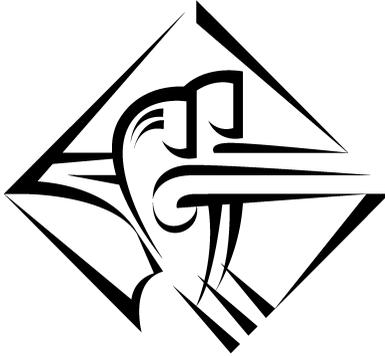


Tango Enterprise 3



Tutorial

Fourth Edition
December, 1998

Pervasive Software
12365 Riata Trace Parkway, Building II
Austin, Texas 78727 USA

Telephone: (512) 231-6000
Toll Free: 1 800 287-4383
Fax: (512) 231-6010

Email: info@pervasive.com
Web: <http://www.pervasive.com/>

PERVASIVE
SOFTWARE

Pervasive Software Inc.
Tango(TM) License Agreement

IMPORTANT: DO NOT INSTALL THE ENCLOSED SOFTWARE UNTIL YOU HAVE READ THIS LICENSE AGREEMENT. BY INSTALLING THE SOFTWARE, OR AUTHORIZING ANY OTHER PERSON TO DO SO, YOU, AND SUCH OTHER PERSON, IF APPLICABLE ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE AGREEMENT, RETURN THE ENTIRE PACKAGE WITHIN TEN DAYS OF RECEIPT.

1 DEFINITIONS.

- a. "Pervasive" means Pervasive Software Inc., a Delaware corporation, 12365 Riata Trace Parkway, Building II, Austin, Texas (U.S.A.) 78727.
- b. "You" or "Your Company" means the person or business entity which is licensing the Software pursuant to this Agreement.
- c. "Software" means all of the software You or Your Company have received from Pervasive with this License.
- d. "Documentation" means the manuals and any other printed material provided by Pervasive with the Software.
- e. "License" means the license purchased and granted pursuant to this Agreement.
- f. "Effective Date" means the date on which the Software was licensed by You or Your Company.

2 LICENSE AND PROTECTION.

2.1 License Grant. Pervasive grants to You or Your Company, subject to the following terms and conditions, a limited nonexclusive, nontransferable, revocable right to use the Software solely for Your internal development and Your internal testing purposes only on a single-user, desk-top device. You are prohibited from deploying any application programs to other parties that You develop using the Software.

2.2 Documentation. Documentation may not be copied. Pervasive reserves all rights not expressly granted to You or Your Company.

2.3 Protection of Software. You and Your Company agree to take all reasonable steps to protect the Software and Documentation from unauthorized copying or use. The Software source code represents and embodies trade secrets of Pervasive and/or its licensors. The source code and embodied trade secrets are not licensed to You or Your Company and any modification or addition thereto, or deletion therefrom is strictly prohibited. You and Your Company agree not to disassemble, decompile, or otherwise reverse engineer the Software in order to discover the source code and/or the trade secrets contained in the source code.

3 COPIES. You may make a single copy of the Software for archival purposes only. All proprietary rights notices must be faithfully reproduced and included on all copies. You may not copy the Documentation.

4 OWNERSHIP. Ownership of, and title to, the Software and Documentation (including any copies) shall be vested solely in Pervasive. Copies are provided to You or Your Company only to allow You or Your Company to exercise Your or Your Company's rights under this Agreement. Only the License is purchased by You or Your Company, as applicable.

5 RESTRICTIONS. Except as expressly authorized in this Agreement, You and Your Company agree not to use, rent, lease, sublicense, distribute, transfer, copy, reproduce, display, modify, create derivative works of, time share or dispose of the Software or Documentation or any part thereof. You or Your Company, as applicable, may use the Software and Documentation solely for your internal business purposes.

6 ASSIGNMENT. You or Your Companies may not assign or otherwise transfer in whole or in part or in any manner any rights, obligations, or any interest in or under this Agreement without

Pervasive Software's prior written consent and any attempted assignment will be void. A merger or other acquisition by a third party will be treated as an assignment. Pervasive may at any time and without Your or Your Company's consent assign all or a portion of its rights and duties under this Agreement to a company or companies wholly owning, owned by, or in common ownership with Pervasive.

7 TERM, TERMINATION. This Agreement is effective from the date You or Your Company open the software envelope and will remain in force until terminated. You may terminate this License at any time by destroying the Documentation and the Software together with all copies and adaptations. This Agreement shall also automatically terminate if You or Your Company breach any of the terms or conditions of this Agreement. You and Your Company agree to destroy the original and all copies of the Software and Documentation, or to return them to Pervasive upon termination of this License. Sections 2.2, 4 and 5 of this Agreement shall survive any termination hereof.

8 LIMITED WARRANTY AND LIMITED LIABILITY.

8.1 Magnetic Media and Documentation. Pervasive warrants that if the magnetic media or Documentation are in a damaged or physically defective condition at the time that the License is purchased and if they are returned to Pervasive within 90 days of purchase, then Pervasive will provide You or Your Company with replacements at no charge.

8.2 Disclaimer of Warranty. PERVASIVE LICENSES THE SOFTWARE PRODUCT TO YOU OR YOUR COMPANY UNDER THIS AGREEMENT SOLELY ON AN "AS IS" BASIS. PERVASIVE MAKES NO OTHER REPRESENTATIONS, CONDITIONS OR WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, REGARDING THE SOFTWARE; PERVASIVE EXPRESSLY STATES AND YOU OR YOUR COMPANY ACKNOWLEDGES THAT PERVASIVE DOES NOT MAKE ANY REPRESENTATIONS, CONDITIONS OR WARRANTIES, INCLUDING, FOR EXAMPLE, WITH RESPECT TO MERCHANTABILITY, TITLE, OR FITNESS FOR ANY PARTICULAR PURPOSE OR ARISING FROM COURSE OF DEALING OR USAGE OF TRADE, AMONG OTHERS.

9 GENERAL CONDITIONS.

9.1 Governing Law. This License Agreement will be governed by, and interpreted in accordance with, the laws of the State of Texas (U.S.A.) exclusive of its choice of law provisions. This Agreement expressly "excludes the United Nations Convention on Contracts for the International Sale of Goods."

9.2 Complete Understanding. This License Agreement sets forth the entire understanding and agreement between You or Your Company and Pervasive and may be amended only in writing signed by both parties. NO VENDOR, DISTRIBUTOR, DEALER, RETAILER, SALES PERSON OR OTHER PERSON IS AUTHORIZED TO MODIFY THIS AGREEMENT OR TO MAKE ANY WARRANTY, REPRESENTATION OR PROMISE WHICH IS DIFFERENT THAN, OR IN ADDITION TO, THIS AGREEMENT ABOUT THE SOFTWARE.

9.3 Waiver. No waiver of any right under this Agreement shall be effective unless in writing, signed by a duly authorized representative of Pervasive. No waiver of any past or present right arising from any breach or failure to perform shall be deemed to be a waiver of any future right arising under this Agreement.

9.4 Severability. If any provision in this Agreement is held invalid or unenforceable, that provision shall be construed, limited, modified or, if necessary, severed, to the extent necessary, to eliminate its invalidity or unenforceability, and the other provisions of this Agreement shall remain unaffected.

9.5 Export Controls. None of the Software or underlying information or technology may be downloaded or otherwise exported or reexported (i) into (or to a national or resident of) Cuba, Iraq, Libya, North Korea, Iran, or any other country to which the U.S. has embargoed goods; or (ii) to any person or entity on the U.S. Treasury Department's list of Specially Designated Nationals, or the U.S. Commerce Department's Table of Denial Orders, or the

U.S. Commerce Department's Entity List of Missile, Nuclear, and Chemical and Biological Weapons Proliferators, or the U.S. Department of State's Foreign Terrorist Organization List. You agree to the foregoing and you represent the warrant that you are not located in, under the control of, or a national or resident of such country or on any such list. The Software may also be subject to U.S. laws and export regulations of the U.S. government that require an explicit export license prior to any export or reexport of the Software. You agree to obtain any such explicit export license that may be required.

9.6 U.S. Government End Users. The Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire the Software with only those rights set forth herein. Contractor/Manufacturer is Pervasive Software Inc., 12365 Riata Trace Parkway, Building II, Austin, Texas (U.S.A.) 78727.

9.7 Consequential Damages. WITHOUT LIMITING THE FOREGOING, IN NO EVENT WILL PERVASIVE BE LIABLE TO YOU OR YOUR COMPANY FOR INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE AGREEMENT, WHETHER UNDER THEORY OF WARRANTY, TORT, PRODUCTS LIABILITY OR OTHERWISE.

9.8 High Risk Activities. The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments regarding fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). Pervasive and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

9.9 English will be the controlling language of this Agreement.

Microsoft Software Disclaimer

NO OTHER WARRANTIES, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, MICROSOFT AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH REGARD TO THE SOFTWARE PRODUCT, AND ANY ACCOMPANYING HARDWARE. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCT, EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

Apple Software Disclaimer

APPLE COMPUTER, INC. ("APPLE") MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT

LIMITATIONS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A SPECIFIC PURPOSE, REGARDING THE APPLE SOFTWARE. APPLE DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE APPLE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE APPLE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL APPLE, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTIONS, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE APPLE SOFTWARE EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. Apple's liability to you for actual damages from any cause whatsoever, and regardless of the form of the action (whether in contract, tort [including negligence], product liability or otherwise), will be limited to \$50.

Copyright Notices

The Tango manual, program design, and design concepts are copyrighted, with all rights being subject to the limitations and restrictions imposed by the copyright laws of the United States of America and Canada. Under the copyright laws, this manual may not be copied, in whole or part, including translation to another language or format, without the express written consent of Pervasive Software Inc.

Copyright © 1999 Pervasive Software Inc. All rights reserved.

Tango Enterprise, the Tango software, Tango Editor, Tango Server, Tango CGI, the documentation, and associated materials are © 1992-1999 Pervasive Software Inc. All rights reserved. Tango is a trademark of Pervasive Software Inc.

FileMaker is a registered trademark and the FileMaker Pro design is a trademark of FileMaker Corporation.

Microsoft, Windows, Windows NT, Windows 95, and the Windows logo are registered trademarks or trademarks of the Microsoft Corporation in the United States and/or other countries.

All other trademarks mentioned are the property of their respective owners.

JavaScript 1.2 Compatible. Portions © Netscape Communications Corporation 1996, All Rights Reserved.

THE LICENSED ELEMENTS ARE LICENSED "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, PERFORMANCE, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY LICENSEE. SHOULD THE SOFTWARE PROVE DEFECTIVE, LICENSEE ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY SHALL NETSCAPE OR ITS SUPPLIERS BE LIABLE TO LICENSEE OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING WITHOUT LIMITATION ANY COMMERCIAL DAMAGES OR LOSSES, EVEN IF NETSCAPE HAS BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES.

Calendar Conversion Code. Copyright 1993–1995, Scott E. Lee, all rights reserved.

Permission granted to use, copy, modify, distribute and sell so long as the above copyright and this permission statement are retained in all copies. THERE IS NO WARRANTY – USE AT YOUR OWN RISK.

Regex. Copyright © 1992 Henry Spencer.

Copyright © 1992, 1993

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Henry Spencer of the University of Toronto.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of this software are copyrighted by INTERSOLV, Inc., 1991–1996.

Table of Contents

	Learning Tango	1
	<i>Things You Need to Know and Do Before You Begin</i>	
	Overview of Lessons	2
	Getting Started	3
	Executing Application Files	4
Tutorial A	Application File Execution Flow and Control Actions	5
	<i>Introducing the Dynamic Web Site</i>	
	Creating a Static HTML Web Page	6
	Generating a Web Page With Tango	8
	Adding a Second Results Action to the Application File	11
	Inserting a Return Action Between the Results Actions	13
	Adding a Branch Action	15
	Adding If, Else If, and Else Actions in the Application File ...	17
Tutorial B	Pass Values on the Web and in Tango	23
	<i>Post Arguments and Search Arguments on the Web and in Tango</i>	
	Passing Data Values Using HTML Forms	24
	Passing Values From One Application File to Another	31
	Using Meta Tags to Display Search Arguments Passed From a Form in Application Files	38
	Using Meta Tags to Display Post Arguments in Application Files	41
	Using Post Arguments to Change Web Page Appearance	44
	Passing Values Within One Application File	48

Tutorial C	Data Sources	57
	<i>Creating a Data Source to Use for the Guestbook and Login Solutions</i>	
	Creating a Tango Tutorial Data Source	58
	Touring the Completed Guestbook Model	62
Tutorial D	Sign My Guestbook	67
	<i>Using the New Record Builder to Build a Sign My Guestbook File</i>	
	Creating a Generic Application File to Insert Records into the Database	68
	Modifying Insert Page Header and Footer for Guestbook Solution	73
	Modifying New Record Response HTML for Guestbook	76
	Setting Required Fields, Field Titles, and Button Titles for Guestbook	79
	Linking “Sign my Guestbook” to the Tango Tutorial Home Page	82
Tutorial E	Tango Builders	85
	<i>Working With the Search Builder and the New Record Builder</i>	
	Creating a Generic Application File to Search Records in the Database	87
	Modifying Search Page Header and Footer for Guestbook	94
	Modifying Search Operators and Field Titles for Guestbook ..	96
	Modifying Search Page Formatting	99
	Modifying Guestbook Search Button Titles	101
	Modifying No Results HTML for Guestbook	103
	Modifying Record List Page Header and Footer for Guestbook	105
	Modifying Record List Page Formatting for Guestbook	108
	Setting the Number of Matching Records Returned	110
	Modifying Record Detail Page Header and Footer for Guestbook	114
	Modifying Record Detail Page Formatting	117
	Adding Search Functionality Using Joins	121

Tutorial F	Guestbook Project	129
	<i>Creating a Tango Project for the Guestbook Solution</i>	
	Creating a Guestbook Project	130
Tutorial G	Guestbook Administration	133
	<i>Allowing Updates and Deletions to the Guestbook</i>	
	Allowing Update of Guest	134
	Updating Hot Links Using the Replace Operation	139
	Allowing Deletion of Guest	142
	Adding a Guestbook Administration Link to the Tutorial Administration Page	146
Tutorial H	Login Solution	147
	<i>Creating a Login Solution With Search Builder and Customizing It</i>	
	Using Search Builder to Create a Framework for the Login Application File	148
	Modifying the Password Field	155
	Setting the Business Logic of the Login Solution	157
	Creating a Personalized Welcome Page	162
	Storing and Tracking User Information	166
	Preventing Unauthorized Access to Application Files in the Options Menu	170
	What's Next	175
	<i>Concluding Remarks</i>	

Learning Tango

Things You Need to Know and Do Before You Begin

Welcome to the Tango Enterprise tutorials. Working through the lessons in these tutorials helps you understand the key functions of Tango Editor. When you are done, you should have a good grasp of the basics required to create Web-based applications with Tango. At that stage, refer to the *Tango Enterprise User's Guide* for complete details on how to use Tango to its fullest potential when creating your Web solutions.

These tutorials are designed for users who are familiar with the basics of HTML and the operation of a Web server.

Some of the common terms you will encounter in these tutorials are briefly described here.

For more information on each term, see the *User's Guide*.

- *Tango Editor* is the development environment featuring a complete graphical user interface in which to develop applications suitable for execution by Tango Server.
- *Tango Server* is an application server that works in conjunction with a Web server (more specifically, an HTTP server).
- *Tango CGI* and *plug-in* both communicate a user's request from the Web server to Tango Server, and return the results to the Web server. The CGI runs as a separate application on your Web server. The plug-in, as the name implies, plugs in directly to your Web server application.
- You create *application files* in Tango Editor. They are like programs or scripts that determine what operations Tango Server performs. These dynamic applications run on your Web server and interact with databases, other applications, and users running Web browsers.
- *Tango builders* help you create and generate a sequence of actions in an application file to perform specific tasks. Tango Editor provides two builders. The *Search Builder* builds the actions required to perform a search of database records, and update and delete them. The *New Record Builder* builds the actions required to add a new record to a database.

- A *Tango data source* contains all the information needed to connect to a particular database. You use data sources to tell your Tango applications which databases to connect to.
- *IP number* (Internet Protocol Number) is a unique number consisting of four parts separated by dots, such as 207.107.95.106. Every machine on the Internet has a unique IP number. Most machines also have one or more domain names that are easier for people to remember. These are more generally used, and are part of the URL.
- *URL* (Uniform Resource Locator) is a standard way to give the address of any resource on the Internet, most commonly used for the Web. An example of a URL is:

`http://www.example.com/`

The most common way to use a HTTP-type URL is to enter it into a Web browser program, such as Netscape Navigator or Microsoft Internet Explorer.

Overview of Lessons

The tutorials are divided into eight topics.

- *Tutorial A* introduces you to the dynamic Web site. You learn about the Tango application file and using control actions.
- *Tutorial B* introduces the concept of passing values. Passed values, such as user inputs, are used to generate dynamic Web sites.
- *Tutorial C* teaches you how to create a data source.
- *Tutorial D* teaches you how to insert records into a database. Using the New Record Builder, it enables you to create a guestbook solution.
- *Tutorial E* teaches you more about Tango builders. You work with the Search Builder and New Record Builder to help you create Tango applications in which you insert new records, search for them, and display the results.
- *Tutorial F* introduces you to the Tango project. Using a guestbook model, you see how a Tango project contains any number of application files and represents a larger organization of your Tango site.
- *Tutorial G* adds more components to your dynamic Web site. You learn how to update or delete a guest from your Guestbook model.

- *Tutorial H* shows you how to create a login application using the Search Builder.

