directory to WEBFLEX.SYS.

Fourthly, you need to rename PROWEB.OVL in the C:\INETPUB\PWS\EXEC\FLX directory to WEBFLEX.OVL.

Your C:\INETPUB\PWS\EXEC\FLX directory should now look like this:
Copying The FLEX.PC File

You now need to copy FLEX.PC from the C:\PROGRAM FILES\WIN-PROLOG 4420\SYSTEM directory and place it into the C:\INETPUB\PWS\EXEC\FLX directory.

Your C:\INETPUB\PWS\EXEC\FLX directory should now look like this:
Setting Up The Aliases

The files making up WebFlex are split over three distinct areas, each requiring its own alias to be set up within Personal Web Server or IIS.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Alias Name</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\INETPUB\WWWROOT</td>
<td>This is IIS’s ‘home directory’</td>
<td>read</td>
</tr>
<tr>
<td>C:\INETPUB\PWS\EXEC</td>
<td>pws_exec</td>
<td>execute</td>
</tr>
<tr>
<td>C:\INETPUB\PWS\DATA</td>
<td>pws_data</td>
<td>read</td>
</tr>
</tbody>
</table>
Viewing The WebFlex Launch Pages

Go into your web browser and execute the following URL:

http://<computer_name>/wfs_dem.htm

replacing '<computer_name>' with the name of your computer.

WebFlex Demos

WebFlex is LPA’s Expert System for the Internet. Based on ProWeb Server technology, this toolkit enables you easily to deliver expert system applications over the World Wide Web.

We have assembled a small selection of demonstration programs to give you some examples of what can be done with WebFlex; to explore these demos, please select from the following links:

<table>
<thead>
<tr>
<th>Demo</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forms</td>
<td>A demonstration of the use of forms from WebFlex</td>
</tr>
<tr>
<td>Robbie</td>
<td>Robbie helps select items to put into a shopping cart</td>
</tr>
<tr>
<td>Solvents</td>
<td>Select the correct solvent to use when cleaning equipment</td>
</tr>
<tr>
<td>Forest Yield</td>
<td>Recommend a species of tree seed to maximise forest yield</td>
</tr>
<tr>
<td>Insurance</td>
<td>Detecting insurance fraud using Bayes theorem.</td>
</tr>
<tr>
<td>Fuzzy Car</td>
<td>Ranking cars through fuzzy logic.</td>
</tr>
<tr>
<td>Pesticide</td>
<td>An expert for selecting the appropriate pesticide.</td>
</tr>
<tr>
<td>Timetable</td>
<td>Generating time-tables using forward chaining.</td>
</tr>
<tr>
<td>Holidays</td>
<td>An implementation of a decision tree.</td>
</tr>
</tbody>
</table>

Click on the ‘Robbie’ hyperlink:
Before we can click on the ‘Run Demo’ hyperlink, we need to modify WEBFLEX.INI.

**Changing WEBFLEX.INI Settings**

You now need to modify some of WebFlex’s settings within the WEBFLEX.INI file. This file can be found in the C:\INETPUB\PWS\EXEC\FLX directory:

Load WEBFLEX.INI into a text editor (e.g. Notepad) and modify the following settings accordingly:
### Creating a Temp Directory

Two of the WEBFLEX.INI settings above require a TEMP directory to be present in the `C:\INETPUB\PWS\DATA\FLX` directory. Create a new directory called TEMP within the `C:\INETPUB\PWS\DATA\FLX` directory:

<table>
<thead>
<tr>
<th>Original Setting</th>
<th>Change To</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TEMP_PATH=c:\inetpub\pws\data\flx\temp\</code></td>
<td>Leave as supplied but see next section about creating a TEMP directory.</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Creating a Temp Directory

You are now in a position to return to your web browser window and click on the ‘Run Demo’ hyperlink and run WebFlex’s Robbie The Robot example:
## Trouble Shooting

<table>
<thead>
<tr>
<th>HTML Page Displayed</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ProWeb Server</strong>&lt;br&gt;Copyright (c) 1996-2003 Logic Programming Associates Ltd&lt;br&gt;4.31 - 03 Mar 03&lt;br&gt;Main program terminated with an error!&lt;br&gt;Error #31&lt;br&gt;File Not Found&lt;br&gt;ensure_loaded / 1</td>
<td>You have forgotten to copy the FLEX.PC file.</td>
</tr>
<tr>
<td><strong>System Error (OSE:001:0)</strong>&lt;br&gt;Please try again...</td>
<td>You need to create a TEMP directory and point the WEBFLEX.INI settings correctly to it.</td>
</tr>
</tbody>
</table>
Chapter 3 – Using WebFlex

Introduction

WebFlex allows flex programs to run through a browser, to do this it generates HTML forms for the questions it needs to ask. Indeed most flex programs will run unchanged immediately through WebFlex, though they will look rather plain. Unlike previous versions of flex several questions can be asked at once, obviously this is important for a system that is generating HTML pages to send to a browser.

INI File Settings

There are a number of settings contained in the WEBFLEX.INI file to control webflex.

