



中央研究院
資訊科學研究所

Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-05-010

Agent Toolbox Version 2 User Manual

Author: Siek Harianto

Reviewer: Chunnan



July 2005 || Technical Report No. TR-IIS-05-010

<http://www.iis.sinica.edu.tw/LIB/TechReport/tr2005/tr05.html>

Agent Toolbox Version 2 User Manual

Author: Siek Harianto

Version: 1.0

Date: June 9, 2005

Reviewers: Chun-Nan Hsu

Current State: Reviewed

(Choices are inactive, started, in review, reviewed, obsolete)

REVISION HISTORY

Revision 0.1	June 9, 2005	In Review
Revision 1.0	July 8, 2005	Reviewed by Chun-Nan Hsu

Copyright © by Academia Sinica, Taiwan and DeepSpot Intelligent Systems Inc.,
Taiwan. Embeded Java browser © 2004 by IceSoft Technologies Inc., USA.
(Agent Toolbox Version 2.0 is an unpublished Beta release and a work in progress.)

Table of Content

Table of Content.....	3
Agent Toolbox Fundamentals	4
Introduction to Agent Toolbox	4
1 What Is Agent Toolbox?.....	4
2 Web Agents and Information Integration.....	5
3 Agent Building Blocks	5
4 Agent Toolbox User Interface	6
5 Agent Toolbox Project	21
Tutorials: Basic	24
1 Tutorial: Creating a single-frame agent	24
2 Tutorial: Creating a multi-frame agent	30
3 Tutorial: Creating a query form agent.....	35
Building Web Agents with Agent Toolbox	38
Building Agent.....	38
1 Creating and managing agent project	38
2 Working with query form.....	42
3 Working with extraction tool	43
4 Using field setting to set agent behavior.....	47
5 Using name-set manager.....	48
6 Conditional branch.....	49
7 Agent testing	50
Tutorials: Building Agents	51
1 Tutorial: Loop	51
2 Tutorial: Conditional branch	52
Executing Web Agents	54
Run agent dialog	54
1 Input tab pane.....	54
2 Output tab pane	55
3 View tab pane.....	55
4 Status tab pane	56

Agent Toolbox Fundamentals

Introduction to Agent Toolbox

1 What Is Agent Toolbox?

Agent Toolbox is a tool for producing *web agents* for automatic data collection from the Internet. Here are some basic features of Agent Toolbox:

- *Web workflow creation*
Using Agent Toolbox, you can create a simple linear workflow or a more complex one with a combination of linear, branch, and loop workflows.
- *Form query recording*
Agent Toolbox remembers which query parameters are used in the workflow.
- *Structured or semi-structured information extraction*
Agent Toolbox is able to extract well structured information like a table or a list, and is also able to extract from less structured documents.
- *Text and binary files collection*
In addition to its ability to do text extraction, the agent can also save text and binary files.
- *Store the collected data in structured format*
The data collected by the agent can either be stored in XML, CSV, or plain text format.
- *Visual design environment*
Agent Toolbox provides a well designed GUI plus a step by step wizard to help the user to easily and rapidly produce a web agent.

As a web data collection tool, Agent Toolbox also supports some well-known Internet protocols and standards such as:

- *HTTP and HTTPS*

- *Cookies*
- *Web Authentication*
- *JavaScript*
- *HTML4*
- *IFrame or Frameset*

2 Web Agents and Information Integration

The Internet contains tons of useful data which are scattered across different web sites. To find them, we have to browse from one site to other sites. Furthermore, to collect data from these various sources, we have to write a different parser for different web pages. These works are error prone and cumbersome. Agent Toolbox helps you building web agents which are capable of doing the same works without any programming. With the help of Agent Toolbox you can create a web agent for each site and integrate the data collected by each agent seamlessly.

3 Agent Building Blocks

A web agent contains three types of building blocks:

- *Node*

A node represents a web page template. Any web pages with the same layout or format belong to the same node. You can also consider a node as a step in your browsing session.
- *Action*

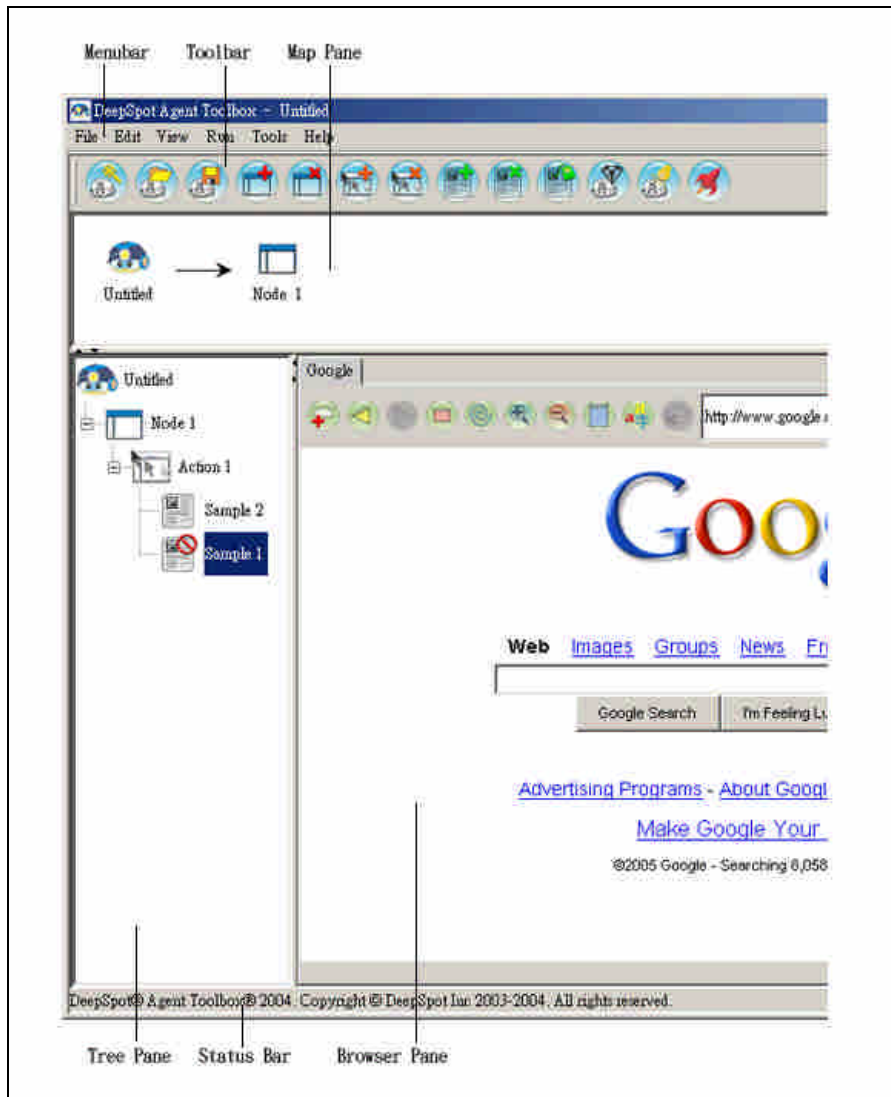
An action represents your desired operation against the node. It might be either submitting a query or doing an extraction. If the action is a query then it means that you want to set some specific form parameters inside the web page and then submit them. If it is an extraction then it means you want to extract some text or binary data from the web page.
- *Sample*

A sample is a realization of a node. In other word, it is a web page that belongs to a specific node. A sample is needed for Agent Toolbox to create an action.

Of these three building blocks, nodes and actions comprise the workflow. A workflow can be considered as a graph that contains nodes connected by

actions. Samples are not a component of the workflow, but it is a kind of “knowledge base” that enables Agent Toolbox to “learn” how to perform the actions accurately.

4 Agent Toolbox User Interface



Menu bar

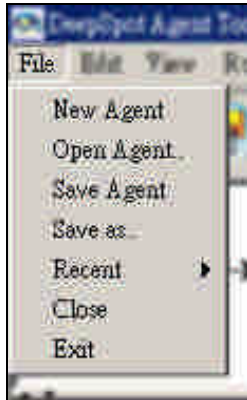
The menu bar is located at the top of the Agent Toolbox UI.



There are six menus available:

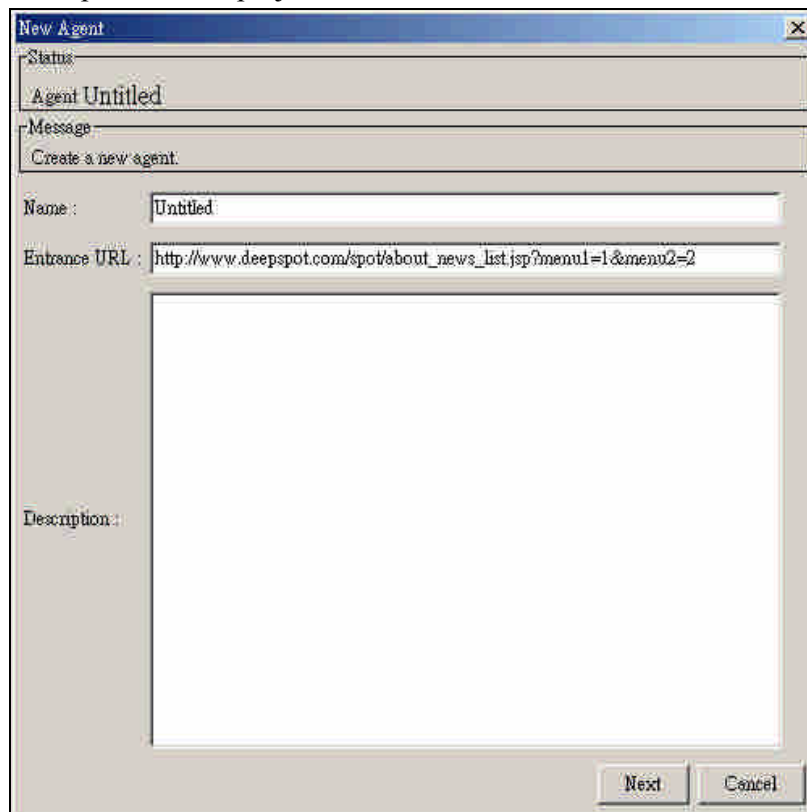
File

The File menu contains commands for creating and opening a project, closing and saving a project, and exiting Agent Toolbox.



File | New Agent

Displays the project wizard, where you set name, entrance URL, and description for the project.



File | Open Agent

Displays the Open dialog box where you browse and choose the name

of existing project to open. Only project files (*.apj) are visible in the Open dialog box. Opening a project displays the project and its building blocks in the tree pane. Accordingly the workflow graph is displayed in the map pane and the entrance URL is loaded in the browser pane.

File | Save Agent

Stores changes made to the current project using the current file name.

File | Save As

Opens a *Save As* dialog box where you can save the current project. You may specify a new name in the File Name field and browse to a new location for the project.

File | Recent

Views a list of agent project files recently opened.

File | Close

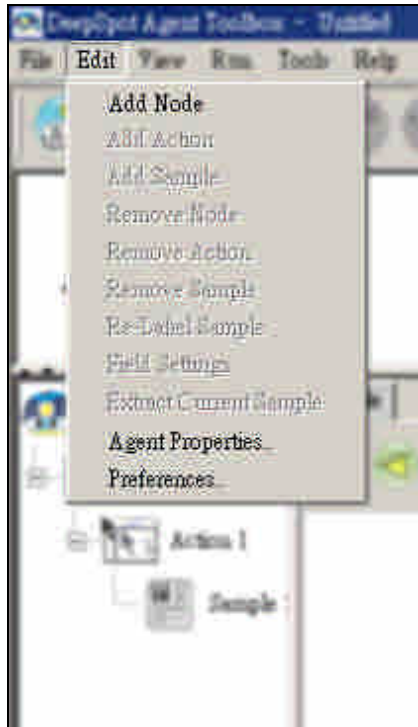
Opens a confirmation dialog where you can choose OK to close the project or Cancel to keep working with the project.