You should go through the lessons sequentially, because they build upon the skills you acquire in each lesson.



The tutorial lessons are not just a series of steps. Explanations of what, how, and why are given for many of the steps to help you build a greater understanding of Tango. You can easily identify these explanations by the ABC's graphic.

Getting Started

Before starting the tutorials, install the Tango Editor software. Install Tango Editor on the Web server or a computer that has a network link to the server. You need to save the files you create in the `TangoTutorial` folder within the `Tango3` folder that is on your Web server.

See the *Tango Enterprise Getting Started Guide* for installation instructions.

Tango Server must be running on your Web server machine. Before you begin the tutorials, there are some items you need to identify. The tutorial refers to these items in general, so you must keep these items in mind as you proceed through the lessons.

- *What Web server are you using?* If you are on Windows NT, you are using either Netscape Server (NS) or Microsoft Internet Information Server (IIS). If you are on Windows 95, you are using NS; IIS is not available for Windows 95.
- The following are the corresponding Web server document root directories or folders for the servers.

MS Internet Information Server

`C:\InetPub\wwwroot\`

Netscape Server

NT `C:\Netscape\Server\docs\`

95 `C:\Program Files\Netscape\Server\docs\`

The above directories are where all documents or files that you want served on the Web must be located. When you call the IP address or domain name of the server machine in the URL line in your browser, it represents the directory or folder specified, and looks there for files to serve up.

- The following are the directories or folders where the Tango CGI and the Tango plug-in are installed on their corresponding Web servers.

MS Internet Information Server

C:\InetPub\scripts\

Netscape Server

NT C:\Netscape\SuiteSpot\cgi-bin\

95 C:\Program Files\Netscape\Server\cgi-bin\

- Ensure you have a browser installed on your machine. You should use Microsoft Internet Explorer or Netscape Navigator.
- In the lessons, you occasionally need to use a text editor, such as Notepad, to edit HTML (Hypertext Markup Language) files or pages.
- The database to be used for this tutorial is provided for you. You will be using a Pervasive.SQL database.

Executing Application Files

In a tutorial lesson, when you are asked to execute a particular file, you must type certain information in the URL line of your browser. The information is slightly different depending on whether you are using the Tango CGI or plug-in.

To execute an application file, type the following in the URL line, replacing each italicized part with the actual information:

- If you are using the Tango plug-in (for both Microsoft and Netscape servers):

`http://your.web.server/path to application file/yourfile.taf`

- If you are using the Tango CGI:

- With Microsoft Internet Information Server

`http://your.web.server/scripts/t3cgi.exe/path to application file/yourfile.taf`

- With Netscape Server

`http://your.web.server/cgi-bin/t3cgi.exe/path to application file/yourfile.taf`

Application File Execution Flow and Control Actions

Introducing the Dynamic Web Site

There is a profound difference between a static Web site and a Tango Web site.

A static Web site is *fixed* in what it can show you. It is a series of pages linked together through references in HTML.

A Tango Web site is fluid, dynamic, and capable of displaying any number of Web pages. It simply depends on what a Tango site is *asked* to show. This series of exercises introduces you to the dynamic Web site.

The following lessons make up Tutorial A:

- creating a static HTML Web page
- generating a Web page with Tango
- adding a second HTML Results action to a single application file
- inserting a Return action between Results actions
- adding a Branch action to an application file
- placing the Branch action within an If action
- adding the If, Else If, and Else actions to the application file.

LESSON A - 1

Creating a Static HTML Web Page

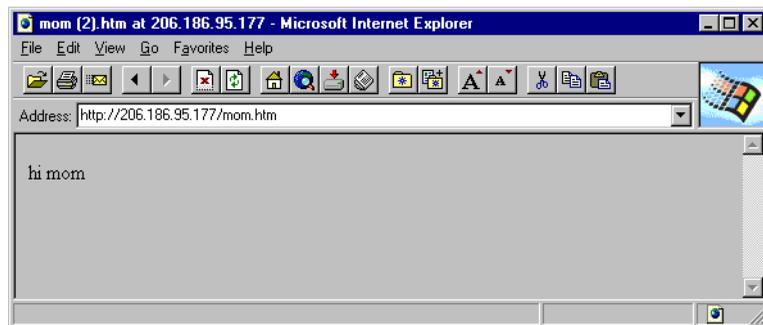
Purpose To create a simple static HTML Web page.

Result A Web page served up by your Web server.

- Exercise**
- 1 Open a new file in Notepad or similar text editor.
 - 2 Type “hi mom”.
 - 3 Save the file as `mom.htm` in your Web server root directory.
 - 4 Open your browser, and in the URL line, type your Web server address (or IP number) followed by the name of the HTML file you just created.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

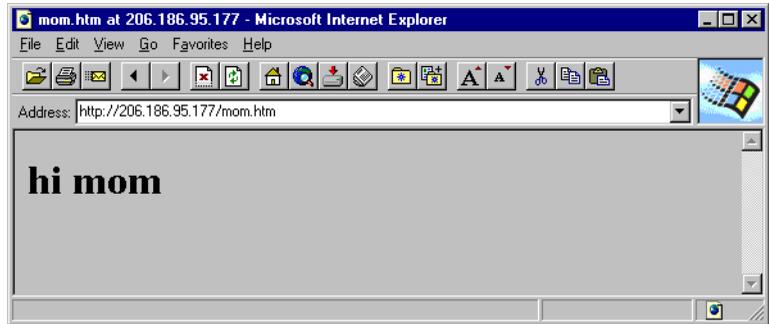
Your message appears in plain text.



This is how easy it is to create a Web page.

- 5 Open the `mom.htm` file in your text editor again.
Now use HTML tags to change the format of your message.
- 6 Place `<H1>` (opening tag for a heading) before your message, and `</H1>` (closing tag) after your message.
This enlarges “hi mom” to the largest heading size possible in HTML code.
- 7 Save and replace `mom.htm`.

8 Return to your browser and reload mom.htm.



Q. Do you see the change in the size of the text?

LESSON A - 2

Generating a Web Page With Tango

Purpose

To create an application file with Tango Editor, which generates HTML in a browser with Tango Server and the Web server working together.

Context

With Tango you do not call and display static HTML files; you execute application files. Application files are executed in the same way HTML files are viewed—simply by specifying the name of the file in the URL. The difference is that the static HTML file displays only one Web page, while one application file can execute any number or variety of Web pages, depending on user input or database results.

In the previous lesson you built a static Web page and viewed it by calling the HTML file in the browser. The static HTML file displayed only one Web page.

Result

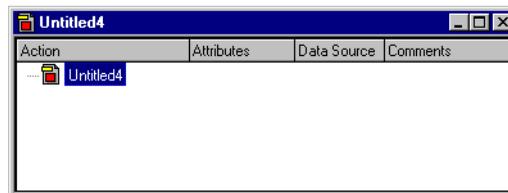
A Tango Web page that produces a message to mom.

Exercise

- 1 Open `mom.htm` in the text editor you are using, and copy the HTML text from the HTML file you created in the previous lesson.
- 2 Open Tango Editor.
- 3 From the **File** menu, choose **New**, or click the **New** icon on the toolbar.

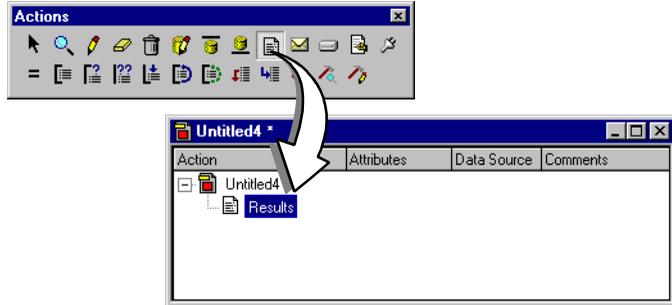


A blank application file window opens.

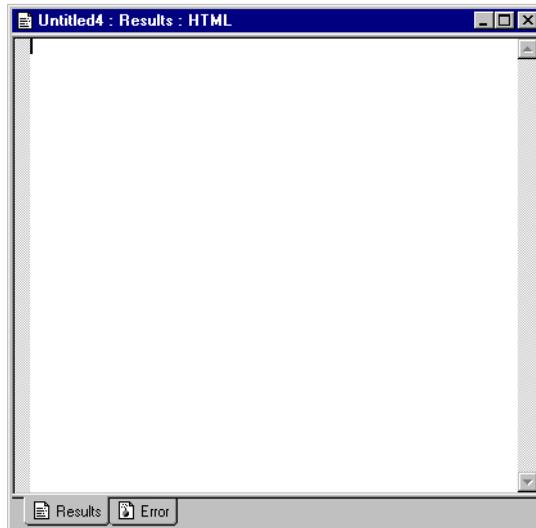


For more information on the Results action, see the *User's Guide*.

- 4 From the Actions bar, drag a Results action into the application file window.



A blank Results window opens.



- 5 Paste in the HTML you copied from `mom.htm`.

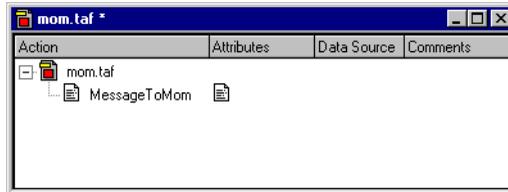
- 6 Close the Results editing window.

The application file now has one action item in it named Results.

- 7 Click the name of the Results action, and change the name to MessageToMom.

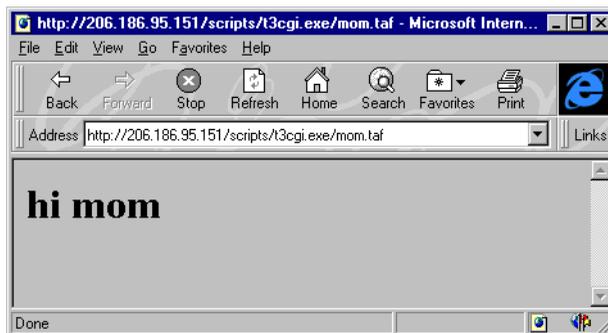
For information on Web server document root directories for different servers, see “Getting Started” on page 3.

- 8 Save the file as `mom.taf` in the Web server root folder, as you did with the static HTML Web page file in the previous lesson.



For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

- 9 Open your browser and execute `mom.taf`. Remember to include the path to your Tango plug-in or CGI in the URL line.



The `.taf` extension indicates that a file is saved as an application file rather than as an HTML file. The extension triggers the operation of the Tango server.

The Web page you see is the same as when you had called `mom.htm`, but this time it was generated by Tango Server executing the application file and sending the results back through the Web server to your browser.

LESSON A - 3

Adding a Second Results Action to the Application File

Purpose

To link together HTML found in separate application file actions.

Result

A Web page that produces a message to mom, as well as a message to dad.

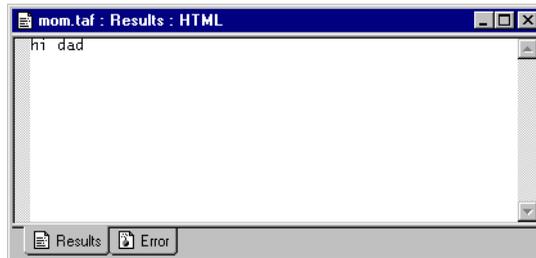
Exercise



- 1 Open `mom.taf` in Tango Editor.
- 2 Drag another Results action from the Actions bar, placing it below `MessageToMom`.

The Results editing window opens.

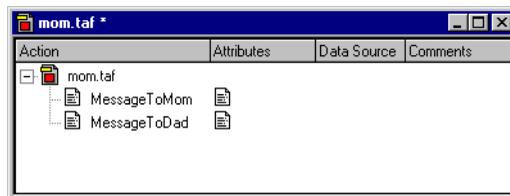
- 3 Type “hi dad” in the HTML window.



- 4 Close the Results editing window.

The Results are automatically saved.

- 5 Name the second action `MessageToDad` in the same way you named the first action.

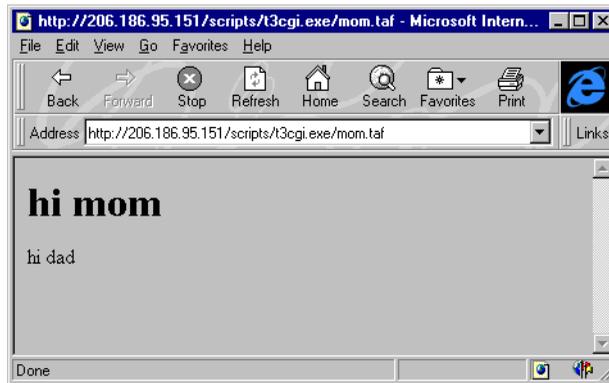


- 6 Save and replace `mom.taf`.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

7 Open your browser and execute the file as you did in the previous lessons.

You see your message, “hi mom” followed by your message, “hi dad”.



When Tango executed the `mom.taf` file, it linked together the HTML of both actions and displayed them in the browser.

Application files are executed from the top down, until control actions alter this path. In your current application file, the first action `MessageToMom` was executed, then Tango dropped down and executed `MessageToDad`.

The browser displayed both actions when the execution ended. The message “hi dad” appears on another line because “hi mom” is an HTML heading, which automatically places a hard return after itself.

LESSON A - 4

Inserting a Return Action Between the Results Actions

Purpose

To learn how a Return action works in an application file.

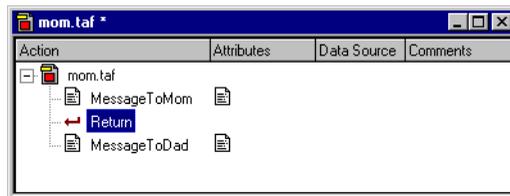
Result

The message to mom displayed in the browser.

Exercise

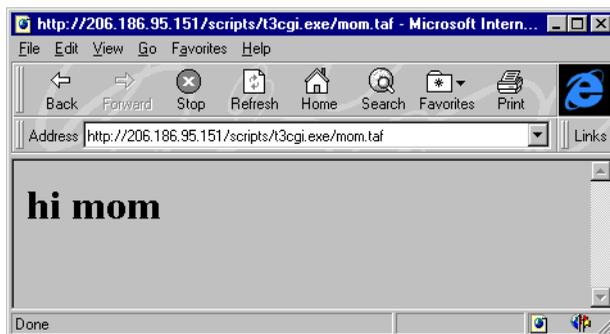


- 1 Return to Tango Editor with `mom.taf` open.
- 2 From the Actions bar, drag a Return action between the two Results actions.



When Tango comes to the Return action, it ends executing the file and returns all collected HTML to the browser.

- 3 Save and replace the application file as `mom.taf`.
- 4 Return to your browser and reload the file.



You see your message to mom as expected.

Tango encountered the Return action before the MessageToDad action. The next lesson tells you how to jump over the Return action and execute MessageToDad.

LESSON A - 5

Adding a Branch Action**Purpose**

To see how a branch action works.

Result

A Web page that displays the message, “hi dad” only.

Exercise

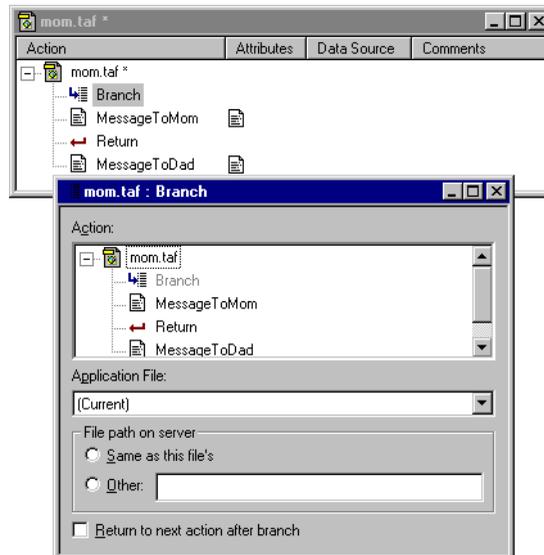
- 1 Return to Tango Editor with `mom.taf` open.
- 2 From the Actions bar, drag a Branch action above `MessageToMom`.



Tip The Branch action works similar to a programming “goto” command.

For more information on the Branch action, see the *User's Guide*.

The Branch window opens.



- 3 Choose `MessageToDad` from the list of actions in the current application file.

You are setting the branch to always go to the second Results action, `MessageToDad`, thereby skipping over the first Results action.

- 4** Close the Branch window.
- 5** Save `mom.taf`.
- 6** Return to your browser and reload the application file.



You see the message to dad. When Tango executed the application file, it began at the top and followed the directions of the first action, a branch, which in this case always forces Tango to go to `MessageToDad`. Because there are no other actions, Tango ends the execution of the file and sends all the collected HTML back to the browser.

LESSON A - 6

Adding If, Else If, and Else Actions in the Application File

Purpose

To display either the message to mom or the message to dad, based on a condition or expression.

Context

To display either the message to mom or the message to dad, a condition or expression must be used and evaluated. The If action evaluates expressions; if the expression resolves to true, the items grouped within the If action are executed by Tango Server.

For more information on the If action, consult the *User's Guide*.

In the mom/dad scenario, you are checking for the value for a variable named “parent”.

- If “parent=mom”, execute the MessageToMom action.
- If “parent=dad”, execute the MessageToDad action.

The user sends the value for “parent” by typing it in the URL and sending it as a *search argument* to Tango. The term search argument means the variable and its value are passed in the URL line of the browser.