[general]

This section contains settings that are generally applicable.

The path to the flex KSL files can be specified here (it defaults to examples) the path should be named as:

    scripts_directory=path

where path is a relative directory to where WEBFLEX.SYS (PRO386W.EXE) lives.

For example scripts_directory could be set to the directory KSL which is a sub-directory of where WEBFLEX.SYS lives, by the entry:

    scripts_directory=ksl

The KSL files for your application will be picked up from this path.

[name]

A NAME section corresponds to a NAME.KSL file in the examples directory and applies to that. Settings here override settings in the general section.

Applicable settings

    main_goal=my_main_goal  This specifies the main goal to run for a given example; this is not applicable to the general section. All other settings can be in the general section.
    explain_text=Explain  The text for an explain button.
    help_text=Help  The text for an help button.
submit_text=OK The text for an OK button.
question_separator=br The HTML code used to separate questions (could be hr)
explain_image= An image URL to display instead of the explain button.
help_image= An image URL to display instead of the help button
submit_image= An image URL to display instead of the submit button

The following three settings can also be overridden in the style definition of a question group to allow individual pages for each group of questions.

header=testhead.htm The head part of the HTML page
body=test.htm The body part of the HTML page
footer=testfoot.htm The foot part of the HTML page

General behaviour

For each example a header page, a body page and a foot page can be defined together with the output from the flex program up to the question(s) being asked and the form containing the question(s) constitute one page to be sent to the browser.

Example HEAD.HTM, BODY.HTM and FOOT.HTM

HEAD.HTM:

    <HTML>
    <HEAD>
    <TITLE>Webflex example</TITLE>
    <LINK REL="stylesheet" TYPE="text/css" HREF="/white.css">
    </HEAD>

BODY.HTM:

    <BODY>
    <H2>Robbie the Robot</H2>
    <HR>

    WebFlex output will be inserted here.
Questions

The basic user input to flex is through questions, webflex allows flex questions to be grouped together, and for individual questions to have style information associated, and for question groups to have style information associated. The way the styles are associated to questions and groups is through the frame hierarchy by a naming convention. Say you have the question foo, then by defining a frame foo_style you can add slots to specify various things.

The ask mechanism allows for the following use:

    ask groupname,

will ask the questions defined in the group groupname:

    group group_test1
    multi_choice, single_choice_radio, single_choice_listbox,
    single_choice_dropdown.

Ask can also ask several questions (or one!) as:

    ask q1,q2,q3

The difference is that by using a group you can change the appearance of the containing table, through the group style frame:

    frame group_test1_style
    default caption is 'Form 1' and
    default body is 'test1.htm' and
    default footer is 'tstfoot1.htm' and
    default tablestyle is { bgcolor='#FFccFF', border-1 } .

or for a question:

    question single_choice_radio
    A single-choice radio-button question ;
    choose one of choices
    because I need to test a single-choice radio-button question .
frame single_choice_radio_style ;
default method is radio and
default prefill is apples and
default infix is br .

All output from flex up to the ask will be prepended to the form in the page, so for example:

...  
  and write( '<h1>hi there</h1>' )  
  and ask group_test1  
...  

will write out the '<h1>hi there</h1>' before the table containing the group_test1 form, notice that it is written out as HTML so if you want to use characters such as > then you will have to write the HTML equivalent, in this case &gt; as:

    and write( '<h1>hi there 3 &gt; 2</h1>' ).

Also notice that / is doubled up this is because flex uses / as an escape character.

Any output after the last question in a program is executed is sent to the final results page.

**Styles**

The following is a list of styles for various questions and groups. All of the styles are optional and default to various settings.

**Group Styles**

caption      The caption for the containing table.  
columns      The number of columns in the table.  
header       A header HTM file for use with this form.  
body         A body HTM file for use with this form.  
footer       A footer HTM file for use with this form.  
tablestyle   A set that defines the style of the containing table of the form { attribute-value, ... } see your HTML documentation for attributes applicable to a table, (cellpadding etc.).
**Question Styles**

<table>
<thead>
<tr>
<th><strong>multiple choice styles</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>method</strong></td>
</tr>
<tr>
<td><strong>rows</strong></td>
</tr>
<tr>
<td><strong>prefill</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>single choice styles</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>method</strong></td>
</tr>
<tr>
<td><strong>rows</strong></td>
</tr>
<tr>
<td><strong>prefill</strong></td>
</tr>
<tr>
<td><strong>infix</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>input styles</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>prefill</strong></td>
</tr>
<tr>
<td><strong>maximum_length</strong></td>
</tr>
<tr>
<td><strong>cols</strong></td>
</tr>
<tr>
<td><strong>rows</strong></td>
</tr>
<tr>
<td><strong>lower_bound</strong></td>
</tr>
<tr>
<td><strong>upper_bound</strong></td>
</tr>
</tbody>
</table>
Styles can also be passed down by inheritance, for example if you want to apply the style 'method is radio' to several fields you can use the following:

```plaintext
frame radio_style ;
    default method is radio .

frame q1_style is a radio_style .
frame q2_style is a radio_style ;
    prefill is fred .
```

**Because and Help**

The because clause of a question will be associated with a button (or graphic) on the form that will when clicked show a popup dialog with the contained because text.

```plaintext
question single_choice_radio
    A single-choice radio-button question ;
    choose one of choices
    because I need to test a single-choice radio-button question .

question input_set
    A set input field ;
    input set ;
    browse file 'http://www.lpa.co.uk/ln.htm' .
```

**Problems**

There is a clash of operators between ProWeb and Flex. The nested use of the @ operator in ProWeb (e.g. li @ b @ `this is some text`) can not be used with WebFlex.
Chapter 4 – WebFlex Demos

This chapter gives a brief introduction to the supplied WebFlex demos.