File | Exit

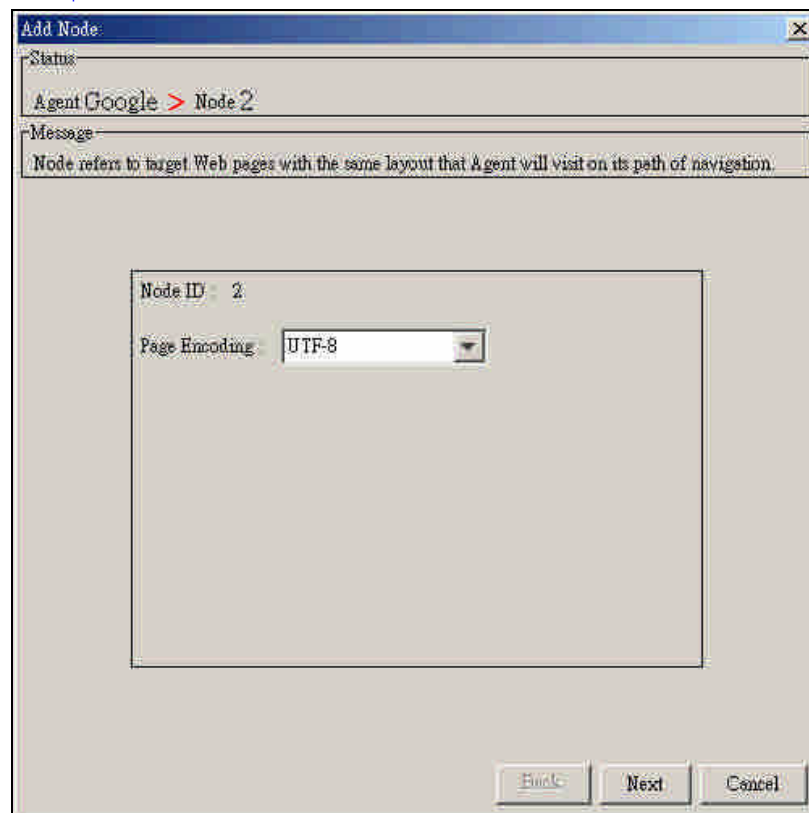
Closes currently opened project and then closes Agent Toolbox. If you exit Agent Toolbox before saving your changes, Agent Toolbox prompts you to save the project.

Edit

The Edit menu contains commands for adding node, action, and sample, removing node, action, and sample, re-labeling a sample, changing field settings, extracting a current sample, changing agent properties, and setting preferences.



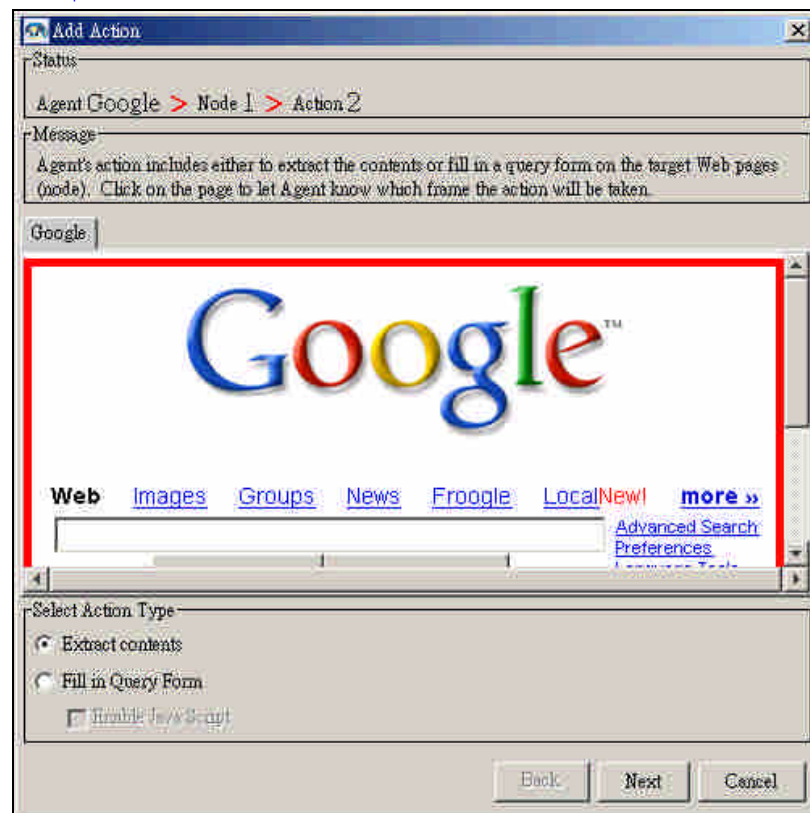
Edit | Add Node



Opens an *Add Node* dialog where you can choose the page encoding of

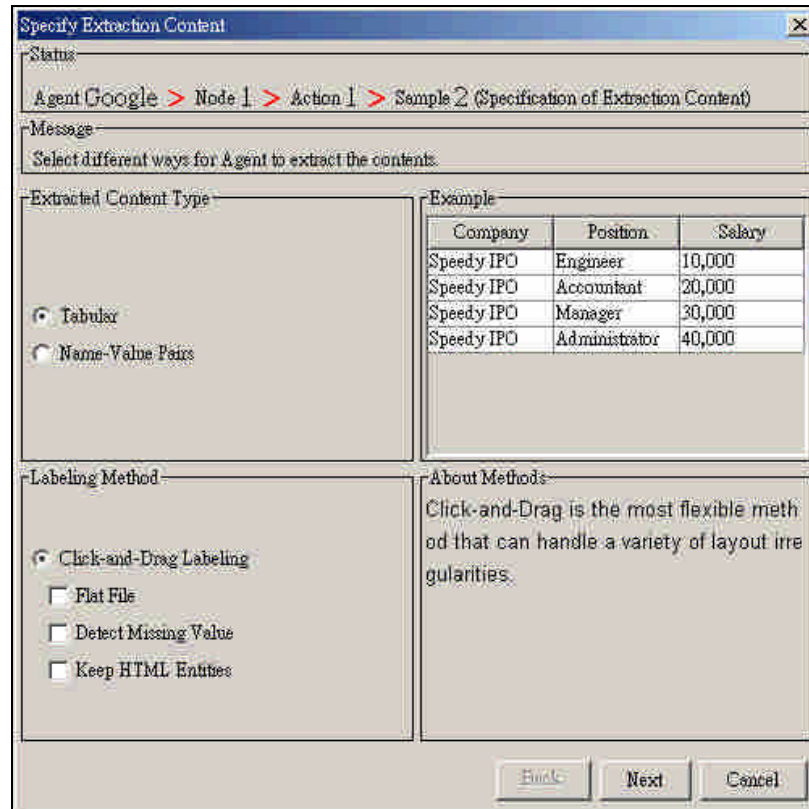
the current web page or use the encoding detected by Agent Toolbox.

[Edit](#) | [Add Action](#)



Opens an *Add Action* dialog where you select a frame at the browser pane (by clicking) and choose an action type for the next step. The frame selection action is not strictly required since not all web pages consist of multi frames. If nothing is selected, the top frame is selected as default, but if a specific frame is selected then a red border will appear around the selected frame. After the frame selection, you have to consider your next action towards the frame/page whether it is “extract content” or to “fill in query form”.

[Edit](#) | [Add Sample](#)



Opens a *Specify Extraction Content* dialog where you choose extracted content type and a labeling method for the next step. We divide content types into tabular and name-value pairs. An example of tabular and name-value content type can be seen at the right side of the *Extract Content Type* pane when you click on the radio buttons. There is only one type of labeling method available now that is the click-and-drag labeling. It has three settings:

- Flat File
If flat file setting is selected then the web page will be rendered as a text file, otherwise it will be rendered as a HTML file.
- Detect Missing Value
If this option is checked, then empty cells inside a table in an HTML page will be replaced by a “N/A” strings.
- Keep HTML Entities.
The default setting will filter HTML entities. If you wish to keep HTML entities (i.e., when you want to extract them as data), then you have to check this setting.

[Edit](#) | [Remove Node](#)

Opens a confirmation dialog asking if you are sure you want to remove the node. Only *end node*, which is the node at the end of the workflow, can be deleted.

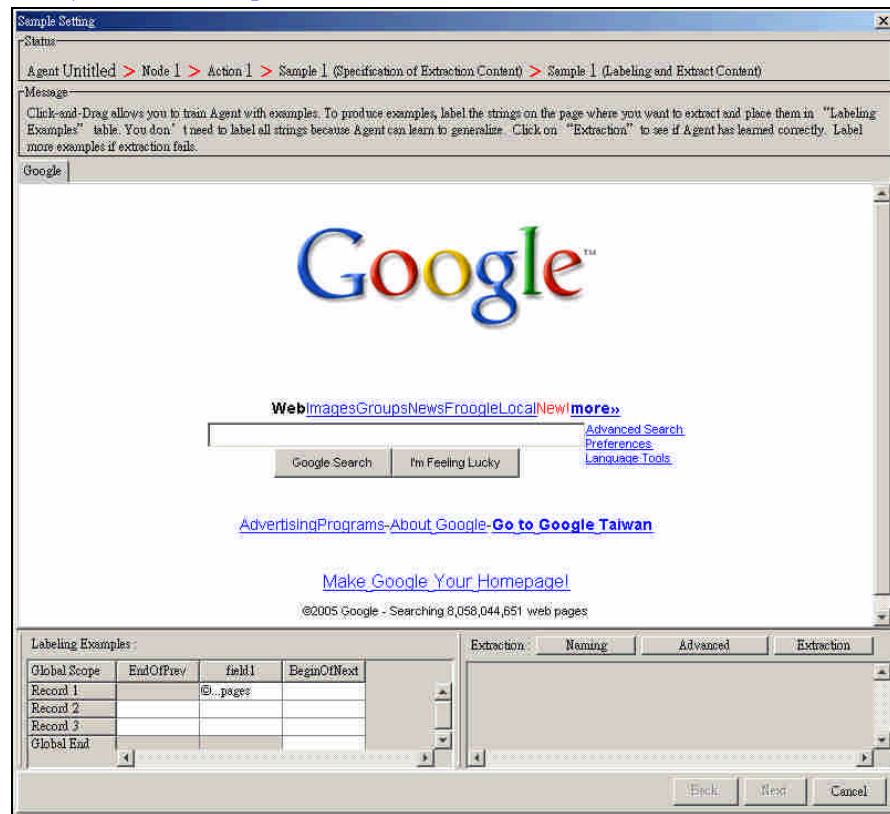
[Edit | Remove Action](#)

Opens a confirmation dialog asking if you are sure you want to remove the action.

[Edit | Remove Sample](#)

Opens a confirmation dialog asking if you are sure you want to remove the sample. The sample has to be in ignored status to make it removable. See 5 Agent Toolbox Project, Removing and deleting, *Removing sample* in this chapter for how to change the status of a sample.