Result

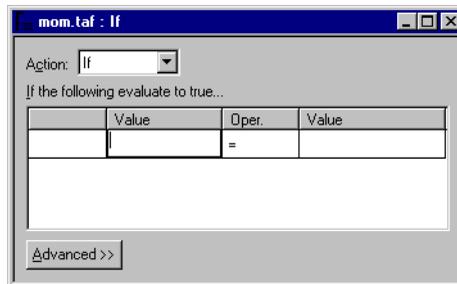
The message “hi mom” or “hi dad” displayed in the browser, depending on the values sent for the variable, “parent”.

Exercise

- 1 Return to `mom.taf` in the Tango Editor.
- 2 Click the Branch action and delete it; delete the Return action as well.
- 3 From the Actions bar, drag an If action above MessageToMom.



The If action window opens.



When Tango Server encounters an If action in the application file, it evaluates the condition set within it. If the condition is true, Tango Server proceeds to execute the indented list of actions within the If action. In the If action window, leave the default action set at If (so the action remains an If action, not an Else If or Else action).

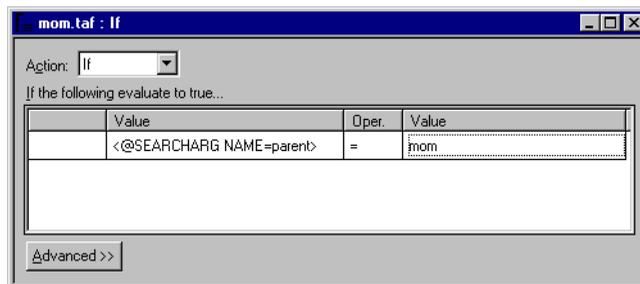
4 In the first **Value** field, type “<@SEARCHARG NAME=parent>”.



Note The window that appears for a new If action automatically displays a new value row with the first **Value** field ready for entering a value.

5 In the **Oper.** field, make sure = is selected.

6 In the second **Value** field, type “mom”.



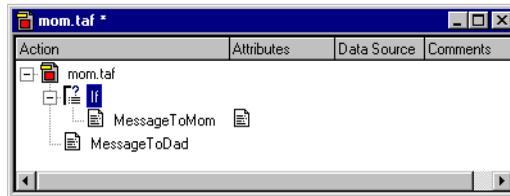


By setting the If action with these values, you are setting the condition that if the search argument named “parent” equals “mom”, then execute the indented Results action, MessageToMom. You insert the search argument in the URL after the name of the application file.

7 Close the If action window.

8 Drag the MessageToMom action into the If action.

MessageToMom becomes an indented action within the If action.



Q. If you saved `mom.taf` as is and executed it in the browser without saving a value for “parent”, what HTML message would appear?

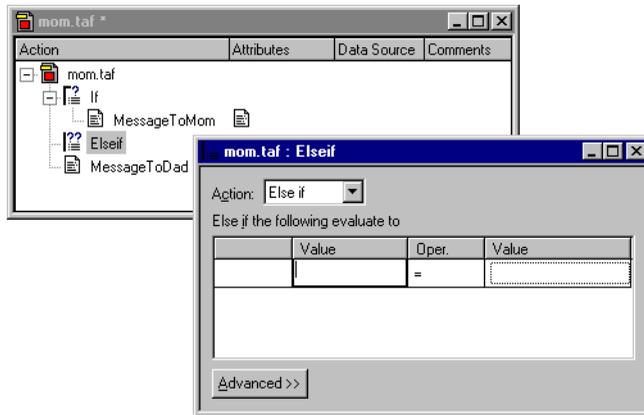
A. The result of the execution would be MessageToDad. The condition in the If action did not resolve to true during execution. However, if “parent=mom” was sent to Tango during execution, both messages would appear. MessageToMom would appear because the condition in the If action resolved to true, and MessageToDad would appear because it is in the next action in the order of execution.

You need to set a condition so that MessageToDad only appears if parent=dad.

9 Drag an Else If action above MessageToDad.



The Else If window opens.

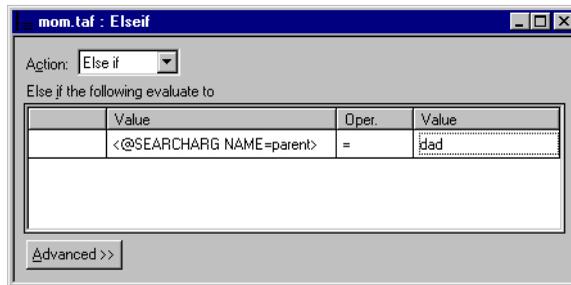


10 Enter the following values and operator.

Value: <@SEARCHARG NAME=parent>

Oper.: =

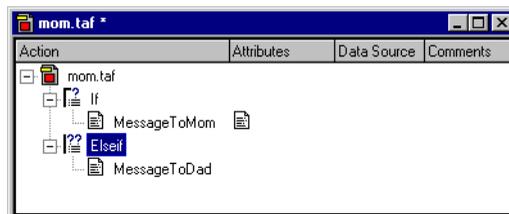
Value: dad



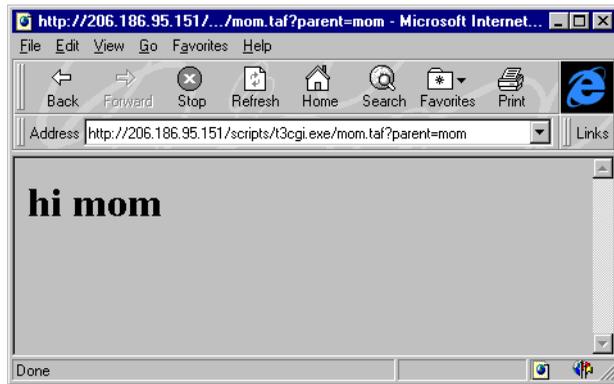
11 Close the Else If window.

12 Drag the MessageToDad action into the Else If action.

MessageToDad becomes an indented action within the Else If action, and is only executed if the Else If condition is met.



- 13 Save and replace `mom.taf`.
- 14 Return to your browser and type “`?parent=mom`” after the name of the application file, that is, `mom.taf?parent=mom`.
- 15 Press ENTER.



- Q.** What do you see?
- A.** Only the message to mom appears. The search argument called “parent” was equal to “mom” (because you typed it that way in the URL) and so `MessageToMom` was executed. `MessageToDad` was not executed because the Else If condition was not true—“parent” did not equal “dad”.
- 16 Type “`?parent=dad`” after the name of the application file, that is, `mom.taf?parent=dad`.
 - 17 Press ENTER.

Q. What do you see?

A. Only the message to dad appears because the search argument called “parent” was equal to “dad” (because you typed it that way), and so `MessageToDad` was executed. `MessageToMom` was not executed because the If condition was not true—“parent” did not equal “mom”.
 - 18 Delete “`?parent=dad`” from the URL line, and press ENTER.

Q. What page is produced?

A. No page is produced and the browser reports an error. Neither of the conditions for generating a message was true, so there was no HTML to display.

You have now developed some logical processes to determine which HTML pages—found within one application file—are displayed. This logic is based on the condition or value for a certain variable (the search argument named “parent”). This is an arbitrary variable name.

Additional Exercise

1 Type “?parent=blah” after the application file name in the URL line.

Q. What message do you get? Why?

A. Once again, the conditions for displaying either message are not being met. Tango does not execute either Results action, and sends HTML back to the browser.

2 Open the Else If action in `mom.taf` and change it to an Else action.



Note There is no place to enter values or expressions in the Else action. This is because it is evaluated to true whenever the If condition *is not met*.

3 Save and replace the application file. Then, execute it in the browser and send “?parent=mom” as the search argument.

Q. What message is displayed?

A. The message to mom is displayed because the If expression evaluated to true.

4 Execute `mom.taf` again, this time sending “?parent=dad” to Tango.

Q. What message do you get?

5 Send “?parent=blah”.

Q. What message is displayed? Why?

A. The message displayed is “hi dad”. The Else action is executed when the If action fails, and it did fail in both previous cases.

6 Execute `mom.taf`, and send “?cheese=blue”.

Q. What message do you get? Why?

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

Pass Values on the Web and in Tango

*Post Arguments and Search Arguments on the
Web and in Tango*

This tutorial introduces the concept of passing values, one of the most important aspects of Tango. Passed values, such as user inputs, are used to generate dynamic Web pages.

B1: HTML Forms

- passing data values using HTML forms

B2: HTML Forms in Tango, Many Application Files

- passing values from one application file to another
- using Tango meta tags to display search arguments in application files
- using Tango meta tags to display post arguments in application files
- using post arguments to change Web page appearance

B3: HTML Forms in Tango, One File

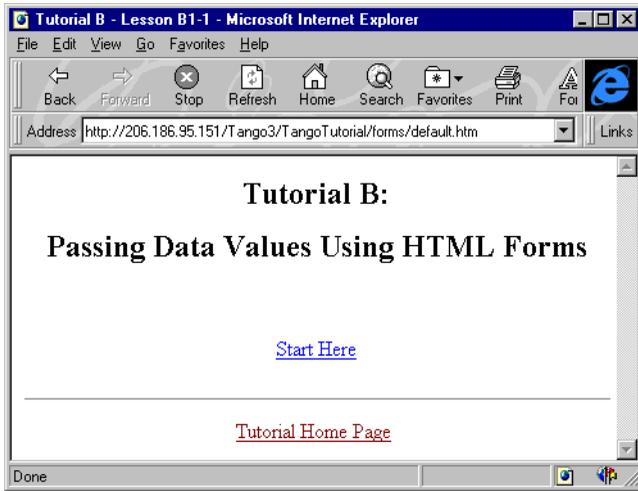
- passing values within one application file.

LESSON B - 1

Passing Data Values Using HTML Forms

- Purpose** To learn how data or information can be passed from one Web page to another, and why. Although this lesson does not involve Tango, the concept it teaches is fundamental to understanding Tango and dynamic Web sites.
- Context** Fundamental to a dynamic Web site is the concept (and method) of passing values. The search criteria entered by a user for an on-line search is an example of a value. Results from a database search are also values.
- The challenge for the Web site developer is that he or she cannot determine the values that the user enters. For example, in an on-line search, the user's actual search criteria varies based on the need or whim of the user. The search engine must be able to recognize search criteria no matter how much or often it varies.
- To solve this issue, dynamic Web site solutions use *variables*. The users, or the database, or even some other application provides values for these variables. For example, the search action can be set to search on the variable "Name". The value for name is provided by the user in the on-line search form. "Name" holds any value at any given time—"John" now, "Mary" five minutes later. The variable "Name" resolves to the value specified for it at that point in time.
- This lesson teaches how to create variables in HTML, and demonstrates how values are passed using these variables.
- Result** See how values are passed using a set of static HTML Web pages.
- Exercise**
- 1 Open your Web browser and load:
`/Tango3/TangoTutorial/Forms/default.htm`

The home page for this lesson appears.

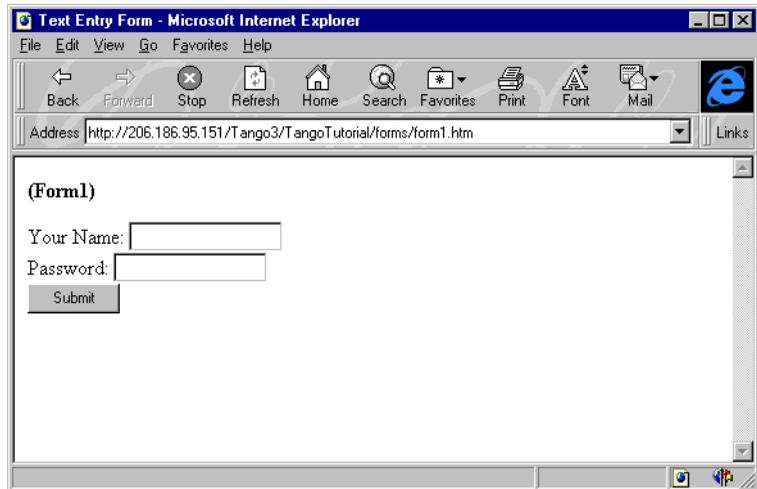


- 2 Click **Start Here**.
- 3 Check the URL line.



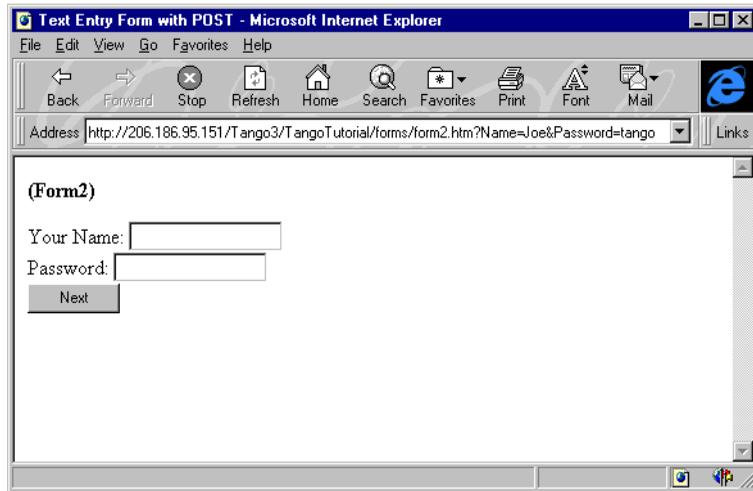
Note The **Start Here** link took you to the `form1.htm` page.

This page displays a form with two input fields named **Your Name** and **Password**.



- 4 Type your first name only, and some random password into the input fields. Do not use any special characters or spaces.
- 5 Click **Submit**.

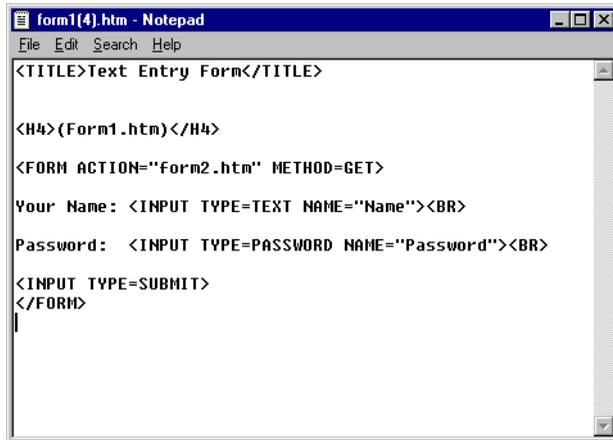
A second page appears, `form2.htm`, which contains the same two input fields as `form1.htm`.



Examine the URL line of the second page. You see your Web server address, the file folders or path to the file being displayed, `/Tango3/TangoTutorial/Forms/`, and the name of the file being displayed, `form2.htm`. After the file name there appears a “?”. The question mark is a delimiter, splitting up the URL line. Everything to the left of the question mark is the full and complete path to the file being displayed; everything to the right of the question mark is information that was sent from the previous page to this page via the HTML form, specifically, two variables, “Name” and “Password”, and their respective values.

- Q.** What do you notice about the values after the question mark in the URL?
- A.** The values should be familiar to you; they are the values you typed in the previous page’s form. The values you assigned to the variables “Name” and “Password” were sent to this second page, `form2.htm` in the URL line, appended to the file path. This is one way of passing values, called the *get* method.

- 6 Examine the HTML code that served to pass the values you entered on `form1.htm` to `form2.htm` by opening `form1.htm` in Windows Notepad.



```
form1(4).htm - Notepad
File Edit Search Help
<TITLE>Text Entry Form</TITLE>

<H4>(Form1.htm)</H4>
<FORM ACTION="form2.htm" METHOD=GET>
Your Name: <INPUT TYPE=TEXT NAME="Name"><BR>
Password: <INPUT TYPE=PASSWORD NAME="Password"><BR>
<INPUT TYPE=SUBMIT>
</FORM>
|
```



A hot link or anchor reference takes you to another Web page. It cannot pass any information to that next page. An *HTML form* is the element used to pass information. A form has an opening tag, `<FORM>`, and a closing tag, `</FORM>`. Within the form tags, there are HTML input elements. Parameters are specified to define the input element. Examine the parts of the first input.

```
Your Name: <INPUT TYPE=TEXT NAME="Name" >
```

An input with `TYPE=TEXT` physically establishes the form field, or white space into which the user enters data or values. The `NAME` parameter establishes and names a variable, in this case “Name”. The user-entered value is assigned to this variable. This first input sets up a container—the form field—to collect data from the user and gives the container a name, “Name”, so it can refer to the information later. The container is the variable. The label, `Your Name` is just text that is displayed beside the form field so that the user knows what type of information to enter.

```
<INPUT TYPE=PASSWORD NAME="Password" >
```

The next input works the same as the previous one. It sets up a physical form field on the screen, and names a variable, “Password”. This form field is slightly different than the first, however, because `TYPE=PASSWORD`. The text typed into the field by the user is encoded by asterisks or dots. The variable is established and named “Password” to logically describe the data that is entered or assigned to it.

```
<INPUT TYPE=SUBMIT >
```

An input of `TYPE=SUBMIT` physically creates a button on the Web page. When clicked, the button collects and sends—or submits—the values entered by the user to a program on the server such as Tango, which then returns the next Web page.

Where and how the submit button sends the variables and their values is specified by the `ACTION` and `METHOD` parameters in the opening form tag. `ACTION` specifies what page users go to when they click the button. Notice the `ACTION` in the above form is `/Tango3/TangoTutorial/Forms/form2.htm`, which is the page you went to when you clicked the button. In this example, the values were not passed to any server program such as Tango because the program was not specified in the `ACTION` parameter.

The `METHOD` parameter specifies how the values are passed. There are two methods, *get* and *post*. In this lesson, you used `METHOD=GET`. The values were passed in the URL line as *search arguments*. The *post* method will be discussed shortly.