Introduction

WebFlex Demos

WebFlex is LPA’s Expert System for the Internet. Based on ProWeb Server technology, this toolkit enables you easily to deliver expert system applications over the World Wide Web.

We have assembled a small selection of demonstration programs to give you some examples of what can be done with WebFlex, to explore these demos, please select from the following links:

<table>
<thead>
<tr>
<th>Demo</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robbie</td>
<td>Robbie helps select items to put into a shopping cart</td>
</tr>
<tr>
<td>Solvents</td>
<td>Select the correct solvent to use when cleaning equipment</td>
</tr>
<tr>
<td>Forest Yield</td>
<td>Recommend a species of tree seed to maximise forest yield</td>
</tr>
<tr>
<td>Insurance</td>
<td>Detecting insurance fraud using Bayes theorem</td>
</tr>
</tbody>
</table>
Robbie

This example demonstrates the use of forward-chaining in a configuration and resource allocation problem, by way of a shopping expert system in which "Robbie the Robot" helps select groceries.

Robbie the Robot

Asking for the initial shopping list

What is on your shopping list today?

- bread
- butter
- coffee
- ice cream

Explain

Submit
Solvents

This example recommends a solvent to be used when cleaning equipment. The recommendation depends on the equipment class, the ventilation of the site, the main material of the equipment and whether the equipment contains rubber compounds.
Forest Yield

This example implements a simple expert system that recommends a species of tree seed, where to get the seed from, the normal yield of the seed and how that yield should be varied.

Forest Yield

Which region is the site located in?
- North Scotland
- East Scotland
- West Scotland

Submit

Back to WebFlex demos

visit the LPA Home Page!
Insurance

This example uses forward-chaining uncertainty rules to implement a probabilistic fraud detection expert system based on Bayesian Networks.

The example also does some hypothetical reasoning and the user is able to go backwards and forwards using the browser, changing answers to questions and getting alternative suggestions.
Fuzzy Car

This example uses Fuzzy Logic to compare car attributes.

Using the age, engine capacity, cost, fuel efficiency, top speed and capacity of a car, this program calculates an overall rating of how suitable the car is for purchasing in comparison to a sample of pre-defined cars.
Pesticide

This example demonstrates the use of user-defined questions within WebFlex.

This enables the presentation of questions to be owned by the developer and supplied, in this case, using ProWeb and Prolog.
Timetable

This example shows a forward-chaining method for creating a timetable.
Holidays

This program is an example of a backward-chaining decision tree in Flex.

The example uses two frames:

- one to facilitate the expert system’s questions and answers
- the other to show the current status of the execution

This second frame is automatically updated whenever a new answer is given. The Flex code has been written in a generic way, and can be easily adapted to other areas.
Forms

This example demonstrates the control WebFlex has over the display of forms using KSL.

The following are shown in this example:

- The asking of several questions on the same page through the use of groups.
- The declaring of attributes for the containing table.
- The control of attributes for individual questions.
- The constraint of numeric input fields to given ranges.
- The control of the position of questions in the containing table.
- The declaration of the containing HTML pages.
- The use of explanations and browse files.
- The declaration of graphics to use with command buttons.
Index

@ operator, 21

Aliases, 10

BASE_URL, 14
Bayesian Networks, 26
because, 21
BODY.HTM, 17

Demo
  Forest Yield, 25
  Forms, 31
  Fuzzy Car, 27
  Holidays, 30
  Insurance, 26
  Pesticide, 28
  Robbie, 23
  Solvents, 24
  Timetable, 29

FLEX.PC, 9
FOOT.HTM, 18
forward-chaining, 26, 29
frames, 30
Fuzzy Logic, 27

Group Styles, 19

HEAD.HTM, 17
HTTP Server, 7
hypothetical reasoning, 26

IIS, 7
INI File Settings, 16
input styles, 20
Launch Page, 12
multiple choice styles, 20
PRO386W.EXE, 8
PROWEB.OVL, 7
PWS Directory, 6
pws_data, 10
pws_exec, 10
Question Styles, 20
Questions, 18
single choice styles, 20
Styles, 19

TEMP_PATH, 14
TEMP_URL, 14
Trouble Shooting, 15
uncertainty rules, 26
user-defined question, 28
WEBFLEX.OVL, 8
WEBFLEX.SYS, 8
WWWROOT Directory, 6