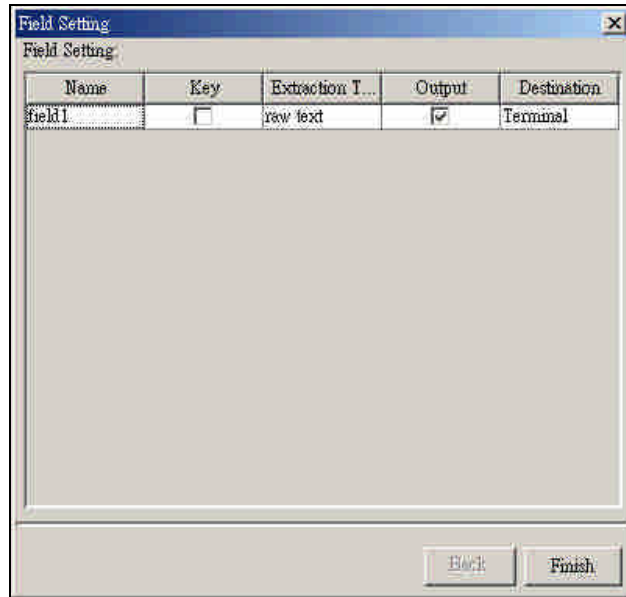
[Edit | Re-Label Sample](#)



Re-labeling means you want to redefine the data, record, and field scope you have defined earlier. For example, if you want to redefine the global end of the data then you click the global end location inside the browser pane and click again at the global end cell of labeling

examples table. See Tutorial chapter for how to create and re-label a sample.

Edit | Field Settings

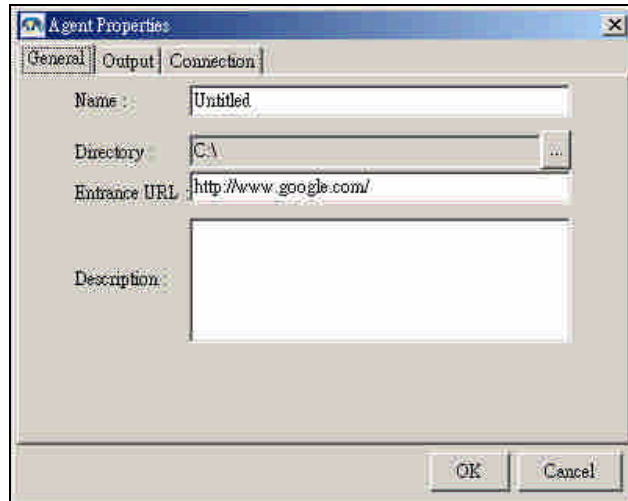


Opens *Field Setting* dialog where you reset “name”, “key”, “extraction type”, “output”, and “destination” so that the agent know how to process the extracted strings from a Web page.

Edit | Extract Current Sample

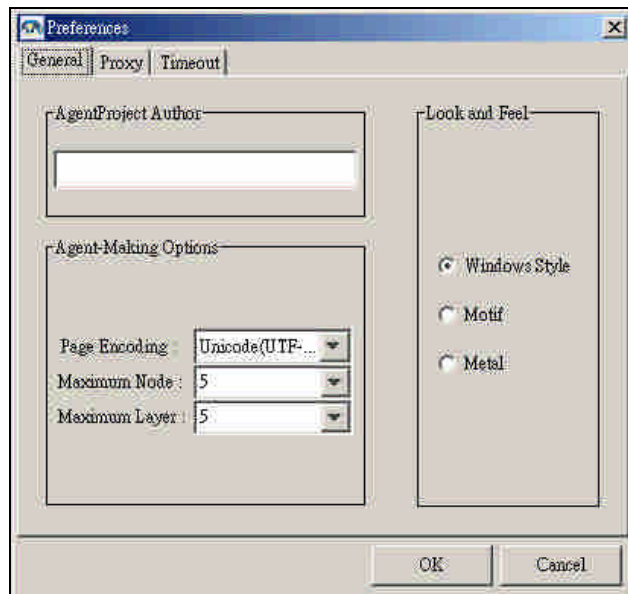
This command opens *Extract Current Sample* dialog showing the extraction result by the current action. It is used to test the accuracy of the extractor.

Edit | Agent Properties



This command opens *Agent Properties* dialog which enables you to change agent's general settings, output settings, and connection settings.

Edit | Preferences



This command opens *Preferences* dialog which enables you to change general settings, proxy settings, timeout policy when you run an agent to access the Web.

View

This View menu contains commands for viewing/hiding status bar, map pane, and tree pane.



View | Status Bar

Checking this item will show the status bar while un-checking it will hide the status bar.

View | Map

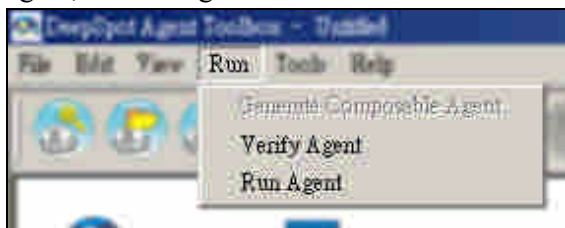
Checking this item will show the map pane while un-checking it will hide the map pane.

View | Tree

Checking this item will show the tree pane while un-checking it will hide the tree pane.

Run

This Run menu contains commands to generate composable agent, verify agent, and run agent.



Run | Generate Composable Agent

Not implemented.

Run | Verify Agent

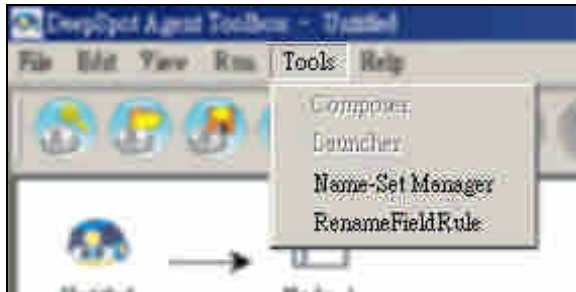
This command opens a dialog showing the status of the agent verification which checks the graph structure of the agent's workflow and other properties to ensure that the agent can run.

Run | Run Agent

Opens Run Agent dialog where you can set agent input, view agent execution process, view agent execution message, and view agent output.

Tools

This tool menu contains commands to launch Composer, Launcher, Name-Set Manager, and Rename Field Rule.



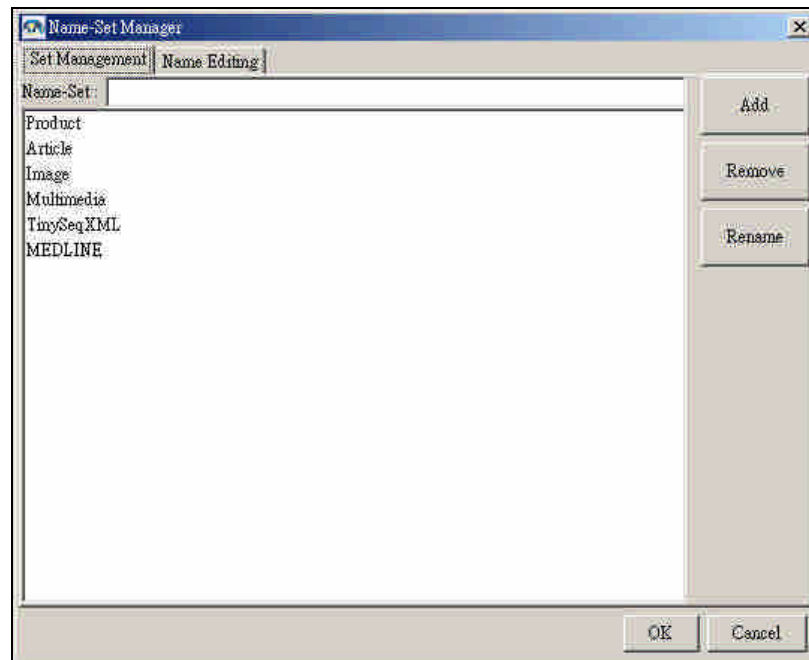
Tools | Composer

Not implemented.

Tools | Launcher

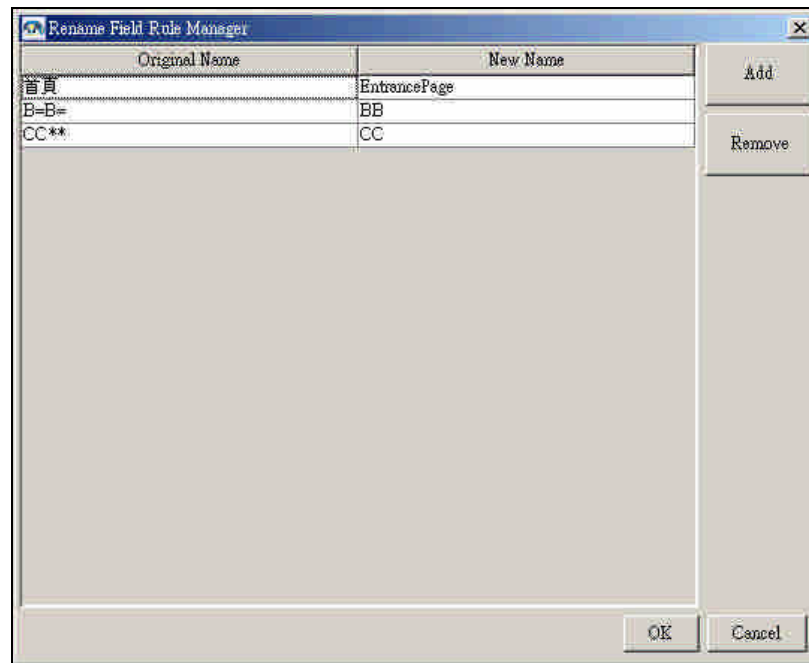
Not implemented

Tools | Name-Set Manager



This command opens *Name-Set Manager* which is a schema management tool. You can add a new schema, remove or rename existing schema, or edit an existing schema.

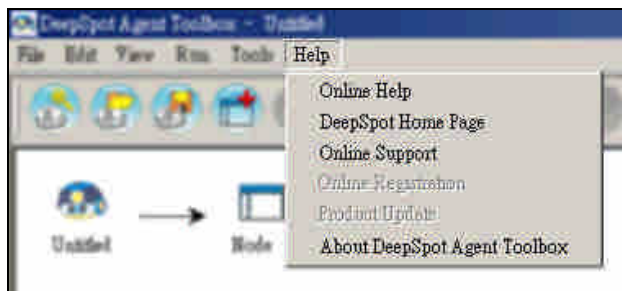
Tools | *RenameFieldRule*



This command opens *Rename Field Rule* which is a field name replacement tool. Using this tool you can define a mapping of old names and new names pair. After the map is well defined, all the old names will be replaced by the new names.

Help

Commands on the Help menu access Agent Toolbox online help, link to the deepspot.com home page and online support, and display the DeepSpot Agent Toolbox About box.



Help | *Online Help*

Displays Agent Toolbox online help in the browser pane.

Help | *DeepSpot Home Page*

Displays deepspot.com home page in the browser pane.

Help | Online Support

Displays Agent Toolbox online support in the browser pane.

Help | Online Registration

Not to be implemented.

Help | Product Update

Not to be implemented.

Help | About DeepSpot Agent Toolbox








Displays the *About DeepSpot Agent Toolbox* dialog box that contains copyright and version information.






Toolbar

The main toolbar is located under the menu bar. It is composed of some frequently used commands from the menu bar.

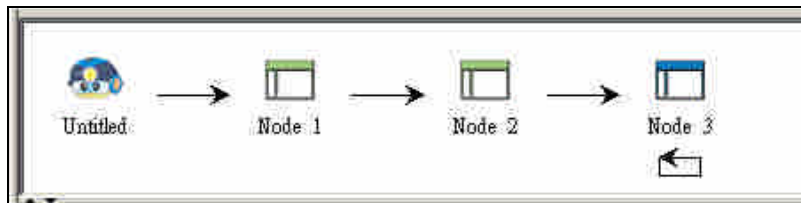


The toolbar provides shortcut buttons for the following menu commands:

Icon	Menu Equivalent
	File New Agent
	File Open Agent
	File Save Agent
	Edit Add Node
	Edit Remove Node
	Edit Add Action
	Edit Remove Action

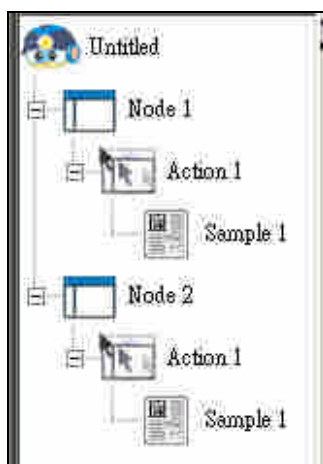
	Edit Add Sample
	Edit Remove Sample
	Edit Re-Label Sample
	Edit Agent Properties
	Edit Preferences
	Run Run Agent

Map pane



This pane provides a read-only view of the workflow in a graphical form. The workflow appears as boxes with arrows between them to represent nodes and control links. It is intended to allow a quick visual overview of the structure of a workflow.