Tip You may use HTML forms just to create navigational buttons, without passing any values. All you need to include between the form tags is a submit input, and in the `ACTION` parameter of the opening form tag you specify the page you want displayed when the button is pressed.

7 Close the text editing window. Your browser still displays `form1.htm`.

8 Re-enter your name and password, and click **Submit**.

The `form2.htm` appears, and the values that were passed to it appear in the URL line as before.

9 The `form2.htm` presents you with another form. View the HTML source code for this page and examine how this form is set up.

```

form2[G].htm - Notepad
File Edit Search Help
<TITLE>Text Entry Form with POST</TITLE>

<FORM ACTION="form3.htm" METHOD=Post>

Your Name: <INPUT TYPE=TEXT NAME="Name"><BR>
Password: <INPUT TYPE=PASSWORD NAME="Password"><BR>

<INPUT TYPE=SUBMIT name="button" VALUE="Next">
</FORM>

```



The code for the form is the same as that on `form1.htm`, except the method is now *post* instead of *get*. The `POST` method passes values invisibly in the HTTP header that is passed with every Web page. The values are being *posted* to the CGI or other server side program, such as Tango. The variables and their values do not appear in the URL. Variables and values passed with the post method are termed *post arguments*. The terms search argument and post argument are important to knowing and understanding Tango.



Note When you do not specify a method in the form, the get method is used.

10 Close your text editor.

11 In the browser, enter a name and password in the form and click **Submit**.

If you are using any of the modern browsers, you are likely to receive an error.



The error occurs because the browser expects to *post* your values to a server side program or application. However, you did not specify one in your action parameter; you simply specified another static Web page file. With no server side program such as Tango to return a Web page, the server returns an error.

In the next lesson, you transfer these static Web page files into application files and run through them again, with better results.

Q. What is a search argument and a post argument?

A. A search argument is a value that is passed in the URL line of the browser and is delimited by a question mark (?) after the file path and name. Values collected from a form using `METHOD=GET` become search arguments. A post argument is a value that is passed in the HTTP header, invisible to the user. Values collected from a form using `METHOD=POST` become post arguments.

LESSON B 2 - 1

Passing Values From One Application File to Another

Purpose

To pass values from one application file to another, with emphasis on the post method.

Context

In the previous lesson, you saw that values can be passed from one Web page to another using the HTML form element. In this lesson, you use the same form element to pass values to a server side application, Tango.

The static Web pages from the previous lesson are made into application files. On each page the post arguments and search arguments are displayed using Tango meta tags.

Result

You make the static Web pages from the previous lesson into application files. You display the post arguments and search arguments on each page using Tango meta tags.

Exercise

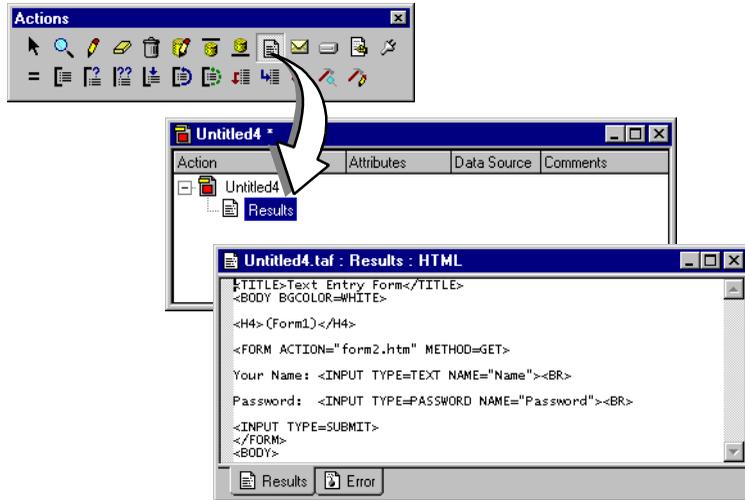
You need to toggle between Tango Editor and your text editor for this exercise.

- 1 Navigate to the `\Tango3\TangoTutorial\Forms\` folder and open `form1.htm` in your text editor. You may also call up `form1.htm` in the browser and view the source code.
- 2 Select the entire text of `form1.htm` and copy it.
- 3 In Tango Editor, open a new application file.
- 4 Drag a Results action into the new application file.

An HTML editing window opens for the Results action.



5 Paste in the text you copied from form1.htm.



6 Identify the ACTION=" form2.htm" line in the form, and change it to read ACTION=" /Tango3/TangoTutorial/Forms/form2.taf".

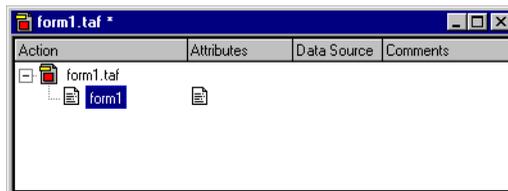
If you are not using the Tango plug-in, add the appropriate Tango CGI to the file path in the action. (For example, for Microsoft's Internet Information Server, the action is ACTION=" /scripts/t3CGI.exe/Tango3/TangoTutorial/Forms/form2.taf".)



You are converting your static .htm files to application files, which have the .taf extension. If you are using the Tango plug-in, it is important to use the .taf extension because that is what triggers the operation of Tango Server.

7 Close the HTML editing window for the Results action.

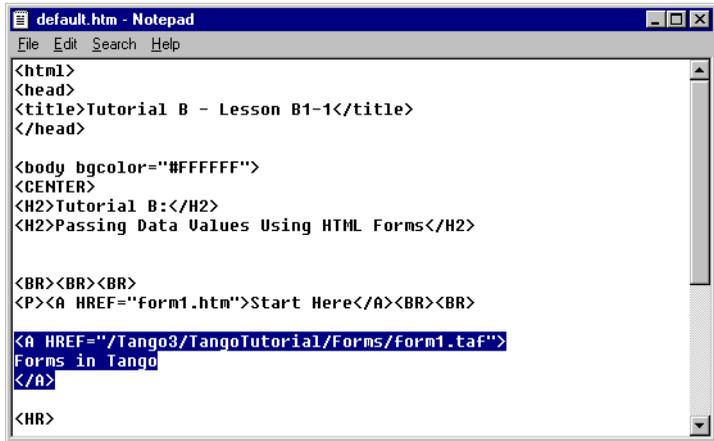
8 Rename the Results action, form1.



- 9 Save the application file as `form1.taf` in the `\Tango3\TangoTutorial\Forms\` folder and close `form1.taf` in Tango Editor.
- 10 Repeat steps 1 to 9 for `form2.htm`, `form3.htm`, and `form4.htm`, converting them into application files named `form2.taf`, `form3.taf`, and `form4.taf`. Remember to change the file path and file name specified in the ACTION of `form2.htm` and `form3.htm` when you paste them into the application file. Remember also to include the Tango CGI if necessary. No changes are necessary on `form4`.
- 11 In your text editor, open `default.htm` found in the Forms folder.
- 12 Add the following hot link, linking to the first form page in Tango, below the **Start Here** link.

```
<A HREF="/Tango3/TangoTutorial/Forms/form1.taf">Forms in Tango</A>
```

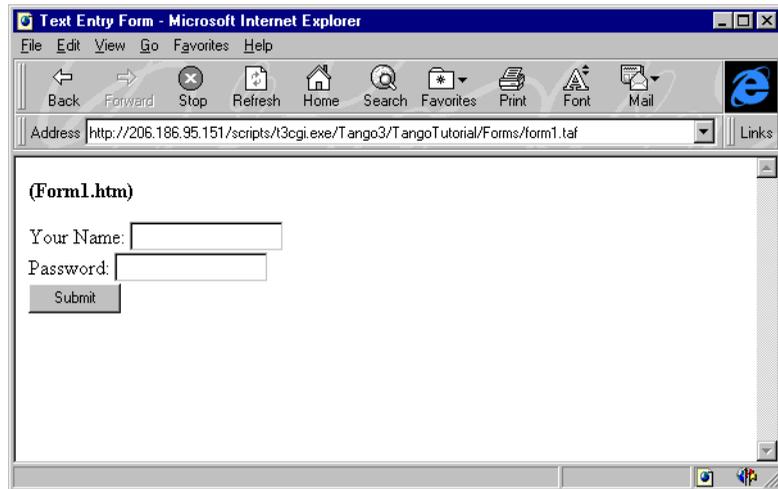
If you are not using the Tango plug-in, specify the Tango CGI in the path as well.



```
default.htm - Notepad
File Edit Search Help
<html>
<head>
<title>Tutorial B - Lesson B1-1</title>
</head>
<body bgcolor="#FFFFFF">
<CENTER>
<H2>Tutorial B:</H2>
<H2>Passing Data Values Using HTML Forms</H2>
<BR><BR><BR>
<P><A HREF="form1.htm">Start Here</A><BR><BR>
<A HREF="/Tango3/TangoTutorial/Forms/form1.taf">
Forms in Tango
</A>
<HR>
```

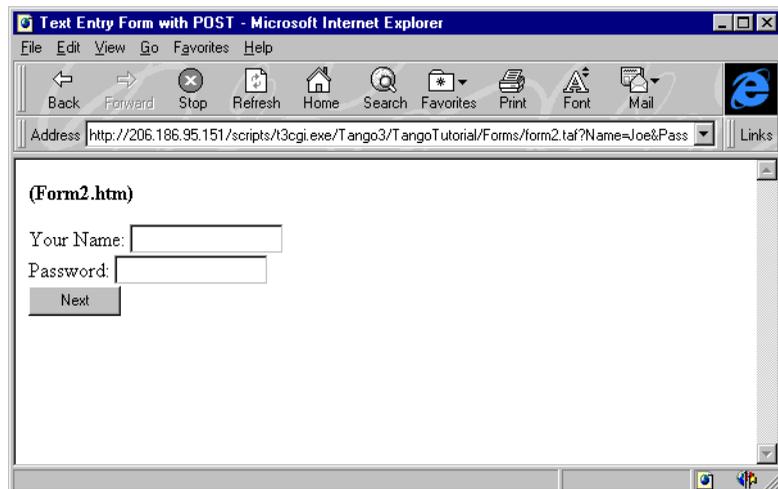
- 13 Save and replace `default.htm`.
- 14 Open your browser and call up the default page from the `Forms` folder. You may have to reload or refresh the page.
- 15 Click the **Forms in Tango** link.

The first form page appears, processed through Tango. It looks exactly like `form1.htm` because you copied and pasted the HTML code from that other file.



16 Enter a name and password, and click **Submit**.

The `form2.taf` appears.





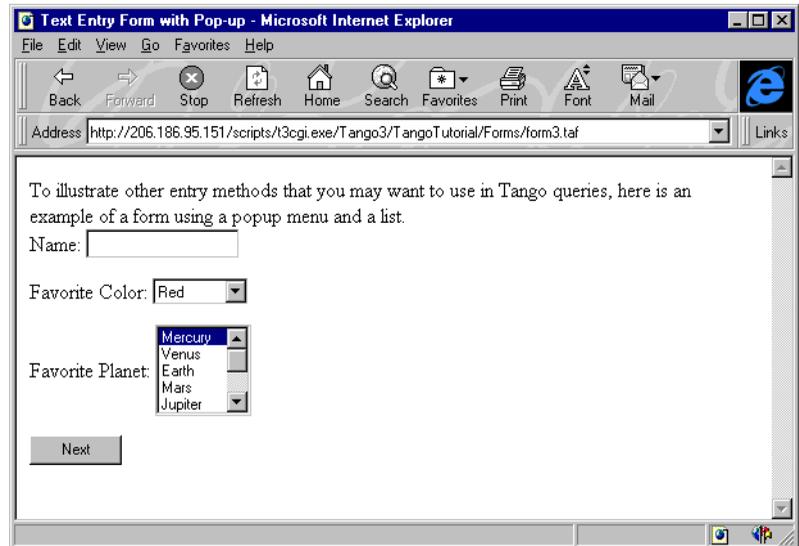
Recall that the method for the `form1.taf` form is set to `GET`, so the values you typed in were passed in the URL to this page. The variables and their values are therefore classified as search arguments. Because you are now executing the files with Tango Server, it is more accurate to state that the search arguments were passed *to Tango*. So far, you have not instructed Tango to do anything with the search arguments.

Two disadvantages of using the `get` method is that values passed are visible, which is inappropriate for such things as passwords, and browsers' URL lines can only hold 255 characters, thereby limiting the amount of information you can pass as search arguments.

17 Enter a name and password in the form of `form2.taf`.

18 Click **Next**.

The `form3.taf` appears, displaying a form with a drop-down list and a scrolling list in addition to the text form field.





Recall that the method for the form on `form2.taf` is set to `POST`, so the variables and their values are sent invisibly in the HTTP header, and are classified as post arguments. When you tried posting a name and password with the static Web page files in the previous lessons, you received an error because there was no server side application to which to send, or post, the post arguments.

Clicking **Next** on `form2.taf` now returns `form3.taf` successfully because you are posting the post arguments to Tango, and Tango returns a Web page generated through `form3.taf`.

The values you entered for name and password do not appear in the URL line. Using the post method allows you to keep such information as passwords invisible, and allows you to pass much more information or values than you could in the URL line of the browser.

The form on `form3.taf` contains two form elements you have not yet seen in these lessons, the drop-down list and the scrolling list.

19 View the document source code for `form3.taf`.

A screenshot of a Notepad window titled "form3(1).htm - Notepad". The window contains the following HTML code:

```
<FORM
ACTION="scripts/t3cgi.exe/Tango3/TangoTutorial1/Forms/Form
4.taf" METHOD=Post>

Name: <INPUT TYPE=TEXT NAME="Name"><BR><BR>

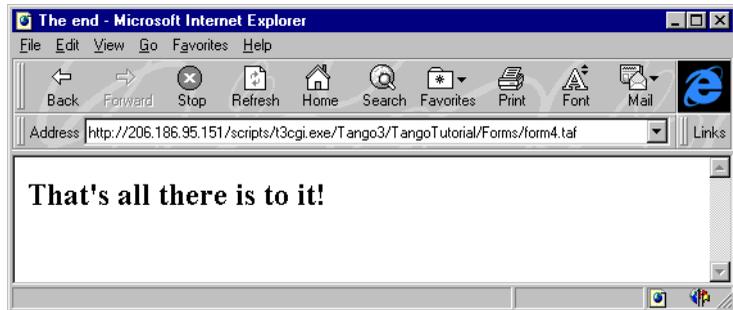
Favorite Color:
<SELECT NAME="Color" SIZE=1>
<OPTION>Red
<OPTION>Green
<OPTION>Blue
<OPTION>Purple
<OPTION>Yellow
<OPTION>Orange
<OPTION>Brown
<OPTION>Black
</SELECT>
<BR>
<BR>
Favorite Planet:
<SELECT NAME="Planet" SIZE=5>
<OPTION>Mercury
<OPTION>Venus
<OPTION>Earth
<OPTION>Mars
<OPTION>Jupiter
```



Drop-down or pop-up lists, and scrolling lists are created using the HTML `SELECT` element, with a list of the options displayed between the opening and closing `SELECT` tags. The only difference between a drop-down list and a scrolling list is the value of the `SIZE` attribute in the opening `SELECT` tag. `SIZE=1` means only one option is displayed in the browser. Anything above 1 displays a list of options.

Note the action that is specified in the opening tag of `form4.taf`. **Next** executes `form4.taf` when clicked. The method specified for passing values from this form to `form4.taf` is the post method. Note also the names of the variables established on this form to collect values: `Name`, `Color`, and `Planet`.

- 20 Close the text editing window.
- 21 Enter a name in the **Name** form field, and select a color and planet from the lists.
- 22 Click **Next** to send these values to Tango to be used in the execution of `form4.taf`.



The last page in the series of forms appears, produced by the execution of `form4.taf`. Note the value you entered or selected for the variables, `Name`, `Color`, and `Planet` do not appear in the URL line. The values were passed using `METHOD=POST`. In the next Lesson, you will use these values in Tango.

LESSON B 2 - 2

Using Meta Tags to Display Search Arguments Passed From a Form in Application Files

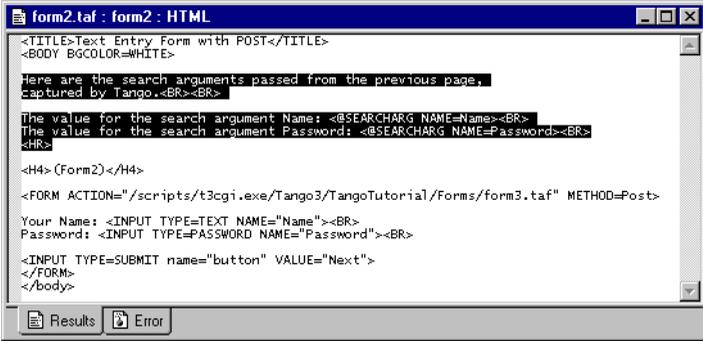
- Purpose** To learn how Tango captures values—search arguments—passed to it from a Web page, and how to display these values on the subsequent Web page using the `<@SEARCHARG>` meta tag.
- Context** In the previous lesson, you learned how HTML forms can pass values from one Web page to another through Tango Server or the CGI. Think of Tango Server as a middleman. When a user clicks a button to execute an application file, the values passed from the current Web page are sent to Tango Server and are incorporated in the execution of the application file. The application file produces a Web page in the browser. In the previous lesson you did not instruct Tango to do anything with the values passed. In this lesson, you display the values passed using the `<@SEARCHARG>` Tango meta tag.
- Result** At the top of the second form page, `form2.taf`, in the browser, the values passed to that page—or more accurately, to that file—are displayed.
- Exercise**
- 1 Return to Tango Editor, and open `form2.taf`, if it is not already open.
 - 2 Open the Results action titled `form2` by doing one of the following:
 - Double click the action's icon.
 - Click the action icon and choose **Results HTML** from the **Attributes** menu.
 - Right click the selected action and choose **Results HTML** from the context-sensitive menu that appears.
- The HTML editing window appears.