Tree pane



This pane provides a tree view of the nodes, actions, and samples. At the top is the agent project followed by existing nodes. When expanding the node

icon, you can see action(s) and sample(s). The project, node, action, and sample icons will respond to a right click by showing a popup-menu dialog. Each icon has a different popup-menu with the following items:

Icon	Menu Item
Project	Add node
	Load entrance page
	Agent properties
Node	Add action
	Remove node
Action	Add sample
	Remove action
	Field setting
	Extract current sample
Sample	Re-label sample
	Remove sample












Browser pane



This pane is a container for the browser component. At the upper part is the browser tab to show the title of the web page. Below the browser tab are the browser toolbar and the browser process status. Browser toolbar contains

commands for creating a new window, navigating backward and forward, stopping a current loading/rendering process, reloading a current page, zooming in/out, viewing page source, changing page encoding, and closing an existing window. When you open a new Web page, the status icon will be rotating until the page is rendered completely.

The toolbar provides buttons for the following command:

Icon	Command
	Open a new window (as a new tab)
	Go backward
	Go forward
	Stop current loading/rendering process
	Reload current page
	Zoom in
	Zoom out
	View current page source
	Change current page encoding
	Close active window
	Load and render the URL string inside the text field

5 Agent Toolbox Project

Agent Toolbox saves the agent and all related settings in a file that uses a .apj extension. We refer to this file a *project*. The agent is defined by the description of the nodes and actions in a workflow. A project also contains

sample files and maintains the settings and agent properties.

Creating a new agent project

Creating a new agent project in Agent Toolbox consists of steps to create a node, an action and a sample.

1. Choose File | New Agent to open the New Agent dialog
2. Make the following changes to the appropriate fields of New Agent dialog:
 - a. Type an agent name in the Name field. The default agent name is “Untitled”.
 - b. Left the Entrance URL field untouched if you accept it as the entrance URL (the URL of the Web page rendered in the browser pane) or change it into another URL.
 - c. Type some words or sentences in the Description field to describe what this agent does. This field can be left blank.
3. Click Next to go to the Add Node dialog.
4. Accept the default encoding in Add Node dialog.
5. Click Next to go to the Add Action dialog.
6. Accept the default action type in Add Action dialog.
7. Click Next to go to the Extraction Option Selection dialog.
8. Accept all default selection in Extraction Option Selection dialog.
9. Click Next to go to the Sample Setting dialog.
10. Do the following steps to label the sample:
 - a. Add, edit or delete field names in Labeling Examples table
 - b. Highlight token ranges in the sample page and associate them to a specific field of a record, the beginning of the next record, the end of the previous record, or the end of the global range.
 - c. Click Extraction button at Extraction table at the right side of Labeling Examples table.
 - d. Check the extraction result based on the labeling sample.
11. Click Next to go to the Field Setting dialog.
12. Accept all default values in Field Setting dialog.
13. Click Finish and the new project and a new node are created.

Adding nodes to a project

Adding a node to an agent project will also add an action and a sample. The steps for adding a new node is included in the steps for creating a new agent project. See creating a new agent project step 4-13 for details.

Adding actions to a node

Adding an action to a node means adding an action plus a sample. The steps for adding a new action is included in the steps for creating a new agent project. See creating a new agent project step 6-13 for details. It is possible to add more than one action to a node. For example, you might want to extract the content of a Web page (first action) and follow the hyperlink to the next page on the same Web page (second action). In that case, you will need two actions for a node.

Adding samples to an action

The steps for adding a new sample is included in the steps for creating a new agent project. See creating a new agent project step 8-13. It is possible to add more than one sample to an “Extract Contents” action. For example, if you want to extract contents of Web pages generated by a search engine, you might need more than one sample Web pages to cover all possible layout variations so that Agent Toolbox can learn from those samples a correct set of extraction rules.

Removing and deleting

Removing sample

Removing a sample is rather tricky. First you have to set the sample as ignored then you are able to remove it. Here are the detail steps:

1. In the tree pane, select the sample and right click.
2. Select the Re-Label Sample menu item.
3. Click Advanced button in Extraction table to open the Advanced Setting for Extraction dialog.
4. Select sample id you want to ignore from the Samples to learn list and move it to the Ignored Sample list using the “>” button.
5. Repeat step 4 again if you want to ignore another sample id
6. Click OK to save the changes.
7. Click Extraction button to generate the new action produced from the new sample list.
8. Click Next button to go to the next step.
9. Click Finish button to finish the procedure.
10. The node icon is replaced by a new ignored node icon.

There are several possible reasons to remove a sample. One of the most likely reasons is that the sample doesn't help improve the extraction

accuracy for Agent Toolbox.

Removing action

Removing an action will also remove the sample(s) inside it. It is an unrecoverable action so you have to be very careful before doing it. To remove an action, right click on the action icon you want to remove and select Remove Action item.

Removing node

You can only remove the node at the end of a workflow. To remove a node, right click on the node icon you want to remove and select Remove Node item.

Opening an existing project

See File | Open for a detailed description.

Closing a project

See File | Close for a detailed description.

Tutorials: Basic


1 Tutorial: Creating a single-frame agent

For a start, let's create a simple agent from a simple web page. We can begin with the famous Google website and extract the total page number it has been indexing.

Step 1: Create a new project

Type in <http://www.google.com> and press down enter key or click GO button. The browser will load and render the Google main page.



Select File | New Agent in menu bar or click  button on toolbar and then a New Agent dialog will appear. Set the Name field into “google”, let the Entrance URL field untouched and write a description as shown in the figure below. Click Next button to go to the next step.

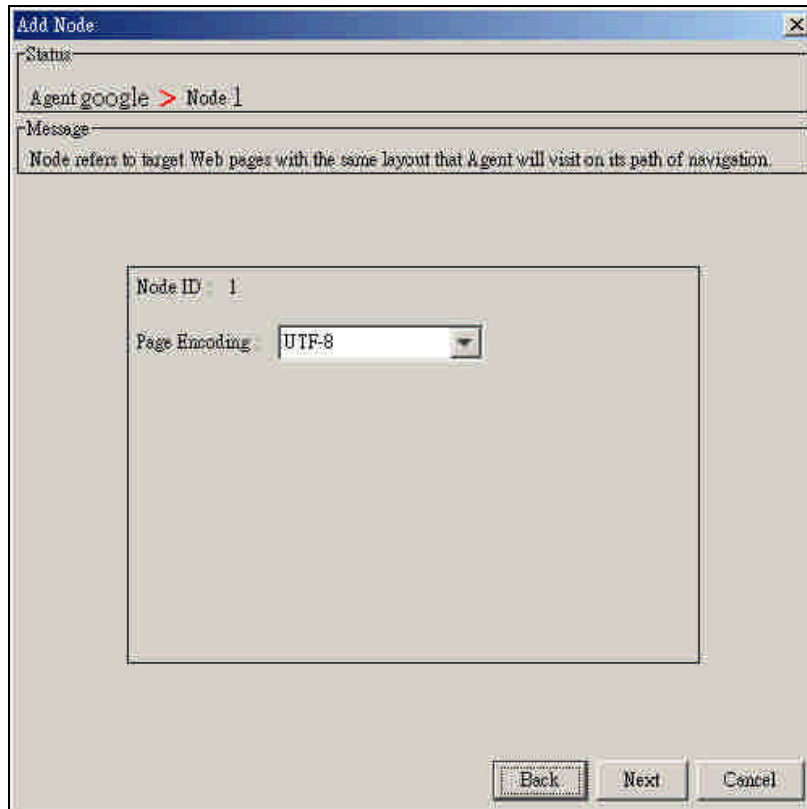
The image shows a 'New Agent' dialog box with the following fields and content:

- Status:** Agent google
- Message:** Create a new agent.
- Name:** google
- Entrance URL:** http://www.google.com/
- Description:** This agent extracts Google's searched web pages number

Buttons: Next, Cancel

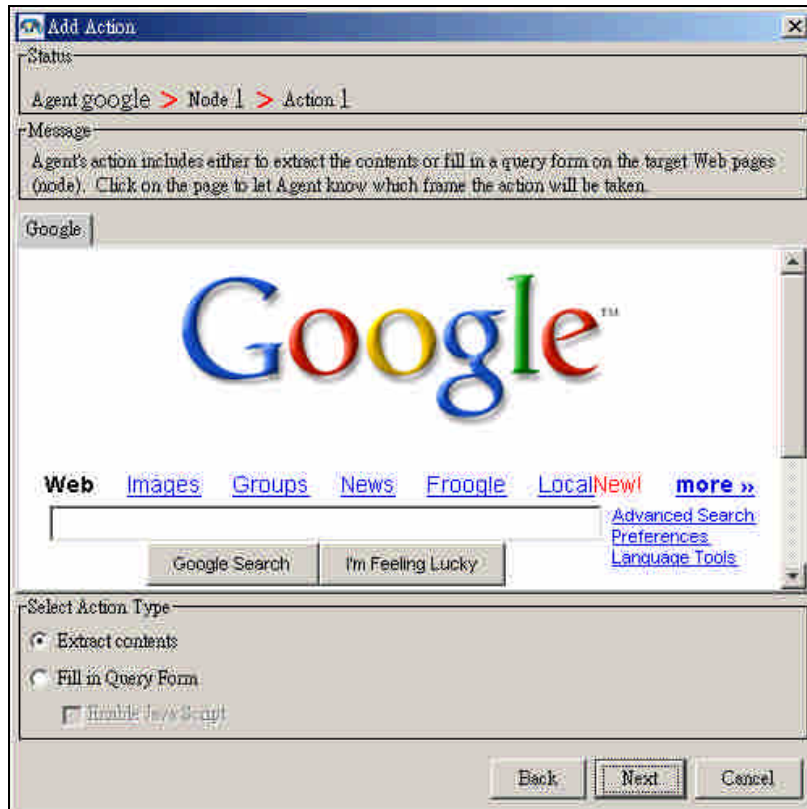
Step 2: Select encoding

Agent toolbox will automatically detect the encoding of the page. Unless the detected encoding is incorrect, you do not need to change the encoding value. Click Next button to go to next step.



Step 3: Select action type

This page is a single frame page so we do not need to make frame selection. Next we want to use the default action type that is Extract Contents option. Click Next button to go to the Extraction Option Selection dialog.



At the Extraction Option Selection dialog, we use all the default settings and click the Next button to go to the next step.

Step 4: Generate an extraction rule from labeling samples

In this step, we will modify the default schema, select the extraction target and do the extraction.

➤ Modify the default schema

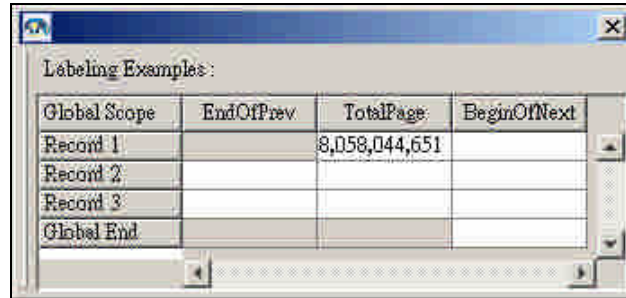
We only need a field with name “TotalPage” so we will remove field2 and rename field1 into TotalPage.

Global Scope	EndOfPrev	TotalPage	BeginOfNext
Record 1			
Record 2			
Record 3			
Global End			

➤ Select the extraction target

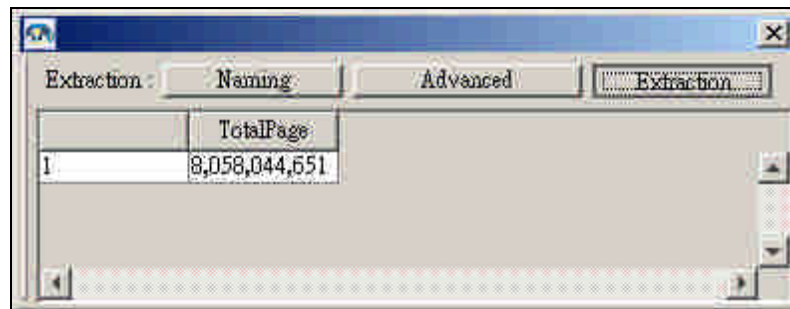
Make a text selection on the web page by clicking at the token we want to extract. The selected token will be highlighted with yellow background. After that click the table cell at location [Record1,

TotalPage]. The selected token will be copied into the cell.



➤ Do the extraction

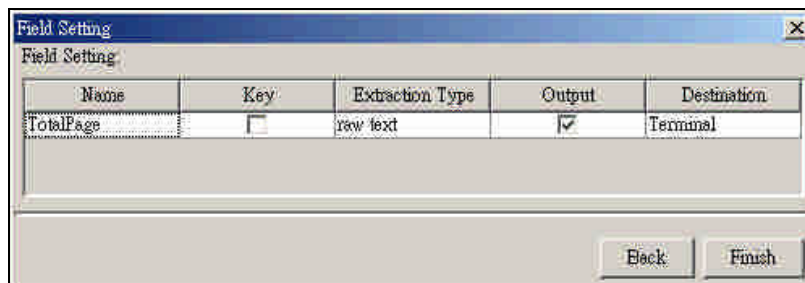
Click the Extraction button and see the result.



Click Next button to go to the next step.

Step 5: Set field attributes

Since we only want the total page text, we will just use the default value of the field setting.



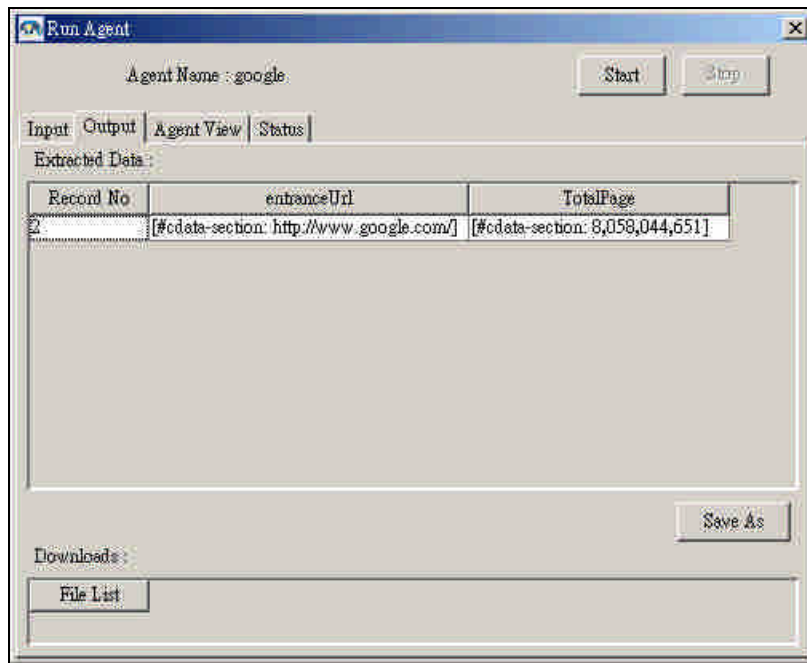
Step 6: Test the agent by executing

After we are finished with creating a new agent, we can test it by executing the agent. We can use Run | Run Agent from the menu bar or



from the toolbar to execute the agent. Calling this command will open a Run Agent dialog starting with Agent View tab being active. Click the Start button to start executing the agent and inside the Agent View tab

we can see the browser automatically navigating the web following the workflow we created. After the execution finished the dialog will switch the view into the Status tab. This tab shows the start time and finish time of the execution. Besides, if there are any errors occurred, the error report will also appear in this tab. To view the execution output we have to switch to the Output tab. The output of the agent is represented in Extracted Data table.



2 Tutorial: Creating a multi-frame agent

This tutorial will be a little bit harder. We will go to Apache Xerces API documentation website: <http://xml.apache.org/xerces2-j/javadocs/api/index.html> which contains multi frames and multi records. This time we will show you an example of how to work on multi-frames and how to extract multiple records.

Packages	
javax.xml.parsers	Provides classes allowing the processing of XML documents.
org.w3c.dom	
org.w3c.dom.css	
org.w3c.dom.events	
org.w3c.dom.html	
org.w3c.dom.ls	
org.w3c.dom.ranges	
org.w3c.dom.stylesheets	
org.w3c.dom.traversal	
org.w3c.dom.views	
org.xml.sax	This package provides the core SAX APIs.
org.xml.sax.ext	This package contains interfaces to optional SAX2 handlers.
org.xml.sax.helpers	This package contains "helper" classes, including support for bootstrapping SAX-based applications.

we want to extract all the package name and description

Step 1: Create a new project

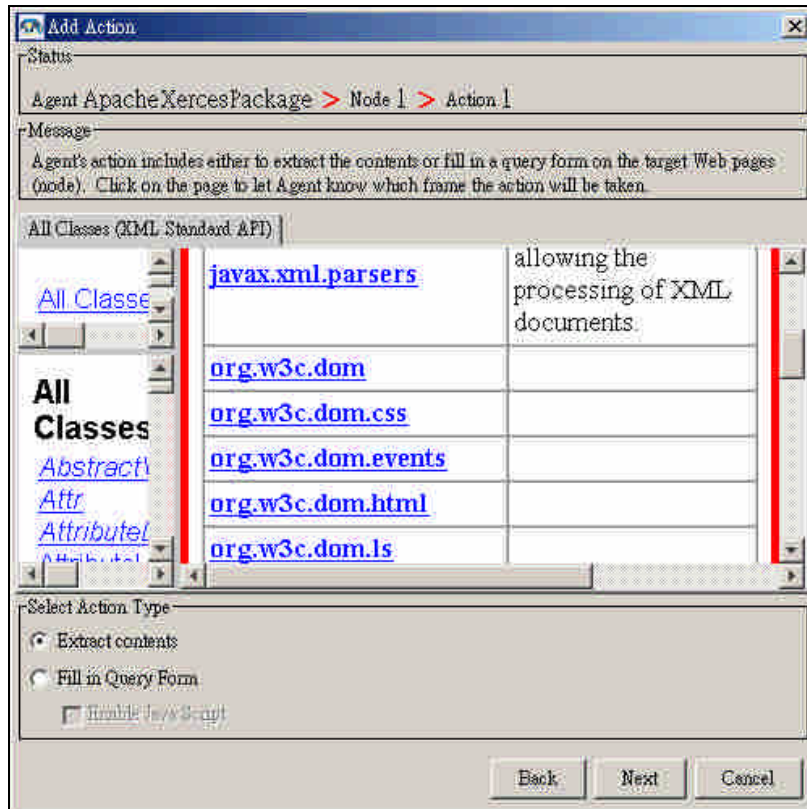
Go to <http://xml.apache.org/xerces2-j/javadocs/api/index.html> and create a new project. Name your project as “ApacheXercesPackage”, use the default value of the entrance URL, and set the description as “This agent is trying to extract apache xerces package names and descriptions”. Go to the next step by clicking Next button.

Step 2: Select encoding

Use the detected page encoding and click Next.

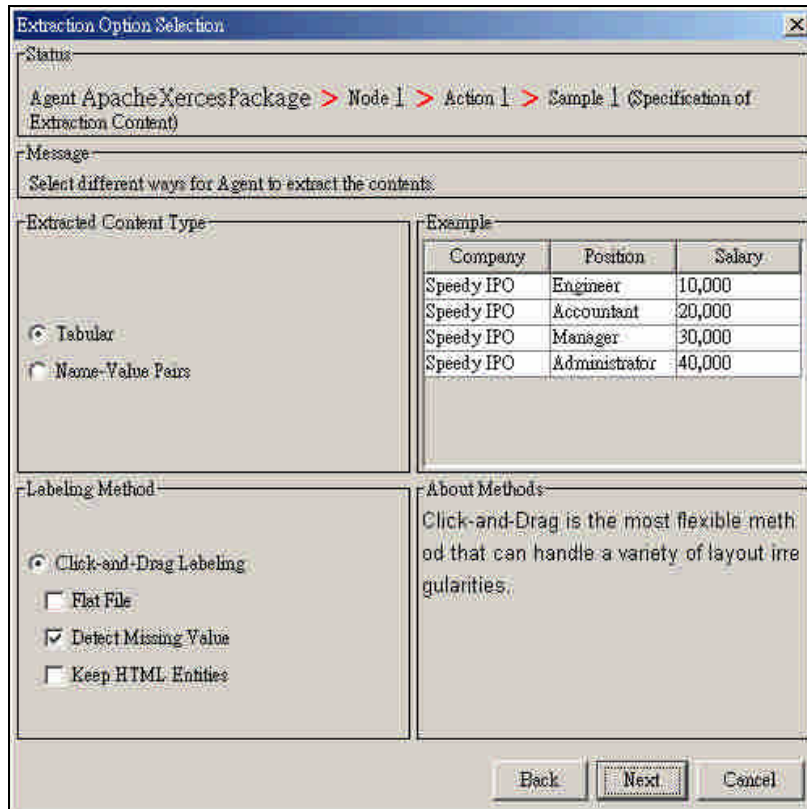
Step 3: Select frame

Make frame selection by clicking at the frame you want to extract. The selected frame will be surrounded by a red border line. Use the Extract contents as the action type and click Next to go to the next step.



Step 4: Set Click-and-Drag Labeling attribute into “Detect Missing Value.”

Looking closely at the API package table, we see that some packages have descriptions but some descriptions are missing. We are aware about this situation very well but the agent’s data extractor is not. We can tell the extractor that some values are missing by selecting the “Detect Missing Value” option below the “Click-and-Drag Labeling”. Selecting “Detect Missing Value” option will tell the extractor to automatically detect missing values and these missing values will be replaced by a “N/A” strings which means “not available”.



Step 5: Generate an extraction rule from labeled samples

Here we have to perform two steps: labeling a sample page and generating an extraction rule. Extraction rule generation step is very simple because the rule will be automatically generated by Agent Toolbox when we click the Extraction button. The quality of the generated rule depends heavily on the labeling step. In the following, we will explain how to label data formatted in a tabular structure. There are four steps you need to do:

- Define the scope of the first record in the table
A table contains several records (rows). You have to start by labeling the **first** record (this is very important). In our example our record contains of two fields which are packageName and description. Labeling the first record means defining which string refers to packageName field and which string refers to description field.
- Define the beginning of next record
After labeling the first record you have to define the beginning of the next record which is “org.w3c.dom”. The definition of the beginning of the next record tells Agent Toolbox that there are more than one record to extract and where the boundary between records may look like.