3 After the Title line, type the following:

Here are the search arguments passed from the previous page, captured by Tango.

The value for the search argument Name: <@SEARCHARG NAME=Name>

The value for the search argument Password: <@SEARCHARG NAME=Password>
<HR>



```
form2.taf : form2 : HTML
<TITLE>Text Entry Form with POST</TITLE>
<BODY BGCOLOR=WHITE>
Here are the search arguments passed from the previous page,
captured by Tango.<BR><BR>
The value for the search argument Name: <@SEARCHARG NAME=Name><BR>
The value for the search argument Password: <@SEARCHARG NAME=Password><BR>
<HR>
<H4>(Form2)</H4>
<FORM ACTION="/scripts/t3cgi.exe/Tango3/TangoTutorial/Forms/form3.taf" METHOD=Post>
Your Name: <INPUT TYPE=TEXT NAME="Name"><BR>
Password: <INPUT TYPE=PASSWORD NAME="Password"><BR>
<INPUT TYPE=SUBMIT name="button" VALUE="Next">
</FORM>
</body>
```



For more information on the <@SEARCHARG> meta tag, refer to the meta tag reference material.

In Tango, the values for search arguments are referenced with the <@SEARCHARG> meta tag. Search arguments appear in the URL, but if you want to display them on the actual Web page, you must use the <@SEARCHARG> meta tag.

To reference a passed value, you identify the type of value it is—a search argument—and the name of the variable that holds the value, Name. <@SEARCHARG NAME=Name> resolves to the value you enter in the form on the previous page, and <@SEARCHARG NAME=Password> resolves to the value you enter for password.



Note You can also reference search arguments using the <@ARG> meta tag. <@ARG> is used to reference either search arguments or post arguments.

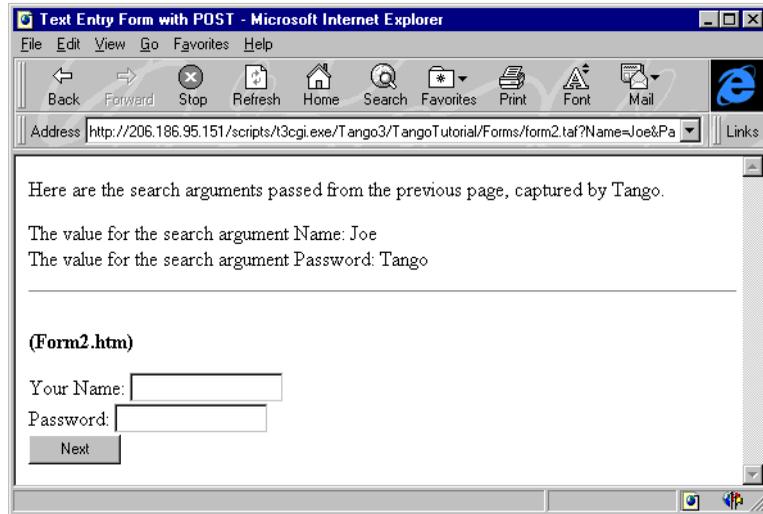
- 4 Close the Results HTML editing window.
- 5 Save form2.taf.
- 6 Return to your browser and navigate to form1.taf.

7 Enter a name and password.

These become search arguments when **Next** is clicked because **METHOD=GET** in the form.

8 Click **Submit**.

The `form2.taf` is executed, displaying the HTML that is in the **Results** action of the `form2.taf` application file.



Notice `<@SEARCHARG NAME=Name>` and `<@SEARCHARG NAME=Password>` resolved to the name and password values you entered on the previous form, and that these match what you see after the question mark in the URL. On this Web page, you are simply displaying the values, but there are numerous other ways to use these values. For example, you can use Tango meta tags in Tango search actions, where they would resolve to the present value—the user-entered value—for the search criteria.

Note also, to display successfully the values for the search arguments—`Name` and `Password`, you must know the method being used and the variable names of the form on the previous page (the page that would be collecting and sending the values to `form2.taf`).

LESSON B 2 - 3

Using Meta Tags to Display Post Arguments in Application Files

Purpose

To display the post arguments sent from one Web page to another using the `<@POSTARG>` meta tag.

Context

In the previous lesson, you used `<@SEARCHARG>` to display the values that came from `form1.taf` to `form2.taf`, because the values were search arguments. The values passed from `form2.taf` to `form3.taf` are post arguments, however, because the form on `form2.taf` is using the post method. As a result, you must use the `<@POSTARG>` meta tag to reference the values.

Result

The post arguments gathered by the form on `form2.taf` are displayed at the top of the Web page for `form3.taf`. This is the only way to see the values of the post arguments.

Exercise

1 Return to Tango Editor, and open `form3.taf`, if it is not open already.

2 Open the Results action titled `form3`.

The HTML editing window opens.

3 After the `Title` line, type the following:

```
Here are the post arguments passed from the previous
page, captured by Tango.<BR><BR>
```

```
The value for the post argument Name: <@POSTARG
NAME=Name><BR>
```

```
The value for the post argument Password: <@POSTARG
NAME=Password><BR>
```

```
<HR>
```

```

form3.taf : form3 : HTML
<TITLE>Text Entry Form with Pop-up</TITLE>
Here are the post arguments passed from the previous page,
captured by Tango.<BR><BR>
The value for the post argument Name: <@POSTARG NAME=Name><BR>
The value for the post argument Password: <@POSTARG NAME=Password><BR>
<HR>
<P>To illustrate other entry methods that you may want to use in Tango
queries, here is an example of a form using a popup menu and a list.<BR>
<FORM ACTION="/scripts/t3cgi.exe/Tango3/TangoTutorial/Forms/form4.htm" METHOD=Post>
Name: <INPUT TYPE=TEXT NAME="Name"><BR><BR>
Favorite Color:
<SELECT NAME="Color" SIZE=1>
<OPTION>Red
<OPTION>Green
<OPTION>Blue
<OPTION>Purple
<OPTION>Yellow
<OPTION>Orange
<OPTION>Brown
<OPTION>Black
</SELECT>
<BR>
<BR>
Favorite Planet:
<SELECT NAME="Planet" SIZE=5>
<OPTION>Mercury
<OPTION>Venus
<OPTION>Earth
<OPTION>Mars
<OPTION>Jupiter
<OPTION>Saturn
<OPTION>Uranus
<OPTION>Neptune
<OPTION>Pluto
</SELECT>
<BR>
<BR>
<INPUT TYPE="SUBMIT" name="button" VALUE="Next">
</FORM>
Results Error

```



For more information about the `<@POSTARG>` meta tag, refer to the meta tag reference material.

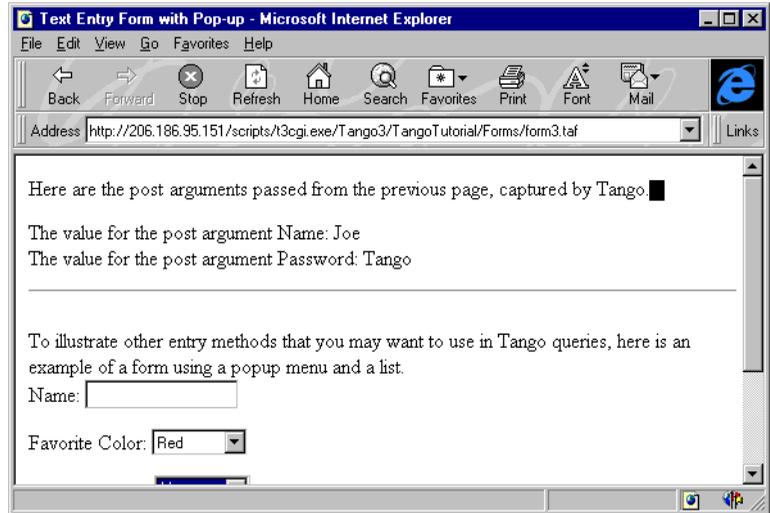
In Tango, the values for post arguments—the values passed in a form using `METHOD=POST`, are referenced with the `<@POSTARG>` meta tag. The meta tags used above resolve to the value entered by the user on the form `form2.taf`. Remember that `form2.taf` is sending post arguments, not search arguments. If it were sending search arguments, the values above would be blank.

- 4 Close the Results HTML editing window.
- 5 Save `form3.taf`.
- 6 Return to your browser and navigate to `form2.taf`.
- 7 Enter a name and password.

Because `METHOD=POST` in this form—the form that sends values to `form3.taf`—the values you enter become post arguments when **Next** is clicked.

8 Click Next.

The `form3.taf` is executed, displaying the HTML that is contained in the Results action of the `form3.taf`.



LESSON B2 - 4

Using Post Arguments to Change Web Page Appearance

Purpose

To demonstrate a practical use for referencing the values of post arguments and search arguments.

Context

In the previous two lessons, you simply displayed the values for the search arguments and post arguments sent from the preceding page. In this lesson, you collect a name, favorite color, and favorite planet from the user, and then demonstrate a use for these values.

Result

When you submit your name, favorite color, and favorite planet, the next page you see, generated by `form4.taf`, displays the message, “Hello, (*your name*)! (*your favorite planet*) is my favorite planet, too!”. The background color of the page is your chosen favorite color.

Exercise

- 1 Return to Tango Editor, and open `form4.taf`, if it is not open already.

The `form4.taf` is the target file of the post arguments sent from `form3.taf`—name, color, and planet.

- 2 Open the Results action titled `form4`.

The HTML editing window opens.



To change the background color of this page in standard HTML, you specify the color in the opening body tag as a body attribute, `<BODY BGCOLOR=Red>`.

To make the background color the same as the favorite color chosen by the user, simply replace the actual color in the previous example with the `<@POSTARG>` meta tag that holds the value for color. Upon execution, the meta tag resolves to whatever color the user chooses for favorite color.



Note Most browsers support colors specified in English words as well as in numbers.

- 3 Add the following body tag with the background color attribute in the Results HTML window for form4:

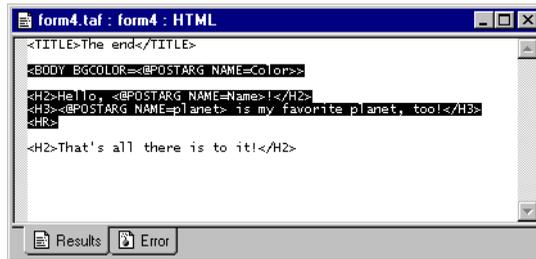
```
<BODY BGCOLOR=<@POSTARG NAME=Color>>
```



Note Be careful to close all brackets.

- 4 Add the following after the opening <BODY> tag:

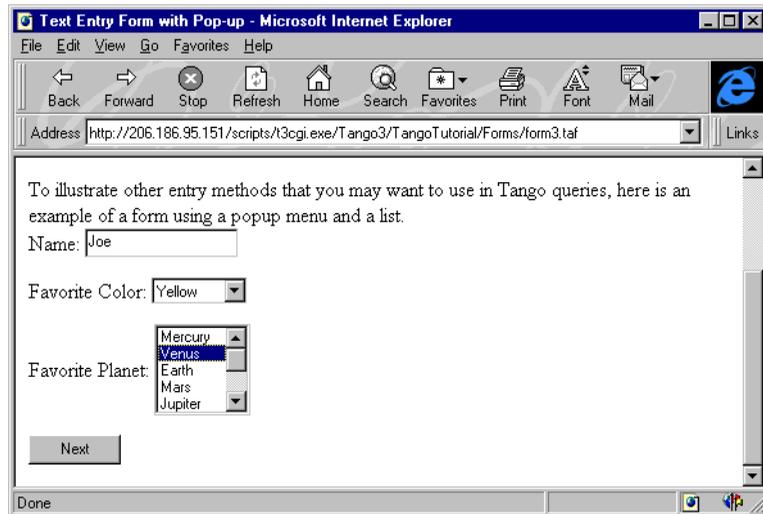
```
<H2>Hello, <@POSTARG NAME=Name>!</H2>  
<H3><@POSTARG NAME=planet> is my favorite planet,  
too!</H3>  
<HR>
```



- 5 Close the Results HTML editing window.
- 6 Save form4.taf.
- 7 Return to your browser and navigate to form3.taf.

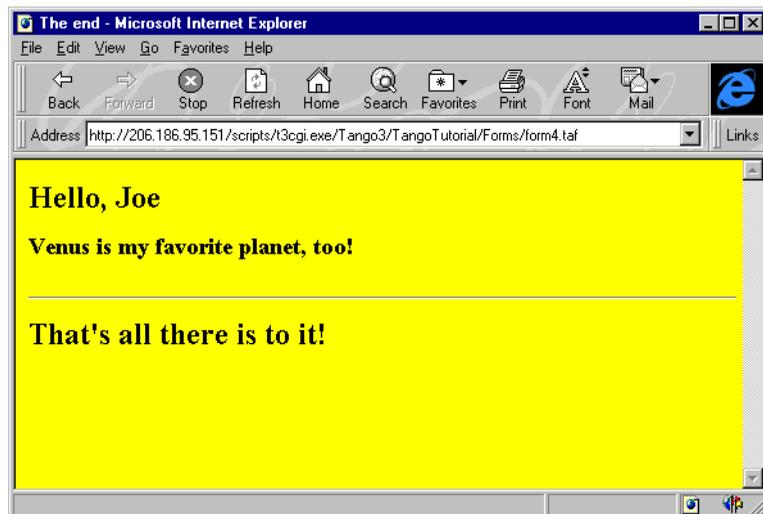
8 Enter your name, and select a favorite color and planet.

Your selections become post arguments for use in `form4.taf` because this form is using `METHOD=POST`.



9 Click Next.

The `form4.taf` file is executed and a page with your favorite background color and personalized message is displayed.



10 Click the browser's **Back** button to return to `form3.taf`.

- 11 Enter different values for name, color, and planet, and click **Next**.

The `form4.taf` page changes according to your selections; it is a dynamic Web page.

Additional Exercise

- 1 In Tango Editor, open the `form4` action in the `form4.taf` application file.
- 2 Change the three `<@POSTARG...>`'s to `<@ARG...>`'s, keeping the names the same.
- 3 Save it as `form4.taf` and check it in the browser.
- 4 Enter values for the three form fields, and click **Next**.
 - Q.** Is the next page personalized like before?
 - A.** Yes, because the `<@ARG>` meta tag captures variables regardless if they are search arguments or post arguments. In the previous case, it looks for arguments `Name`, `Color`, and `Planet` whether they are found in the URL or in the HTTP header. If you use the `<@ARG>` meta tag, you can change the `METHOD` in the forms from `GET` to `POST` or vice versa as often as you like and your application files will still work properly.

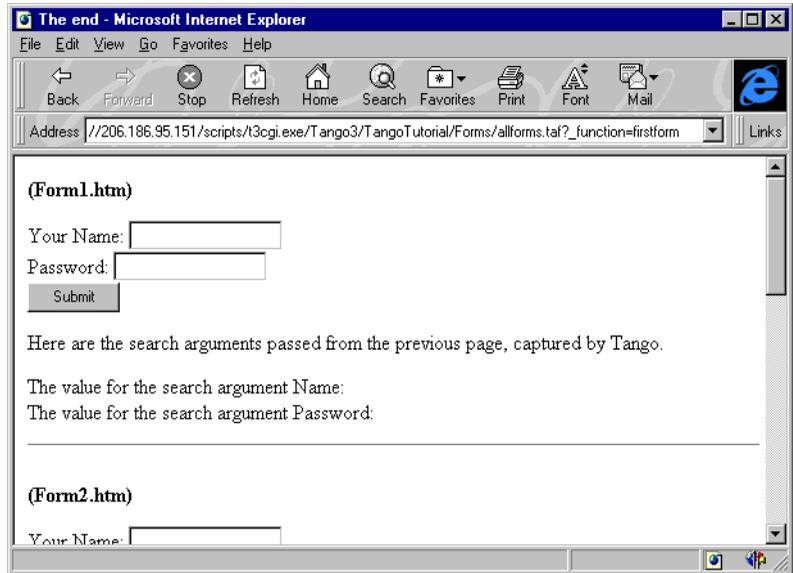
LESSON B 3 - 1

Passing Values Within One Application File

- Purpose** To pass values between different actions in one application file and to use values to determine which action executes in an application file.
- Context** In the previous set of lessons, you learned to pass values—search arguments and post arguments—from one Web page to another, and then from one application file to another. In this lesson, you combine the four different actions of the form application files into one file with If actions to determine which action to display at which time.
- Result** One application file that produces all four form pages.
- Exercise**
- 1 Open a new application file in Tango Editor.
 - 2 Open `form1.taf`.
 - 3 Drag the `form1` action into the new application file.
 - 4 Close `form1.taf`.
 - 5 Rename `form1` to `formA`.
 - 6 Open `form2.taf`, and drag the `form2` action into the new application file under `formA`. Close `form2.taf`.
 - 7 Rename `form2` to `formB`.
 - 8 Repeat steps 6 and 7 for `form3` and `form4`, renaming the actions `formC` and `formD`.
 - 9 Save the untitled application file as `allforms.taf` in the `\Tango3\TangoTutorial\Forms\` folder.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

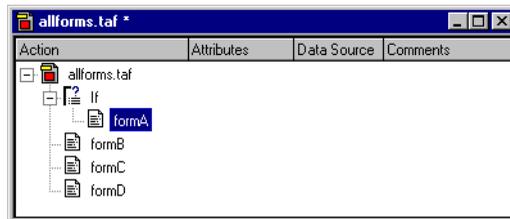
10 Execute `allforms.taf` in the browser.