- Define global end

The third step is defining a global end. The global end definition tells Agent Toolbox where to stop extraction. In our case the global end is the last token of last record.
- Check and label additional records if necessary

After you finished the first three steps you can check by clicking Extraction button. If you are satisfied by the result then the labeling step is finished. But if there are some errors, then you have to label the record in which the error **first** occurs. You can always repeat this step until you get the best result. The worst case is that you have to label all the records in the page to correctly extract all data.

Global Scope	EndOfPrev	packageName	description	BegmOfNext
Record 1		javax.xml.parsers	Provides...	org
Record 2				
Record 3				
Global End				

	packageName	description
1	javax.xml.parsers	Provides classes allowing the processing of
2	org.w3c.dom	NA
3	org.w3c.dom.css	NA
4	org.w3c.dom.events	NA
5	org.w3c.dom.html	NA
6	org.w3c.dom.ls	NA
7	org.w3c.dom.ranges	NA
8	org.w3c.dom.stylesheets	NA
9	org.w3c.dom.traversal	NA
10	org.w3c.dom.views	NA
11	org.xml.sax	This package provides the core SAX APIs
12	org.xml.sax.ext	This package contains interfaces to optional
13	org.xml.sax.helpers	This package contains "helper" classes, incl

Step 6: Set field attributes

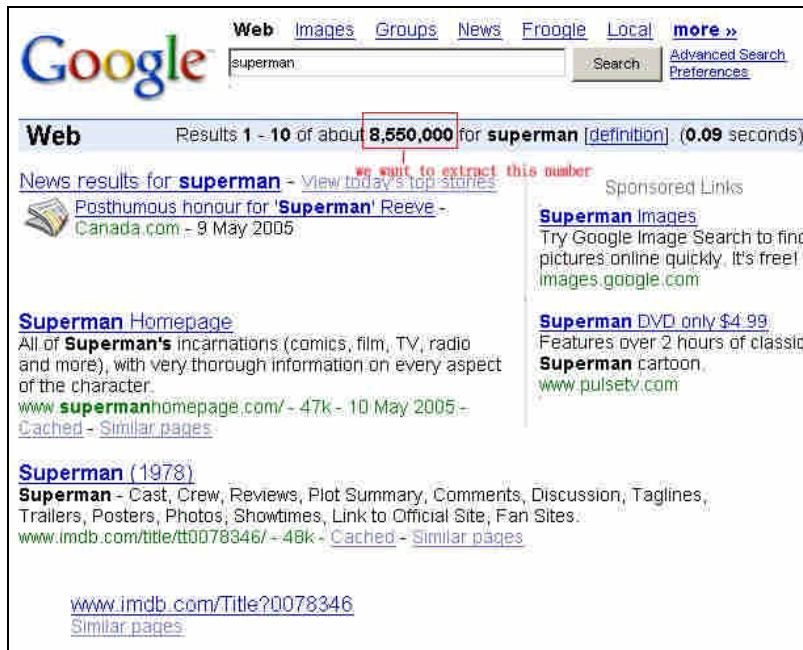
See tutorial 1 step 5.

Step 7: Test the agent by executing

See tutorial 1 step 6.

3 Tutorial: Creating a query form agent

For the last tutorial in this chapter, we will be back again to Google website. We wish to query Google with the keyword “superman” and extract the number of web pages found by Google.



Step 1: Create a new project

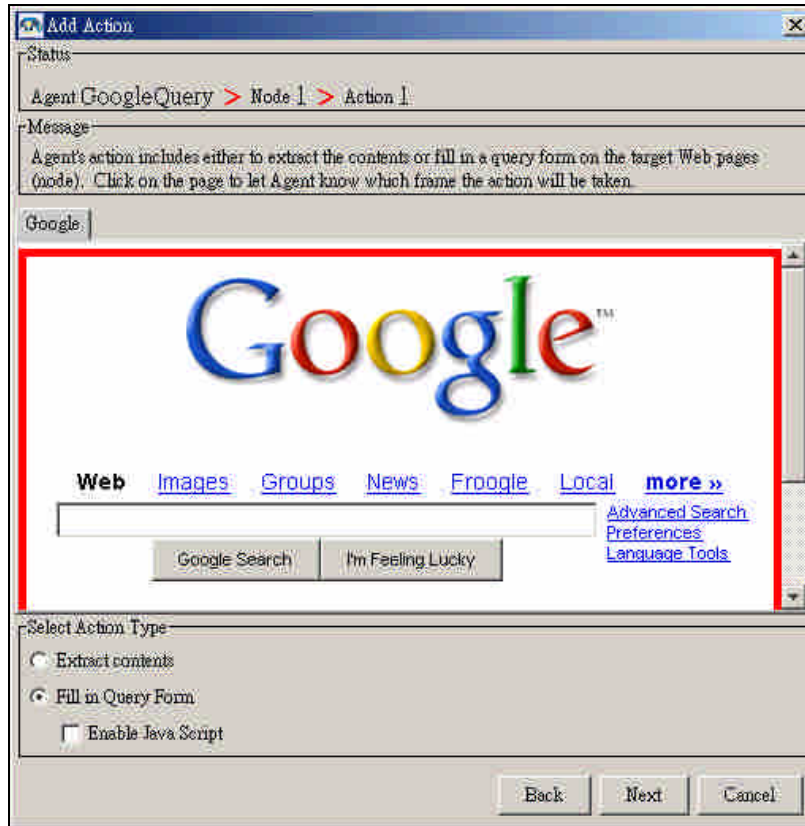
Go to <http://www.google.com> and create a new project. Name your project as “GoogleQuery”, use the default value of the entrance URL, and set the description as “This agent is trying to extract the number of page found based on a query on a specific keyword”. Go to the next step by clicking Next button.

Step 2: Select encoding

Use the detected page encoding and click Next.

Step 3: Select action type

This time we want to perform a form query with keyword “superman”. So at this step we will choose “Fill in Query Form” as our action type. And then click Next to go to the next step.



Step 4: Make a form query

In this step we will perform a form query action. Compared to content extraction, creating a form query action is relatively simple. First type in a keyword or check on radio buttons or do whatever options you want to choose. In our sample we will just type in the keyword “superman” in the text field and left the other options untouched. Then toggle down the “Submission Off” button of the dialog to activate it. After activating the submission-button then you are ready to submit the data. Any button or image you click will be regarded as a submit trigger of the form and Agent Toolbox will collect and remember the form parameter settings. The parameters identified by Agent Toolbox are collected in the table positioned at the lower side of the dialog.

Step 5: Set destination node

Still in the same step, choose a destination node id. In our case the node id is 2. Then click Finish button and a new step is added to the project.

Step 6: Add node 2 for the page number extraction

In the browser pane, which should now render the Google main page with the search form filled in with our keyword “superman”, click the Google Search button which will bring you to the query result page. Add a new

node, select content extraction and label and extract the number of pages found for keyword “superman”. When the second node is added into the workflow, our agent project is done.

Step 7: Test the agent by executing

See tutorial 1 step 6 for detail.

Building Web Agents with Agent Toolbox

Building Agent

1 Creating and managing agent project

Creating a new project

Creating a new project includes setting project name, adding a new node, action and sample. An agent containing a node, an action and a sample is the simplest agent you can create. At the former section we have shown you how to create a new project in detail. In summary, here are the steps that you need to do:

- Set the browser URL location
- Click new agent button on the toolbar
- Set agent name and agent description
- Proceed with add node step
- Proceed with add action step
- Proceed with add sample step
- Finish

Saving project

Save a project

Whenever you are finished working on your project you can save your work using **file | save agent** command. If the project you are working on is a new project then a save file dialog will appear and you can name your project file but if it is an existing project then it is saved with its current name.

Save as another project name

When you are saving your project with **file | save as** command you are

saving it with another project file.

Opening an existing project

Open by exploring file system

Opening an existing agent project by file | open agent command will launch an open file dialog. This dialog enables you to open an agent project file by exploring the file system.

Open from recent history

Selecting file | recent command enables you to view a list of recently opened project files history. We can choose to open a project file from the history list instead of exploring the file system.

Modifying projects

Adding nodes

To add a node to an agent project you have to call the Add Node command. This command can be called from three different locations:

Location	Command
Menu bar	Edit Add Node
Toolbar	Add Node icon
Tree pane	Agent Project icon Add Node

Calling this command will open the Add Node dialog which will show the current node id (un-editable) and the current node encoding (editable).

There are nine encodings currently supported by Agent Toolbox:

- UTF-8
- Chinese Traditional (Big5)
- Chinese Simplified (GB2312)
- Japanese (EUC_JP)
- Japanese (SJIS)
- Korean (EUC_KR)
- Latin (ISO8859-1)
- Arabic (Windows)
- Unicode.

After you finish adding a node, the tree pane and map pane will be updated with an addition of a new node icon.

Adding actions

To add an action to a node you have to call the Add Action command. This command can be called from three different locations:

Location	Command
Menu bar	Edit Add Action
Toolbar	Add Action icon
Tree pane	Node icon Add Action

Calling this command will open the Add Action dialog which will show:

- Current action id.
- Frame selection pane.
In a multi-frame web page you have to select a specific frame as your working space. This selection can be done by clicking the area you want to work on and the selected area will be marked with a red border line.
- Action type option.
There are two types of action which are “extract content” and “fill in query form”.

After you finished adding an action, the tree pane is updated with an addition of a new action icon.

Adding samples

To add a sample to an action you have to call the Add Sample command. This command can be called from three different locations:

Location	Command
Menu bar	Edit Add Sample
Toolbar	Add Sample icon
Tree pane	Action icon Add Sample

Calling this command will open the Add Sample dialog which will show:

- Current sample id
- Extracted content type option
There are two options for extracted content type which are “tabular” and “name-value pairs”.
- Labeling method option

There are three options for labeling method option which are “flat file”, “detect missing value”, and “keep HTML entities”.

After you finished adding a sample, the tree pane is updated with an addition of a new sample icon.

Removing nodes

To remove a node from a project you have to call Remove Node command. This command can be called from three different locations:

Location	Command
Menu bar	Edit Remove Node
Toolbar	Remove Node icon
Tree pane	Node icon Remove Node

You are only permitted to remove the last node of the project. When you are deleting nodes other than the last node an error dialog will appear saying “Only end nodes can be deleted”.

Removing actions

To remove an action from a node you have to call Remove Action command. This command can be called from three different locations:

Location	Command
Menu bar	Edit Remove Action
Toolbar	Remove Action icon
Tree pane	Action icon Remove Action

When you are deleting an action, a confirmation dialog will appear asking if you are sure with the decision.

Removing samples

To remove a sample from an action you have to call Remove Sample command. This command can be called from three different locations:

Location	Command
Menu bar	Edit Remove Sample
Toolbar	Remove Sample icon

Tree pane	Sample icon Remove Sample
-----------	-----------------------------

There are two restrictions on deleting a sample. First, you cannot delete the current sample. Second, you can only delete ignored samples. When you are deleting an ignored sample, a confirmation dialog will appear asking if you are sure with the decision.

Editing field setting

To edit the field setting of an action you have to call Edit Field Setting command. This command can be called from two different locations:

Location	Command
Menu bar	Edit Field Setting
Tree pane	Action icon Field Setting

Calling this command will open the Field Setting dialog which contains a table with four editable attribute. See 4 Using field setting to set agent behavior in this chapter for how to set the values of these attributes.

Name	Description
Key	It is a boolean type attribute with a yes/no value represented as a checkbox.
Extraction type	It is a list type attribute containing six items which controls agent's behavior.
Output	It is a boolean type attribute with a yes/no value represented as a checkbox.
Destination	It is a list type attribute containing currently available node id items and a condition type destination. Note: this attribute is only available when the "extraction type" attribute value is set to "url".