All the forms are displayed one after the other. This is because there are no control actions to determine which Results action to execute, so they are all executed. In the next series of steps, you make the execution of each Results action conditional by placing them within an If or Else If action.

11 Return to `allforms.taf` in Tango Editor.

12 Hold down the CTRL key and drag an If action above formA. Then move formA into the If action.



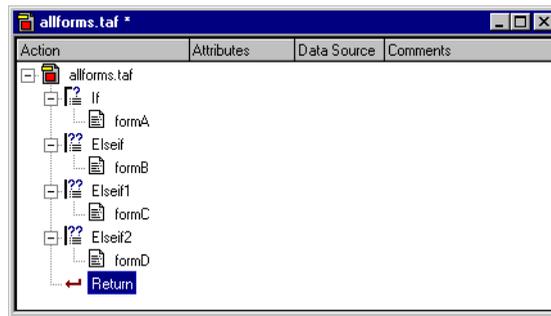


Tip When you hold down the CTRL key while dragging actions into an application file, the action's editing window does not open.

- 13 Hold down the CTRL key and drag an Else If action above formB. Then move formB into the Else If action.
- 14 Hold down the CTRL key and drag an Else If action above formC. Then move formC into the Else If action.
- 15 Hold down the CTRL key and drag an Else If action above formD. Then move formD into the Else If action.
- 16 Drag a Return action in at the end of the series of Else If actions.



The Return action is not necessary as the application file ends execution when there is nothing left to do anyway. However, it is good Tango programming form to include it.

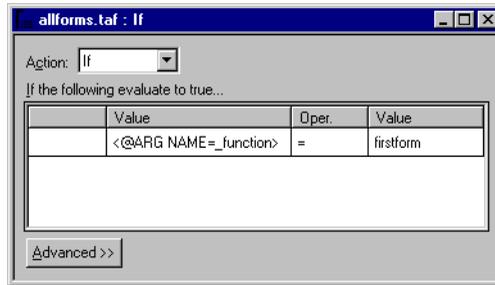


Now you must set the condition in the If and Else If actions. You set up this file so that a variable called `_function` and a value for it is passed as a search argument with every execution of `allforms.taf`.

Each If or Else If action checks the value of `_function` and if there is a match, the action within that If executes. In essence, the value passed for `_function` determines which Results action executes.

- 17 Open the If action by double clicking it.
- 18 Insert `<@ARG NAME=_function>` into the first **Value** field.

- 19 Leave **Oper.** set to =, and type “firstform” in the second **Value** field.

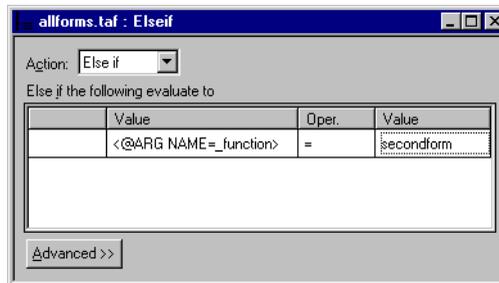


- 20 Close the If action's editing window.



This equation sets the condition so that if the argument is equal to `firstform`, the `formA` action is executed. Otherwise, it is not and Tango continues on to process the next Else If, which you will set to look for the argument `_function=secondform`.

- 21 Open the first Else If action, which contains formB.
 22 Insert `<@ARG NAME=_function>` into the first **Value** field.
 23 Leave **Oper.** set to =, and type “secondform” in the second **Value** field.



- 24 Close the Else If action's editing window.
 25 Open the second Else If action, which contains formC.
 26 Insert `<@ARG NAME=_function>` into the first **Value** field.
 27 Leave **Oper.** set to =, and type “thirdform” in the second **Value** field.

- 28 Close the Else If action's editing window.
- 29 Open the third Else If action, which contains formD.
- 30 Insert `<@ARG NAME=_function>` into the first **Value** field.
- 31 Leave **Oper.** set to `=`, and type `"fourthform"` in the second **Value** field.
- 32 Close the Else If action's editing window.



Note The values for the variable `_function` are completely arbitrary, as is the name of the variable `_function`. The point is not what the value is, but what is executed or displayed when the variable `_function` has a certain value.



If you saved and executed `allforms.taf` now, it would not work properly. Although you have proper If and Else If actions that evaluate the value for the search argument `_function`, you have not yet properly sent these values from any one HTML page. Each of the Results actions—`formA`, `formB`, `formC`, and `formD`—are a complete Web page. If you start at `formA`, then a value for `_function` must be sent *from* `formA` so that `formB` can next be executed and displayed. `FormA` must send `_function=secondform` back to the `allforms.taf`, either as a search argument or post argument.

- 33 Open the `formA` action to edit the HTML.

```
allforms.taf : formA : HTML
<TITLE>Text Entry Form</TITLE>
<BODY BGCOLOR=WHITE>
<H4> (Form1) </H4>
<FORM ACTION="/scripts/t3cgi.exe/Tango3/TangoTutorial/Forms/form2.taf" METHOD=GET>
Your Name: <INPUT TYPE=TEXT NAME="Name"><BR>
Password: <INPUT TYPE=PASSWORD NAME="Password"><BR>
<INPUT TYPE=SUBMIT>
</FORM>
</BODY>
```



Because the form's `METHOD` is set to `GET`, all the values passed from the form are search arguments. All you need to do is add another input element for the variable `_function`, and specify its value as "secondform". The user does not need to see this search argument, so a hidden input can be used.

34 After the password input line, type the following:

```
<INPUT TYPE=HIDDEN NAME="_function"
VALUE="secondform">
```

35 Change the `ACTION` attribute of the form to read:

```
ACTION="allforms.taf"
```

You do this so `allforms.taf` executes again when the user clicks **Submit**.

```
allforms.taf : formA : HTML
<TITLE>Text Entry Form</TITLE>
<BODY BGCOLOR=WHITE>
<H4>(Form1)</H4>
<FORM ACTION="/scripts/t3cgi.exe/Tango3/TangoTutorial/Forms/allforms.taf" METHOD=GET>
Your Name: <INPUT TYPE=TEXT NAME="Name"><BR>
Password: <INPUT TYPE=PASSWORD NAME="Password"><BR>
<INPUT TYPE=HIDDEN NAME="_function" VALUE="secondform">
<INPUT TYPE=SUBMIT>
</FORM>
<BODY>
```

36 Close formA.

37 Save `allforms.taf`.

For information on how to execute application files in your browser, see "Executing Application Files" on page 4.

38 Return to the browser and execute:

```
allforms.taf?_function=firstform
```

The first form page appears.



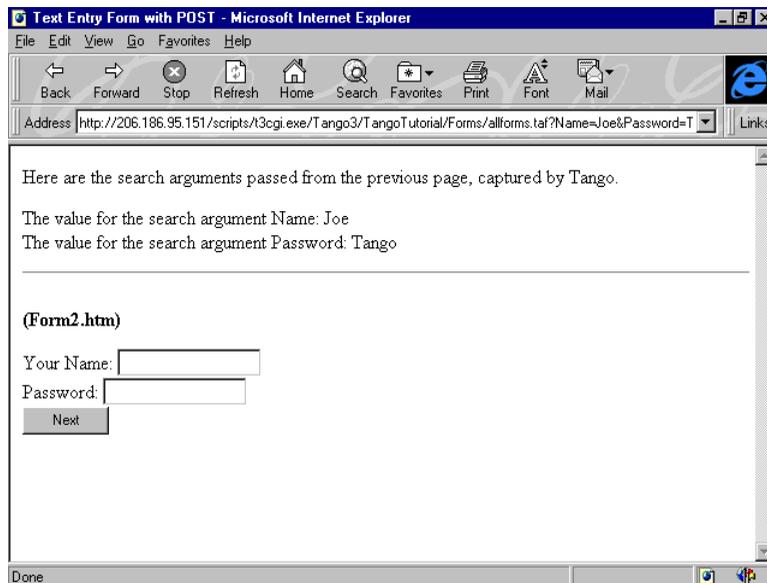
Initially, you must send variable `_function` and its value `firstform` as a search argument by manually placing it in the URL line. When you press ENTER, `allforms.taf` executes. Specifically, the first If action executes because the argument `_function` is equal to `firstform`. You assigned it that value by typing it in the URL.

39 Enter a name and password, and press ENTER.

`Allforms.taf` executes again, and the second form page appears.



The URL line contains the search arguments name and password, and their respective values that you typed on the form, as well as the hidden variable `_function` and its value `secondform`. Tango used the search argument function to determine which Results action to execute, and it used the search arguments `Name` and `Password` to display their values on that second form page.



40 Return to `allforms.taf` in Tango Editor.

41 Open the HTML editing window for formB.

42 Modify the form's action to read:

```
ACTION="allforms.taf?_function=thirdform".
```



You could also send `_function=thirdform` as a post argument by including it as a hidden field in the form, as you did in the previous steps. However, appending it to the URL is easier. Ultimately, it does not matter if you send the value for `_function` as a *search argument* or *post argument* because the If and Else If actions use the `<@ARG>` meta tag to reference the variable `_function`.

43 Close the formB editing window.**44** Open the formC editing window.**45** Modify the form's action to read:

```
ACTION="allforms.taf?_function=fourthform".
```

46 Close the formC editing window.**47** Save `allforms.taf`.**48** Execute `allforms.taf` in the browser beginning with the first form page.**49** Enter values for the form fields and click your way to the last form page.

Q. Did everything work as it did when you used four separate application files?

For information on how to execute application files in your browser, see "Executing Application Files" on page 4.

Data Sources

C

*Creating a Data Source to Use for the
Guestbook and Login Solutions*

The *data source* contains all the information Tango needs to find and access a database.

This tutorial covers how to create a data source. It talks about the database as singular, but Tango can connect with more than one database.

The following lessons make up Tutorial C:

- creating a data source
- touring the completed guestbook model.

LESSON C - 1

Creating a Tango Tutorial Data Source

Purpose

To create a data source that contains information about how to connect to a specific database.



Note Tango needs a data source to interact with a database.

Context

In Tutorial B, you built application files that did not connect with a database. In this lesson, you begin to build a *guestbook* model, which allows people to sign and search a guestbook. Your database is your guestbook, where you store guest information. To hit a database you must set up a data source that points to the database you want to use. This lesson sets up a data source through Tango.



Note You can also set up a data source using the ODBC (Open Database Connectivity) object in the Windows Control Panel.

You use the data source you create here in the rest of the tutorials, including the guestbook and login solutions.

Result

An ODBC System data source stores information about how to connect to the specified database. A System data source is visible to all users of the machine, including NT services.

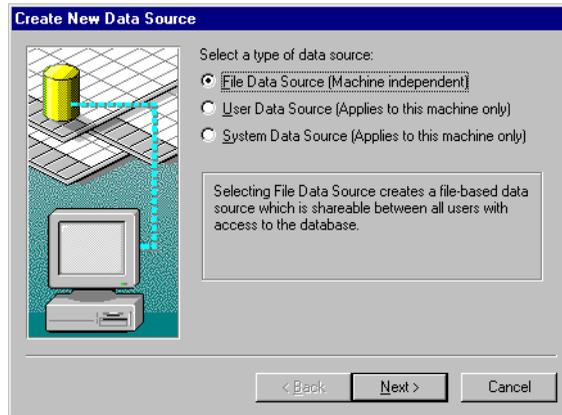
Exercise

- 1 Open Tango Editor.
- 2 Make sure the Data Sources workspace is open. You must have this workspace open to be able to choose the ODBC command in the next step.

If the Data Source workspaces is not already open, choose **Workspace** from the **View** menu, and click the Data Sources tab in the Workspace window that appears.

3 From the **DataSource** menu, choose **New**, then choose **ODBC**.

A data source wizard opens to prompt you through setting up a data source.

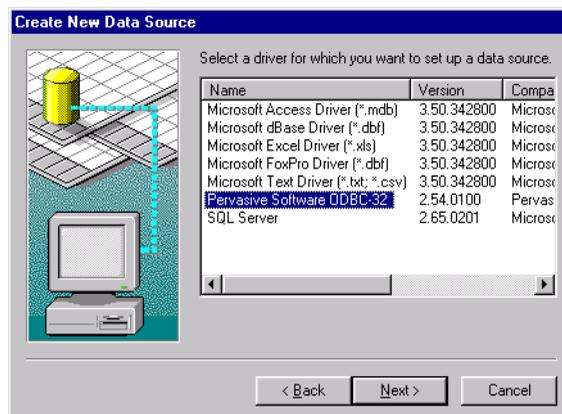


The first set of options determines the type of data source you want to create.

4 Select **System Data Source**, which allows anyone with access to your machine to also use that data source.

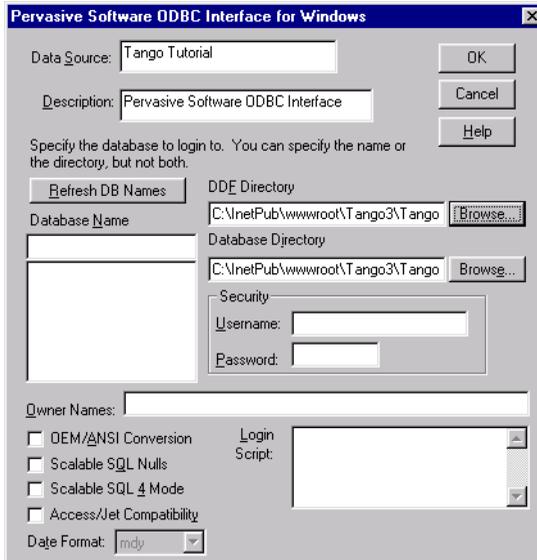
5 Choose **Next**.

6 Select a driver for your data source. Because you are using a Pervasive.SQL database, select **Pervasive Software ODBC-32**.



7 Choose **Next** and then **Finish**.

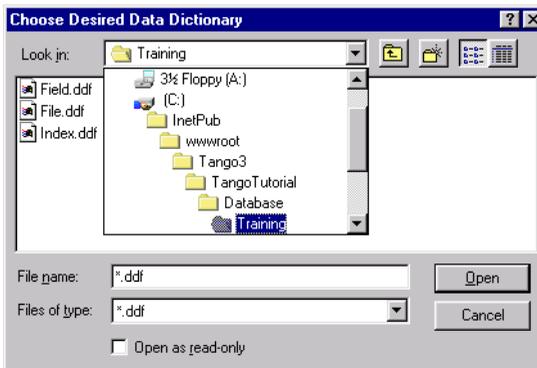
The Pervasive Software ODBC Interface for Windows dialog box opens prompting you for more information.



- 8 In the **Data Source** field, type “Tango Tutorial”.
- 9 Select the database at which you want this data source to point. Beside the **DDF Directory** field, click **Browse**, and navigate to the data dictionary files and database directory, which can be found in:

C:\InetPub\wwwroot\Tango3\TangoTutorial\Database\Training.

Remember to adjust this file path according to the Web server you are using.



- 10 Within the **Training** folder, select any one of the DDF files.

11 Click Open.

The Pervasive Software ODBC Interface for Windows dialog box reappears. The path to the Database directory appears in the **DDF Directory** and **Database Directory** fields.

12 Click OK to save your data source.



Note You have set up the Tango Tutorial data source that points to the `Training` database folder. Tango Editor and Tango Server can now hit the database. You use the Tango Tutorial data source for the rest of the tutorials.

You are now ready to tour a completed version of the guestbook installed with Tango.

LESSON C - 2

Touring the Completed Guestbook Model

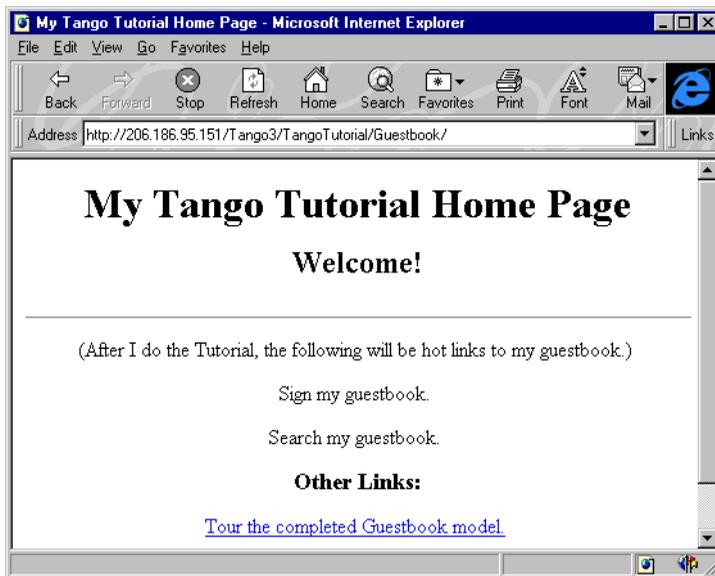
Purpose To familiarize you with the guestbook model, which gives you insight into Tango functionality.

Result A thorough understanding of the guestbook model.

Exercise 1 Open your browser and load:

`/Tango3/TangoTutorial/Guestbook/default.htm`

“My Tango Tutorial Home Page” appears.