2 Working with query form

In the former section we have discussed about adding an action to a node. We define two types of action "extract contents" and "fill in query form". An action with "fill in query form" type is un-editable which means you cannot edit the field setting or re-label the sample.

Enable/disable javascript

At Add Action dialog there is a checkbox to enable/disable javascript. Checking the option will activate the javascript interpreter when working with the query form. On the contrary, leaving it unchecked will deactivate the javascript interpreter.

Collecting query parameters

The main part of the query form action is the browser pane at the Query Form Parameters Setting dialog. This browser pane is specially designed to listen only to form query events. These events can be keying text(s) into a text field or text area, checking a checkbox, choosing a radio button, or selecting an item in a selection list. Finally, when all the form parameters are set, you want to submit the data. But the submission will not be processed without activating the submission trigger. This trigger is activated by clicking the button labeled “Submission Off” in the right side of the parameter table. It is a toggle button and clicking it will change its status into “Submission On”. Only in this status Agent Toolbox is ready to record the form parameters when you click the submit button.

These buttons are designed to handle the Web pages that use javascript or other client-side computation to process users’ input.

Setting destination

After query form parameters have been set, you have to set the destination. That is, the next node when the form query is submitted to the target Web server.

3 Working with extraction tool

Extracting specific data from a specific page needs a specific parser. The World Wide Web contains lots of web pages with different formats. This means if your data sources come from the Web you have to maintain lots of parser. Agent Toolbox gives you a solution by automatically generating extraction rules (the parser) by example.

Label tool

The examples required for Agent Toolbox to generate extraction rules is provided by the user by labeling the data in the web page. By labeling, we

are talking about associating a block of data to one of pre-defined fields. The labeling is done for a data record after you associate all the pre-defined fields to the appropriate blocks of data. Below is the detail procedure of labeling:

1 Defining schema

A schema is a collection of fields. The default schema for a new sample is [field1, field2]. Usually you want to change the field name into something meaningful. You can do this by renaming the field. Sometimes you need more fields, so you add some new fields. Or maybe you only need a field so you remove a field.

Adding a new field

Here are the steps for adding a new field:

- a. Right click on the head of the table
- b. Select “Add New Field” item from the popup-menu dialog
- c. Set field name in New Field Setting dialog.
- d. Finish

Renaming a field

Here are the steps for renaming a field:

- a. Right click on the head of the table
- b. Select “Edit Field” item from the popup-menu dialog
- c. Edit field name in Edit Field Setting dialog
- d. Finish

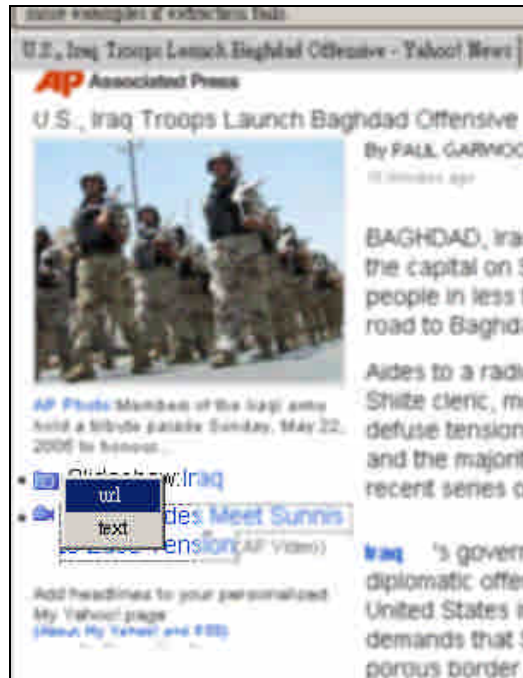
Removing a field

Here are the steps for removing a field:

- a. Right click on the head of the table
- b. Select “Delete Field” item from the popup-menu dialog
- c. Finish

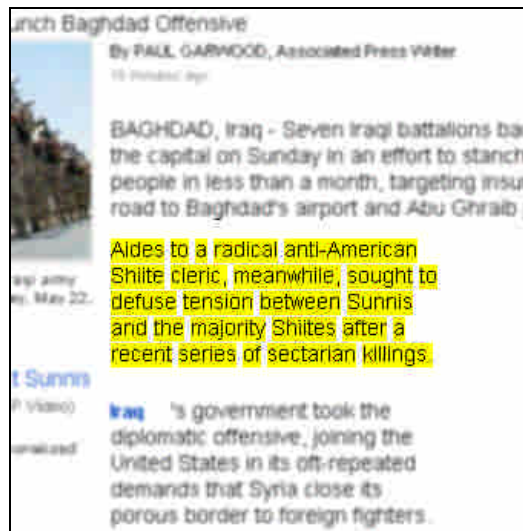
2 Labeling

We label the sample page by highlighting a token or a sequence of tokens and associate the highlighted area to a field from the defined schema. We define two kind of token: url and text. In the sample page (i.e., a HTML file) the URL string is hidden in an anchor tag and is indicated by an underline. When a token is highlighted and there is an anchor in front of it, a popup-menu dialog will appear with two items: “url” and “text”. Selecting the “url” item selects the URL string in front of the text while selecting “text” item selects the text itself.



Highlighting token(s)

To select a token, you click on the token. To select a sequence of token you click on the beginning token and the end token of the sequence. To cancel a selection click on the background of the page.



Labeling a record

Now that the schema has already been defined and you know how to select an area then you are ready to label a record. The idea of adding a sample page and labeling it is to let the extraction rule generator “learns” what a record looks like and which records should be extracted. To label a record, you begin by labeling an area and

associate it to a field from the schema. Repeat this until you associate each field in the schema with different areas in the sample. You associate an area to a field by clicking to a specific table column in Labeling dialog.

Labeling multiple records

You have to do some more steps to label multiple records:

- a. Label the beginning of the next record.
To label the beginning of the next record you label the first field of the next record.
- b. Label the end of the global scope.
To label the end of the global scope you label the last field of the last record.
- c. Test by extracting.
You can test the generated extraction rule by extracting the sample page and check the result.
- d. If necessary, add more sample records.
If the extraction result is not satisfying, try to label another record where the problem occurred.

Extractor tool

Agent Toolbox has a rule learner to produce a set of extraction rules (i.e., a parser) based on the sample pages and the corresponding labeled data. Then an extraction tool will use the produced extraction rules to extract data from web pages that belong to a specific node. When you click the Extraction button Agent Toolbox will extract the sample page using the generated parser. You can examine the extraction result in the Extraction table.

Advanced setting

When you find that adding a new sample makes the parser perform worse you can choose to ignore the new sample. The parser will be re-generated from other samples without using the ignored sample(s).

Docking/undocking labeling and extraction table

The Labeling table and Extraction table can be undocked from the Sample Setting dialog by clicking at the table and dragging it out. This function becomes handy when you want to examine a bulk of sample record or a bulk of extraction result. To dock it back just closes the dialog.

4 Using field setting to set agent behavior

In the labeling step, we associate a specific token sequence to a field. In field setting, we associate a field to a specific set of setting attributes that determines how the agent process the extracted data fields. There are four settings attributes:

- **Key**
When you have more than one content extraction action for one node, you can define some of the fields as keys. The agent will join the extraction results using the keys as in a relational database. If no field is assigned as a key, then the agent will simply append the extracted results.
- **Extraction type**
The extraction type determines how the agent will process the data of a specific field. There are five extraction type options:

Extraction type	Action	Remark
Formatted text	Filter all HTML tags contained inside the text and try to format the text in a nice layout.	-
Raw text	Do nothing	-
Raw text (embedded)	If the text contains links to other binary files, try to save them all.	Links in the saved text will be modified into relative URLs.
Url	Convert into an HTML element and generate a click event on it.	When you set the type into "url," don't forget to set the destination node.
File	Convert into an HTML element, generate a click event on it, and save the linked file.	This option is usually used to save linked multimedia files (e.g., PDF, images, etc.)
Web page (complete)	Convert into an HTML element, generate a click event on it, and save the	Links in the saved Web page will be modified into relative URLs.

	linked file including the files embedded inside it.	
--	---	--

- **Output**
Each node in the agent project produces a set of data. The data from each node are joined together into a large set, constituting data records. This setting enables you to choose which field of the schema will be part of the final output data. Some fields contain links from one node to another might not be useful in the output data. In this case, you can choose to set it not to be in the output. Otherwise, you can check the checkbox and the field will be part of a record.
- **Destination**
The destination field is only editable when you set the extraction type as “url.” It points to a node id.

5 Using name-set manager

Name-set manager is a schema management tool. Using this tool, you can add a new schema, remove an existing schema, or rename an existing schema. You can also export/import a schema into/from the name-set manager. The name-set information is saved in a file “nsmgr.o”.

Adding a new schema

You can add a new schema into the name-set manager in two ways:

- **Calling Tools | Name-Set Manager** command from the menu bar.
Here are the steps to add a new schema:
 - a. Open the name-set manager dialog
 - b. Set a new schema name in the Name-Set text field
 - c. Click the Add button and the new schema name will appear at the bottom of the list
 - d. Select the new schema name from the list
 - e. Switch to the Name Editing tab
 - f. Select the new schema name from the list
 - g. Click Add button and a new blank row will be added to the empty table
 - h. Select the new row and set the field name and data type
 - i. Repeat step h until you set all the field names
- **Calling Naming Window** from the Sample Setting dialog.
Suppose you have defined your schema at the Labeling Examples table

in Sample Setting dialog, here are the steps to add the schema into the Name-Set manager:

- a. Click Naming button to open the Naming Window
- b. Click Save As button to open the Save As dialog
- c. Set the new schema name
- d. Click OK

Removing a schema

To remove an existing schema, open the Name-Set Manager, select the schema-name, and click the Remove button. Afterwards the selected schema will disappear.

Renaming a schema

To rename an existing schema, open the Name-Set Manager, select the schema-name, and click the Rename button. Afterwards a Rename dialog will appear for you to rename the schema.

Mapping name-set schema into the Labeling Examples table

After you added a new schema to the name-set manager, this schema can be used later at the labeling step. In Sample Setting dialog, there is Naming button. Clicking the button will open a Naming Window dialog. In this dialog you can map the current field names to the field names of an existing schema.

6 Conditional branch

At the field setting step, in destination field you can see that beside node id options there is one more option which is condition item. Condition means the destination is not fixed to a node but depends conditionally on the values of some fields. It means that on case A the url will go to node A and on case B it will go to node B, etc. This gives you the flexibility to create a more flexible workflow to handle a more complicated situation.

Syntax

A condition consists of one or more expression connected by a boolean operator. The syntax of a condition looks like this:

expression ((and | or) expression)*

An expression could be a string expression or a numerical expression. A

string expression consists of a field name, a string operator, and a constant string. The syntax of a string expression looks like this:

field (equal | not equal | substring) "string"

A numerical expression consists of a field name, a numerical operator, and a constant number. The syntax of a numerical expression looks like this:

field (= | != | <= | < | >= | >) number

Adding case

To add a new case, choose a field, an operator, and click Add button. A new row is added to the table. It only contains a field name and an operator. And you need to edit it and add a string/number depends on your need.

Removing case

To remove a case, select the case and click Remove button.

7 Agent testing

After you finish creating an agent you have to test it. You test an agent by executing it and as programming chances are you will get some error messages. We distinguish two kinds of errors; extraction error and network error.

When an extraction error occurs in a specific node it means that the extraction rules (parser) of the node is not good enough. Maybe it is because the sample coverage is too small so that it cannot handle a new format. In this case you have to add another new sample to produce a better set of extraction rules (parser). When an extraction error occurs, the agent will save the error page into the error directory. This error page can later be used as a new sample to improve the extraction rules.

A network error might be caused by either server or client network failure or a response timeout. We can use timeout, retry, and interval settings to recover from network errors. Timeout setting is useful for servers with slow response time. Interval and retry settings are useful for servers with heavy server load which frequently responds with server busy response page.

See Executing Web Agent chapter for details.

Tutorials: Building Agents

1 Tutorial: Loop

In this tutorial, we are going to introduce how to create a looping agent. A looping agent is an agent which contains a looping node. An example of a looping node is a page with a “next page” link. This link will take you to another page with different content but similar format. In other words, it takes you to the same node. The trick is to label the first page which contains next page link and the second page which contains previous link and next link. Theoretically, the third page to the n-th page will have the same layout as the second page; this makes the extraction rules work. For the example we will use NCBI site. Our scenario is going to NCBI website (<http://www.ncbi.nlm.nih.gov/>), query for keyword “diabetes” from PubMed database, and navigate all the query result pages. Since the query result exceeds more than one page, our agent will keep clicking the “next page” link until the last page.

Step 1: Create a new project

Go to <http://www.ncbi.nlm.nih.gov/> and create a new project. Name your project as “NCBIQuery”, use the default value of the entrance URL, and set the description as “This agent is trying to query for ‘diabetes’ keyword from PubMed database and view all the result pages by keep clicking next link”. Go to the next step by clicking Next button.

Step 2: Select encoding

Use the detected page encoding and click Next.

Step 3: Select action type

Choose “Fill in Query Form” as the action type and click Next to go to the next step.

Step 4: Make a form query

In this step we will perform a form query action. First, select PubMed from the database selection list and key in keyword “diabetes”. Then toggle down the “Submission Off” button of the dialog to activate it and click the “Go” button to submit.

Step 5: Set destination node

Still in the same step, choose a destination node id. In our case the node

id is 2. Then click Finish button and a new node is added to the project.

Step 6: Add node 2 for the “next” extraction

Click the “Go” button which will bring you to the query result page. Add a new node, select content extraction and label and extract the “next” link. Set the field type as “url” and set the destination to node 2 (loop). The second node is added into the project.

Step 7: Go to the second page and add this page as another sample in node 2

Click “next” to go to the second page and use this page as the second sample.

Step 8: Test the agent by executing it.

2 Tutorial: Conditional branch

In this tutorial we are going to introduce how to create an agent with a conditional branch. A branch is a set of links that comes from the same node and goes to different nodes. A conditional branch is a branch which goes to its destination conditionally, that is it goes to its destination when a condition is established.

For an example we will continue the example in the previous tutorial. Here is the scenario:

We go to the NCBI website and search for “diabetes” articles from PubMed database. We use two actions to extract the query result. The first action is responsible for extracting [url, authors, title, pmid, status]. The second action is responsible for extracting next link. The agent should click the url and extract the abstract of the article only IF the status of the article is “as supplied by publisher”. The next link is always clicked until the last page.

Step 1: Create a new project

Go to <http://www.ncbi.nlm.nih.gov/> and create a new project. Name your project as “NCBIQuery”, use the default value of the entrance URL, and set the description as “This agent is trying to query for ‘diabetes’ keyword from PubMed database and view all the result pages by keep clicking next link”. Go to the next step by clicking Next button.

Step 2: Select encoding

Use the detected page encoding and click Next.

Step 3: Select action type

Choose “Fill in Query Form” as the action type and click Next to go to the next step.

Step 4: Make a form query

In this step we will perform a form query action. First, select PubMed from the database selection list and key in keyword “diabetes”. Then toggle down the “Submission Off” button of the dialog to activate it and click the “Go” button to submit.

Step 5: Set destination node

Still in the same step, choose a destination node id. In our case the node id is 2. Then click Finish button and a new step is added to the project.

Step 6: Add a new node and two actions

The first action is for extracting url, authors, title, pmid, and status from the page and the second action is for extracting the next link.

Set the type of url in action 1 as “url” and set the destination as “condition”. When the condition dialog appears add a new case with this value: **status equals “as supplied by publisher”**

Then set the destination to node 3.

Step 7: Add a new node (node 3) and extract the abstract

This is the last node of the agent. The agent will go to this node if the status of the article is “as supplied by publisher”.

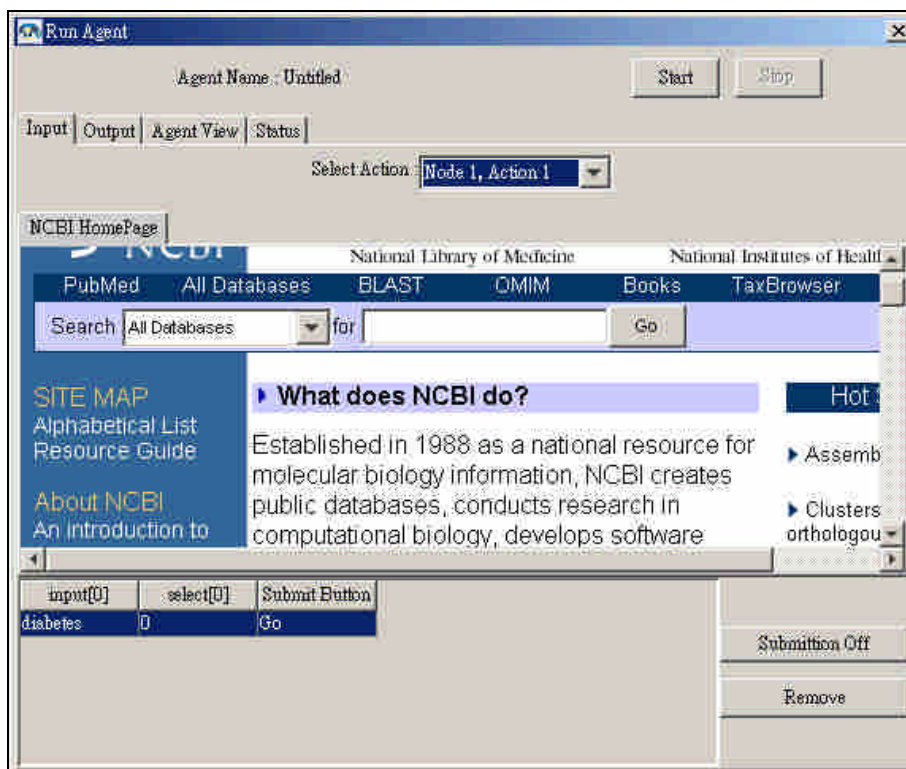
Step 8: Test the agent by executing it.

Executing Web Agents

Run agent dialog

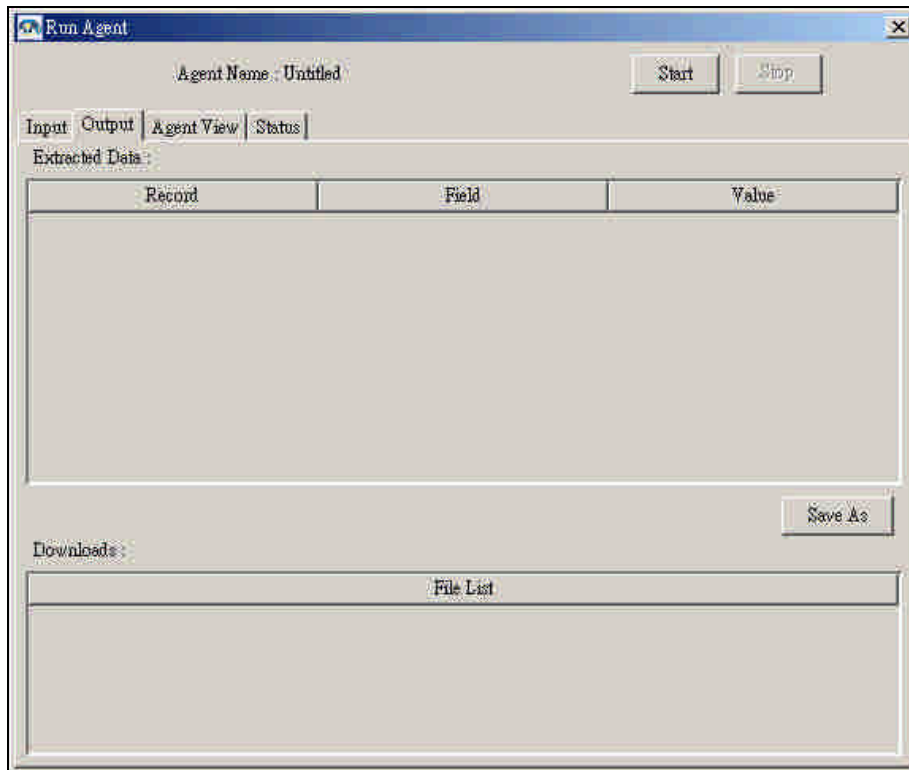
1 Input tab pane

The input tab is positioned as the first tab in Run Agent dialog. It consists of three main areas which are the action selection area, a browser pane, and a parameter list table. This tab is designed to enable you to edit any query form action of any node. The interface to add or remove query parameters is very similar to the interface for adding query form action.



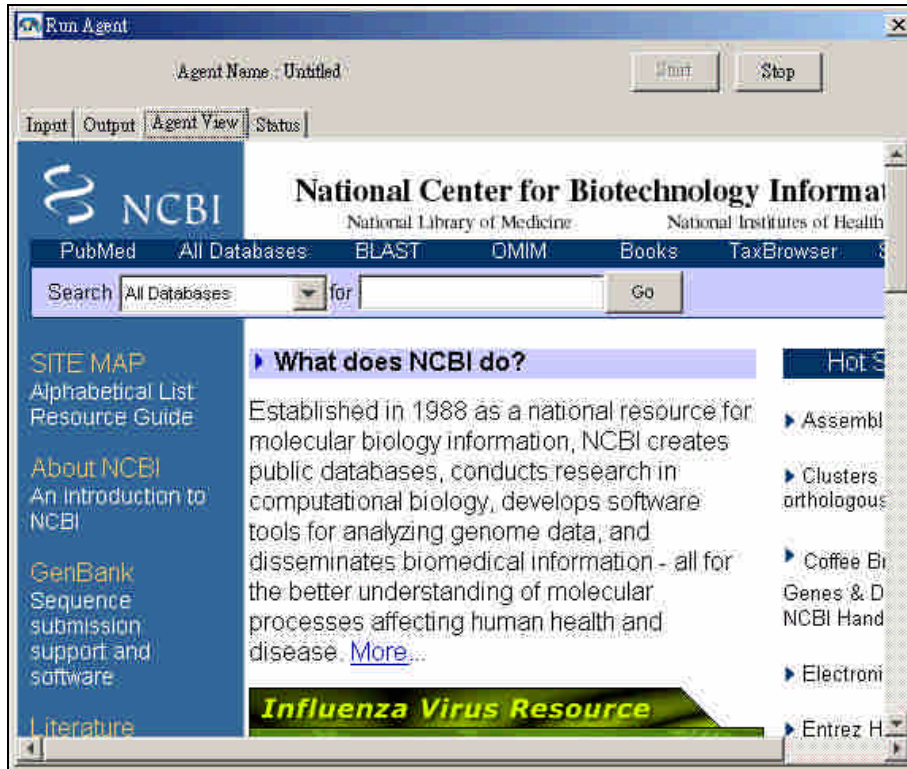
2 Output tab pane

The output tab is positioned at the second position in the Run Agent dialog. It consists of two main areas which are the extracted data area and the download area. Any data in string type will show in Extracted Data area and any data in file type will show in Downloads area.



3 View tab pane

The view tab is positioned at the third position in the Run Agent dialog. It only consists of one area which is a browser pane. The agent execution process will be showed in this pane. We can clearly see how the agent navigating page by page in this pane.



4 Status tab pane

This tab is the last tab of the Run Agent dialog. It contains three main areas which are agent start and end execution time, an execution progress indication bar, and an error report.