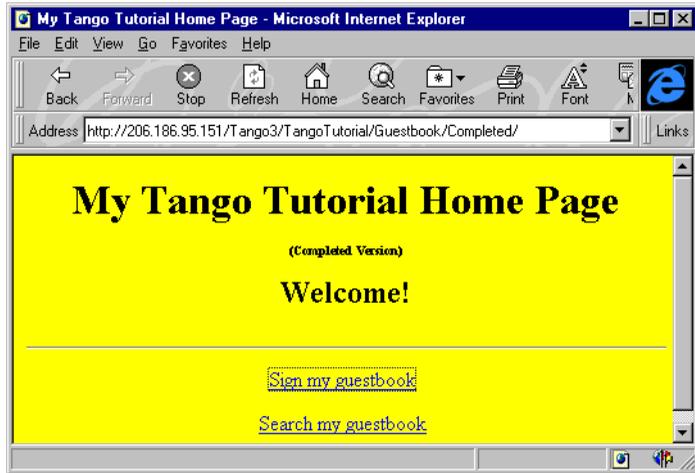


This is your modifiable home page, modifiable because in the coming lessons you will make links to your version of the guestbook files on this page.

This page contains an active hot link to a completed guestbook model.

2 Click Tour the completed Guestbook model.

This link takes you to a completed home page with active links to the guestbook.



3 Click Sign my guestbook.

You are presented with a form to enter your personal information in.

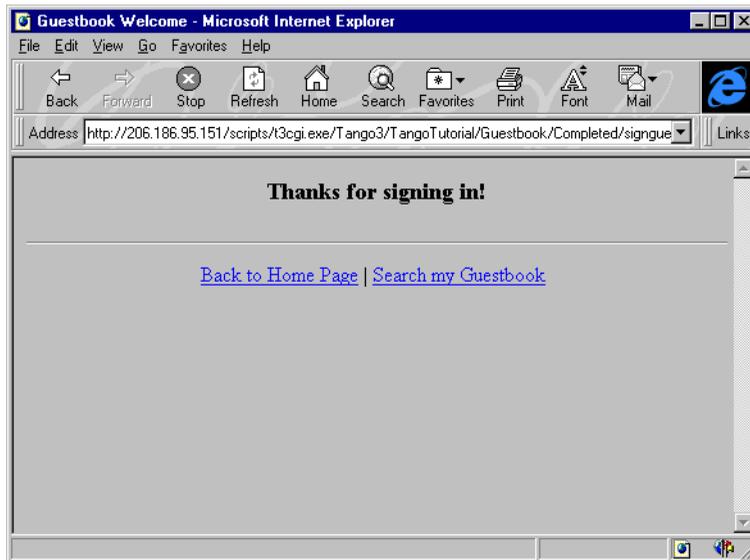


4 Enter your information in the guestbook, filling in as many fields as possible.

5 Click Sign In to enter the information.

You have now created a new guestbook record in the database.

A thank you message and a link allowing you to search the guestbook appears.



6 Click Search my Guestbook.

7 Search for the information you just entered in the guestbook.

Q. What result do you get?

8 Navigate to the search screen again.

9 Leave all fields blank, and click **Search my Guestbook**.

Q. What result do you get?

A. You see a listing of all the guests in the guestbook (database). The search is set to return all records from the database if the search fields are left empty. This is an option that you can easily change in Tango. Click one of the guests to see their full record detail.

- 10 Go back and enter another guest or two in the guestbook and then conduct searches on the guestbook.

Try different combinations—such as searching for a guest you know does not exist.

Q. What result do you get?



Tip As you move through the completed guestbook, pay particular attention to headings, field titles, button titles, and hot links. You need to be aware of these because you will build them in the coming lessons.

Sign My Guestbook

D

*Using the New Record Builder to Build a
Sign My Guestbook File*

This tutorial teaches you how to insert records into a database with Tango. It introduces the New Record Builder, as well as tools for changing presentation logic in Tango-generated Web pages.

The following lessons make up Tutorial D:

- creating a generic application file to insert records into the database
- modifying the insert page header and footer for the guestbook solution
- modifying New Record Response HTML for the guestbook
- setting required fields, field titles, and button titles for the guestbook
- linking “Sign my Guestbook” to the Tango Tutorial Home Page.

LESSON D - 1

Creating a Generic Application File to Insert Records into the Database

Purpose

To learn the basic functionality of the New Record Builder as you use it to create an application file that can insert records into a specified database.

Context

This lesson introduces how you can use application files to create entire Web sites. The guestbook section of the tutorial demonstrates how to create or modify the presentation of Tango results.

Instead of building an application file by dragging Branch and Results actions, you can use Tango builders (similar to wizards) to assemble the actions required for a search or insert into the database. You use these builders to create the guestbook solution.

Result

A Web page that presents a simple form to collect information from the user, with a **Submit** or **Save** button, that, when clicked, has Tango insert the information into the database.

Exercise



- 1 Open Tango Editor.
- 2 From the **File** menu, choose **New**, or click the New icon on the main toolbar.

A blank application file opens.

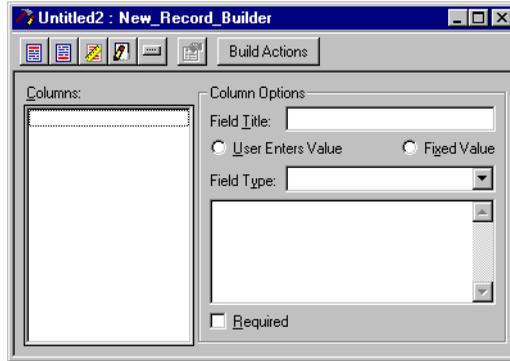
- 3 In Data Sources workspace window, expand ODBC to see the list of data sources.
- 4 Double click the Tango Tutorial data source in the list of data sources.

When prompted for a Username and Password, leave the fields empty. Click **OK** to connect to the database and to load the tables and fields.



- 5 From the Actions bar, click the New Record Builder icon and drag it into the application file.

The New Record Builder window opens.



The builder allows you to create or *generate* a series of actions that work to insert a new, user-entered record into the database.

- 6 In the Data Sources workspace expand the Tango Tutorial data source so you can view the tables in the database.



Locate the USERS table and expand it to see the columns within this table.



- 7 Drag the following fields into the **Columns** list in the New Record Builder:

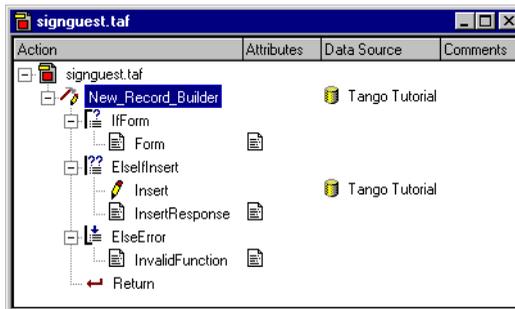
- USERID
- FIRST
- LAST
- COMPANY
- PHONE
- EMAIL.

Build Actions

- 8 Click **Build Actions**.



This generates the series of actions that insert records into the database. These actions appear in the application file window, grouped under the New Record Builder icon.



You may expand or close the group to hide or reveal the actions within it.

9 Save the application file as `signguest.taf` in the `\Tango3\TangoTutorial\Guestbook\` folder.

The `.taf` extension is automatically added so it is not necessary for you to type it after the file name.



Note If the extension does not appear with the file name and you want to display it, disable the hide extension option in Windows.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

10 Return to your browser, and execute:

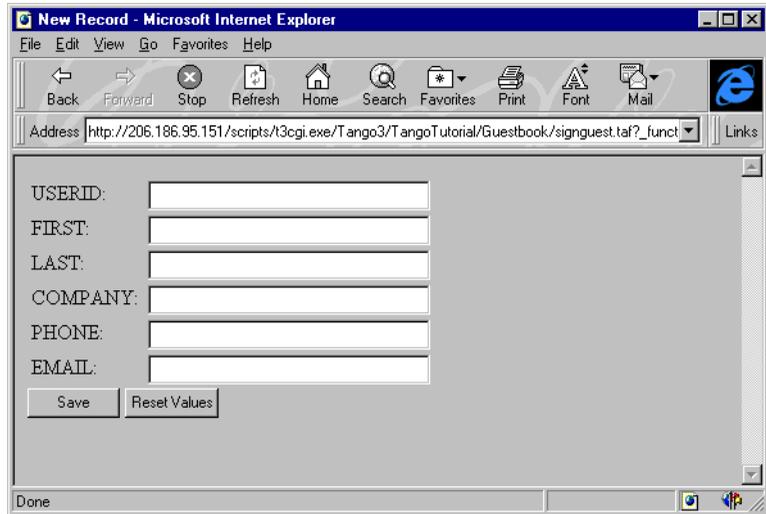
```
signguest.taf?_function=form
```



Recall from Lesson B1-1 that a question mark (?) in the URL represents the end of a file path and what follows the question mark is the search argument.

In this URL, you are passing a value of `form` for the search argument named `_function`. Tango requires the value for this search argument in order to know which part of the file to process, thereby affecting the end results in the browser. By sending `_function=form`, you are simply telling Tango to run the set of actions that produces the Sign my Guestbook form in the browser.

A form appears in your browser. It has no headings and is plain.



This form collects information from the user, or guest, in this case. Tango enters the information collected into the database in the corresponding fields.

The field names beside the white spaces are identical to the field names of the database. You change these later to make them more friendly.

11 Enter values or data for each of the form fields, and click **Save**.

The message, “The record was added successfully” appears.



Note Keep some of the data you enter here in mind (such as User ID or First Name); you can then search for it later, when you build a search function into your Web pages.



Note that after you enter a new record, the message page you see in the browser is generated by the *same* application file, `signguest.taf`. The only difference in the URL of the two pages is the search argument after the question mark, which for the latter page is `_function=insert`. When you click **Save** on the form page, you pass `insert` as the value for `_function`, thereby directing Tango to process a different set of actions in the file and produce a different Web page.

- Q.** Try executing the application file with `?_function=junk` in the URL after the path. What do you get?
- A.** You see a different Web page generated by Tango that states, “Invalid function value specified in CGI call”. This is a valid result of Tango actions. It means Tango did not know which of the two Web pages to produce—the form or the insert record response—so it executed the action that provides this message.

L E S S O N D - 2

Modifying Insert Page Header and Footer for Guestbook Solution

Purpose

To change headings in the New Record Builder that transform the simple insert record Web page to an on-line guestbook sign-in page, and to learn how to add navigational hot links.

Context

The features you set in this lesson alter the overall appearance of the Web page. The steps transform the plain insert application file to a Sign my Guestbook file. You are encouraged to add additional HTML features beyond those in this lesson if you know some.

Result

The same Web page as in the previous lesson, but with headings that make it an on-line guestbook, and navigational hot links.

Exercise

1 Open `signguest.taf` in Tango Editor.

2 Double click the New Record Builder action icon.

This opens the New Record Builder window, which generated the New Record Builder actions originally.

3 Click the Header HTML icon.

A Header HTML window opens.

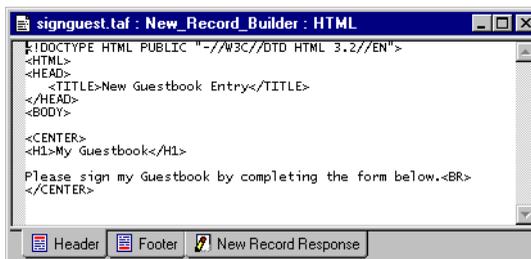


This window represents the HTML that appears above the form that Tango generates on the first Web page.

4 Change the Title to “New Guestbook Entry”.

5 Type the following HTML below the `<BODY>` tag:

```
<CENTER>
<H1>My Guestbook</H1>
Please sign my Guestbook by completing the form
below.<BR>
</CENTER>
```



- 6 Close the window.
- 7 Click the Footer icon.



This opens a Footer HTML window that represents the text and/or HTML that appears below the Tango generated form on the first Web page.

- 8 Add a horizontal rule—a line on a Web page—by typing in `<HR>`.
- 9 Press ENTER to start a new line.
- 10 Type the following to add a hot link that returns the user to the Tango Tutorial Home Page:

```
<A HREF="/Tango3/TangoTutorial/Guestbook/default.htm">Back to Home Page</A>
```

This link offers the user the choice to return to the home page without clicking **Back** in the browser.



In the HTML that you just created, the part before the text is the anchor or hot link's opening tag, and the part after the tag is the anchor's closing tag. The file in quotes is the file this link connects to or, in Tango terms, executes.

- 11 Close the Footer HTML window.
- 12 Click **Build Actions**.

The New Record Builder generates a new group of actions to replace the previous ones.
- 13 Save `signguest.taf` in the Guestbook folder.

14 Return to your Web browser and execute the `signguest.taf` by typing it in the URL.

Q. What do you see?

A. You see your new guestbook headings and instructions.
Enter another guest in the guestbook.

Q. What response do you get?

L E S S O N D - 3

Modifying New Record Response HTML for Guestbook

Purpose To change the HTML that produces the sign-in response page after a guest has signed in.

Result A message that reads, “Thanks for signing in!” after a guest signs in, and two hot links **Search my Guestbook** and **Back to Home Page** to navigate to other parts of the Web site.

Exercise

- 1 Open `signguest.taf` in Tango Editor.
- 2 Double click the New Record Builder icon in the application file.

The New Record Builder window opens.

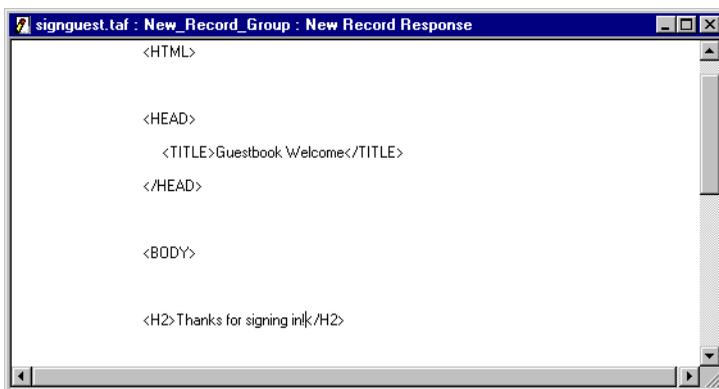
- 3 Click the New Record Response icon.



This opens the New Record Response HTML window that represents the Web page shown to the user after they successfully sign the guestbook.

- 4 Change the title to “Guestbook Welcome”.
- 5 Change the heading `<H2>` to “Thanks for signing in!”.

This is a more appropriate message in our Guestbook model than “The record was added successfully”.



Tip It is good Web interface design to offer hot links to other pages, namely to the home page or to the file that searches the Guestbook, allowing the user to search their entry.

6 Type the following HTML code:

```
<HR>

<A HREF="/Tango3/TangoTutorial/Guestbook/default.htm">Back to Home Page</A>

<A HREF="searchguest.taf?_function=form">Search my Guestbook</A>
```

This HTML inserts a line and two hot links centered beside each other on the Web page.

7 Center the items on the page by inserting an opening `<CENTER>` tag after the title line, and a closing `</CENTER>` tag after the hot links.

8 Close the window.

9 Click **Build Actions**.

The New Record Builder generates a new group of actions to replace the previous ones.

10 Save `signguest.taf` in the Guestbook folder.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

11 Return to your Web browser and execute `signguest.taf`.



Tip You may just have to refresh or reload if you had the file up in your browser already. Your new Guestbook headings and instructions appear.

12 Enter another guest in the Guestbook.

Your Guestbook response with two hot links to navigate to other pages is returned. The “Search my Guestbook” link does not work at this time because you have not yet created `searchguest.taf`.

LESSON D - 4

Setting Required Fields, Field Titles, and Button Titles for Guestbook

- Purpose** To change field titles and button titles in the New Record Builder.
- Result** The field titles—labels beside the text space in the sign-in form in the browser—changed to friendlier titles than the default field names. The titles on the two buttons that appear on the “Sign my Guestbook” page are changed from **Save** and **Reset Values** (Tango defaults) to **Sign** and **Clear**, to suit the Guestbook solution.
- Exercise**
- 1 Open the New Record Builder of `signquest.taf`.
 - 2 Click `USERS.USERID` in the list of database columns.

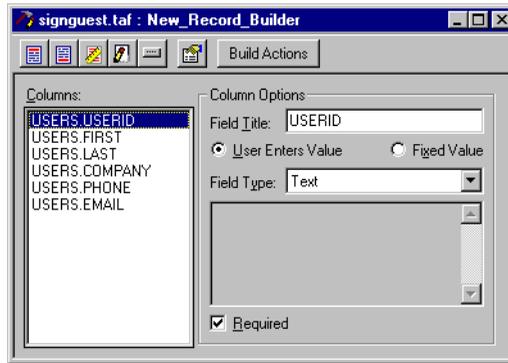


Tip Tango appends the table name to the column or field name in the **Columns** list. To the right of the **Columns** list appears the **Column Options** for the column you have selected, `USERS.USERID`.

- 3 In **Field Title**, change the default field title to “User ID”.
Spaces are allowed because this is the text the user sees in the Guestbook form in the browser. A colon is not necessary as Tango inserts one for you.

4 Click the **Required** option.

The guest is now required to fill this field in, or Tango does not enter the new guest in the database.



5 Click **USERS.FIRST**, which is the column for first name.

6 Change its **Field Title** to “First Name”, and make it a required field as well.

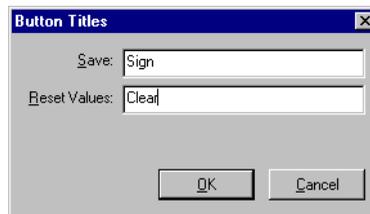
7 Do the same for the rest of the columns, changing their field titles to more friendly titles as follows:

- **USERS.LAST** to “Last Name” (required)
- **USERS.COMPANY** to “Company” (not required)
- **USERS.PHONE** to “Phone Number” (not required)
- **USERS.EMAIL** to “Email Address” (not required).



8 Click the **Button Titles** icon, or choose **Button Titles** from the **Attributes** menu.

9 Change the **Save** field to “Sign”, and change the **Reset Values** field to “Clear”.



The new button titles are more appropriate for our Guestbook solution.

10 Click **OK**.

11 Click **Build Actions**.

The New Record Builder generates a new group of actions to replace the previous ones.

12 Save `signguest.taf` in the Guestbook folder.

13 Return to your Web browser and execute the `signguest.taf` by typing it in the URL. You may have to refresh or reload if you had the file in your browser already.

Q. Do your field titles and button titles reflect the changes you made?

LESSON D - 5

Linking “Sign my Guestbook” to the Tango Tutorial Home Page

Purpose

To integrate application files with standard HTML Web pages.



Note This is a key step to building Tango-based Web site solutions.

Context

Now that you have a working `signguest.taf`, you can complete the link to it on the Tango Tutorial Home Page. You need to modify your `default.htm` file in the `Guestbook` folder. You can do this by navigating to it from the desktop and opening the file. You may also modify `default.htm` by calling it in the browser, and viewing the source.

Result

A working link, **Sign my Guestbook**, on your Tango Tutorial Home Page that when clicked executes `signguest.taf`.

- 1 Open `default.htm` and find the text, “Sign my Guestbook”.
- 2 Wrap a hot link or anchor tag around it as follows:

```
<A HREF="/Tango3/TangoTutorial/Guestbook/signguest.taf">Sign my Guestbook</A>
```

- 3 From the **File** menu, choose **Save As**.
- 4 In the Save As dialog box that appears, navigate to the `Guestbook` folder.
- 5 Save the file as `default.htm`, and click **Yes** to replace the existing one.



Tip You may minimize the file to keep it handy as you make more modifications to it in the following lessons.

- 6 Return to the browser and call up the Tango Tutorial Home Page by loading `default.htm`.

7 Test the link, **Sign my Guestbook**.



If it does not work, try refreshing or reloading the default page; you may be looking at a browser-cached copy instead of the updated default page.

If you see any parts of the HTML code on the screen, it is likely you made a typing error or forgot to close quotes, brackets, or tags. Go back to the source file and double check your syntax.

- 8 Click through the other links in the Guestbook to make sure they work properly.

Tango Builders

*Working With the Search Builder and the
New Record Builder*

This tutorial teaches you more about working with the Tango builder tools, New Record Builder and Search Builder. They are tools that generate a series of actions that perform a certain function, either inserting a new record or searching for a record and returning results.

The following lessons make up Tutorial E:

E1: Search Options

- creating a generic application file to search records in the database
- modifying the Search page header and footer for the guestbook
- modifying search operators and field titles for the guestbook
- modifying Search page formatting
- modifying the guestbook search button titles
- modifying Search page and no results HTML for the guestbook

E2: Record List Options

- modifying the record list page header and footer for the guestbook
- modifying record list page formatting for the guestbook
- setting the number of matching records returned

E3: Record Detail Options

- modifying the record detail header and footer for the guestbook
- modifying record detail page formatting

E4: Joining Database Tables

- adding search functionality using joins.

L E S S O N E 1 - 1

Creating a Generic Application File to Search Records in the Database

Purpose

To use the Search Builder to build an application file that searches for records in the database and returns results.

Result

Three Web pages:

- a page with a form asking the user to specify search criteria
- a Record List page listing matching records from the database (if any)
- a Record Detail page showing the full details of a specific record chosen by the end-user after the initial search.

Exercise

The process for using the Search Builder is the same as for using the New Record Builder.



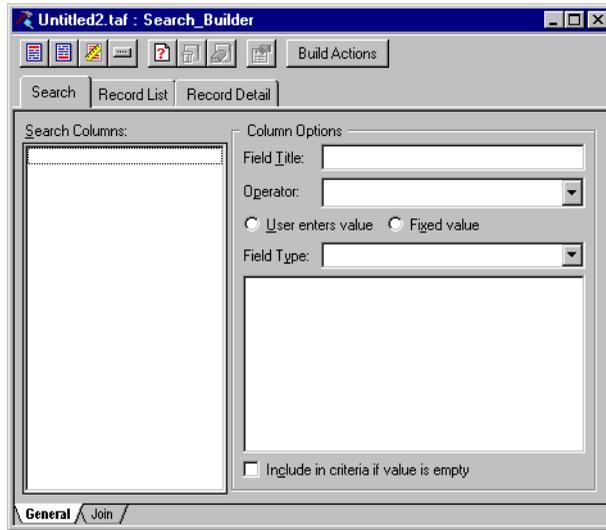
- 1 Open Tango Editor.
- 2 From the **File** menu, choose **New**, or click the New icon to open a new application file.

A blank application file opens, as does the workspace.



- 3 From the Actions bar, drag the Search Builder icon into the application file.

The Search Builder window opens.



If you are doing this lesson in order from the New Record Builder lessons, the Tango Tutorial data source remains loaded. Once the data source is loaded by Tango, it remains loaded until you exit Tango. If you stopped and returned to the lessons, load the Tango Tutorial data source by double clicking or expanding it in the Data Sources workspace. Tango connects to the database and loads its tables and fields.



Application files work using an action-based metaphor. Each action carries out a specific function and may combine with other actions to produce a Web page. Tango builders, on the other hand, use a Web page metaphor.

The Search Builder identifies three Web pages and offers you control over their look and presentation. Tango builders assist the Tango developer to generate a sequence of actions that perform a specific task. Builders are not separate from an application file; they are part of it and the actions they produce.

You are going to use the Search Builder to build actions that produce three Web pages—a Search Form page, a Record List (results) page, and a Record Detail page—when the file is executed in the browser. The Search Builder has three tabs: Search, Record List, and Record Detail. Each tab represents one page in the browser. The Search page displays a form where the user can enter search criteria. The Record List page lists the results of the search, but is only displayed if there are matching records. If there are no matching records, Tango does not display the Record List page; it displays the No Results HTML page. On the Record List page, you can make one of the fields a hot link to the Record Detail page, which displays more details about a specific matching record.

The process of moving through the three pages in the browser is called *drilling down* into the database to arrive at a specific record. In this lesson, you set up which fields the user will search on, which fields will be returned for each matching record in the Record List, and which fields will be displayed for a specific record on the Record Detail page.

-
- 4 In the Data Sources workspace, expand the Tango Tutorial data source so the tables in the database can be viewed.
 - 5 Locate the USERS table and expand it to see the fields within it. You are using the same table as in the `Sign my Guestbook` file because this is where guest records are being entered.
 - 6 Drag the following fields into the **Search Columns** list in the Search Builder:
 - UserID
 - First
 - Last
 - Company

- Phone
- Email.

You want to allow guests to search on any of these fields.

7 Click the Record List tab, and drag the following fields into the **Display Columns** list:

- UserID
- First
- Last
- Phone.



The values for these fields are displayed to the user for each matching record or guest found.

You dragged over UserID and Phone columns because it is important to show fields that can uniquely identify each guest or record. For example, if the user searched for the guest “John Smith”, there may be three matching records in the guestbook. On the Record List page, Tango would display all three John Smiths, but you would want to display some information that can assist the user in distinguishing among them. The user can then click the hot link and get the Record Detail for their John Smith of choice.

You do not necessarily want to return all the fields for each matching record in the database on the Record List page. If there were a large number of records being returned with full details (500 John Smiths, for instance), the user would experience unnecessary slow down as Tango would download all fields for all the matching records.

It is unlikely that the user wants the details of all the matching guests. The user wants full details only when they have narrowed their search to one matching record.

8 Click the Record Detail tab.



Here you may want to display all the fields for a particular guest's record. Instead of dragging into the **Display Columns** list each field individually, or even highlighting them and dragging the group over, highlight the **USERS** table and drag it over. This brings all the fields within that table into the **Display Columns** list.

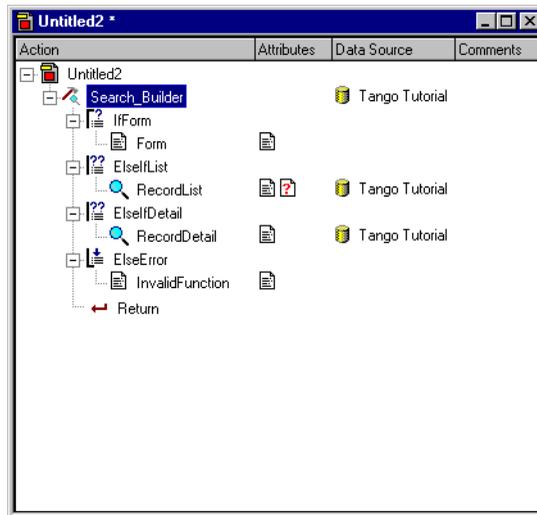
Of course, in your own Tango applications you can drag over any fields you like that are relevant to your situation. In fact, the only requirement for the Search Builder is that *any* two of the three tabs have at least one field in them.

Do nothing else in the builder at the moment. The purpose of this lesson is to build a quick application file that searches the database. When you have accomplished the functionality, you return to the builder and make adjustments or additions to suit the guestbook model.

9 Drag the **USERS** table to the **Display Columns** list in the Record Detail tab.

10 Click **Build Actions**.

This generates the series of actions that serve to search the database. These actions appear in the application file window, grouped under the Search Builder icon. You may expand or close the group to hide or reveal the actions within it.



11 Close the Search Builder.

- 12 Save the application file as `searchguest.taf` in the `\Tango3\TangoTutorial\Guestbook\` folder.

The `.taf` extension is automatically added so it is not necessary for you to type it after the file name.



Note If the extension does not appear with the file name and you want to display it, disable the hide extension option in Windows.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

- 13 Return to your browser and execute `searchguest.taf`.



A form appears in your browser. It has no headings and is plain. This form collects the search criteria from the user, in this case the guest, and returns results of the search. The field titles beside the white spaces are the friendly titles you typed in the Sign my Guestbook lessons. Tango remembers them from when you changed them earlier.

- 14 Enter some search criteria—perhaps your name—in the appropriate form field

- 15 Click **Find**.



The default search operator is **Begins with** for text fields and **equal to** for numeric fields. If there is a matching record in the database, Tango displays the record, or records if there is more than one match. This page corresponds to the Record List page. The first field or column value is hot linked, allowing you to click it and receive the Record Detail page containing all the available information for the guest you searched.



Note There are no headings on the three pages. You add these in the next few lessons. Note also the last page is a dead page, meaning it has no links or buttons that allow you to navigate to another page. When you return to the Search Builder, you add hot links to your home page.

16 Enter search criteria that you know produces no matching records, such as “zzz”.

Q. What response do you get?

A. A message stating “No matching records were found” appears. You change this message shortly to something more friendly but the current concern is functionality.

Note also that all four pages you viewed in the browser were generated by the same application file, `searchguest.taf`. The only difference in the URL of the pages is the value of the search argument named `_function`, found after the question mark.

Run through the previous steps again and pay particular attention to how the value for `_function` changes from form, search, and detail. Each different value for `_function` results in Tango processing a different set of actions in the file, and produces a different Web page from the first. Try executing the Tango file with `?_function=junk` in the URL after the path.

Q. What do you get?

A. You see a different Web page generated by Tango that states: “Invalid function value specified in CGI call”. This is a valid result of Tango actions. It means that Tango did not know which Web page to produce—the Search form, the search results, or the detail—so it ended up at the action that provides this message. In the next lesson, you return to Tango Editor and turn this generic search record file into a Search my Guestbook file.

LESSON E 1 - 2

Modifying Search Page Header and Footer for Guestbook

Purpose To learn the features of the Search Builder and what you must change so that the application file fits the guestbook scenario.

Context The features you set in this lesson alter the overall appearance of the Web page. The steps transform the plain search application file to a Search my Guestbook file. You are encouraged to add additional HTML features beyond those in this lesson, if you know them.



Note You may want to tour the guestbook again to refresh your memory on what the Search my Guestbook section looks like.

Result Three Web pages resulting from the Search Builder you created in the previous lesson, but with headings and terminology that fit the guestbook scenario. They are also linked to the Tango Tutorial Home Page.

Exercise 1 Open `default.htm` in your text editor and find the text “Search my Guestbook”.

2 Wrap a hotlink or anchor tag around it as follows:

```
<A HREF="/Tango3/TangoTutorial/Guestbook/searchguest.taf">Sign my Guestbook</A>
```

3 Save `default.htm`. “Sign my Guestbook” appears hot on the Home Page.

4 Open the `searchguest.taf` file in Tango Editor (if it is not open already).

5 Double click the Search Builder icon.

This opens the Search Builder window, which generated the Search Builder actions originally.

6 Click the Search tab.



7 Click the Header HTML icon.

A Header HTML editing window opens. This window contains the text and/or HTML that appears above the search form that Tango generates on the search Web page in the browser.

8 Change the title to “Guestbook Search”, and type in the following HTML:

```
<CENTER>
<H1>Search my Guestbook</H1>
Which guest or guests would you like to search for?
</CENTER>
```

9 Close the Header HTML editing window.



10 Click the Footer HTML icon.

This opens a Footer HTML window that contains the text and/or HTML that appears below the Tango generated form on the search Web page.

11 Add a horizontal rule by typing `<HR>`.

12 Press `ENTER` to start a new line.

13 Type the following to add a hot link that returns the user to the Tango Tutorial Home Page:

```
<A HREF="/Tango3/TangoTutorial/Guestbook/
default.htm">Back to Home Page</A>
```

This link offers the user the choice to return to the home page without clicking **Back** in the browser, if they decide not to search the guestbook.

14 Close the Footer Results HTML editing window.

You have completed the header and footer changes, and hot link additions to the first page of the guestbook search.

15 Click **Build Actions**.

The Search Builder generates a new group of actions to replace the previous ones.

16 Save `searchguest.taf` in the Guestbook folder.

17 Return to your Web browser and execute the `searchguest.taf` by typing it in the URL.

The search form appears, with your new guestbook headings and instructions.

18 Click the links to make sure they work.

LESSON E 1 - 3

Modifying Search Operators and Field Titles for Guestbook

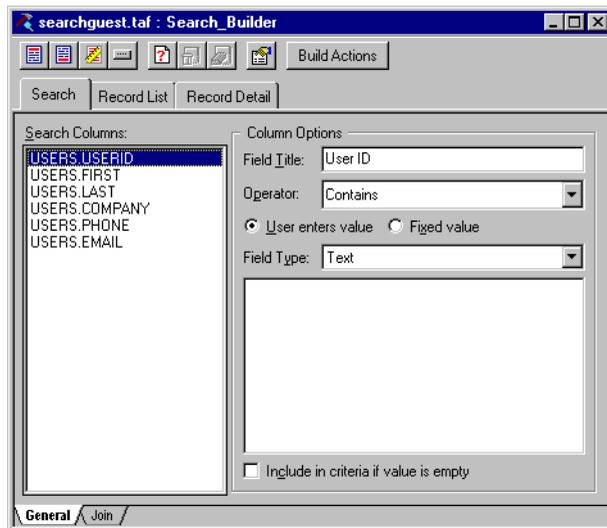
- Purpose** To manipulate the buttons and appearance of your pages.
- Context** Each column you dragged into the **Search Columns** list has options associated with it. These options can be viewed for each column when the column is selected in the **Search Columns** list. This lesson is concerned with the **Field Title** and **Operator** options. You learn more about **Field Type** in later lessons.
- Result** A Search my Guestbook page that has friendly field titles and functions as a more specific search.
- Exercise**
- 1 Return to `searchguest.taf` in Tango Editor. Make sure you are on the Search tab.
 - 2 Click the first column, `USERS.USERID`.
This column's options are displayed in the **Column Options** list.
 - 3 Change **Field Title** to "User ID".
This is a more friendly field title than the database name of the column, which is the default field title.



You may notice the **Field Title** has already been changed to User ID. This is because Tango remembered the change to the field title for this column that you made in the New Record Builder lessons.

The next option is **Oper**. This is the operator Tango uses for the particular column in the database when executing the search. For variable text fields, the Search Builder defaults the operator to **Begins with**. For numeric fields, the default operator is =.

Click the **Oper**. field and a drop-down list of operators appears that are appropriate for the field type of the currently selected column. For USERID, select **Contains**.



This makes the search on the column USERID encompassing. For instance, if the user enters the letter “a” in the search criteria for User ID, Tango searches the column in the database for anything that *contains* the letter “a”. If the default operator **Begins with** is used, Tango searches for User ID’s that *begin with* the letter “a”. Hence, the **Contains** search returns “apple” and “pear”; the **Begins with** search returns just “apple”.

- 4 Change the field titles for the remaining columns as follows, if necessary. You must click each column individually to change its field title.
 - USERS.FIRST becomes “First Name”
 - USERS.LAST becomes “Last Name”
 - USERS.COMPANY becomes “Company”
 - USERS.PHONE becomes “Phone”
 - USERS.EMAIL becomes “E-mail”
- 5 Click the columns individually again, and change each column’s operator to **Contains**.
- 6 Click **Build Actions**.
- 7 Save and replace `searchguest.taf`.
- 8 Return to your browser and check your work.
 - Q. Does the search work as you expected, with all the operators set to **Contains**?

LESSON E 1 - 4

Modifying Search Page Formatting

Purpose

To set the page format of the Search page.

Context

Tango displays the form of the Search page in HTML table format. In the Search Builder, you have control over the table settings, specifically border width and color, background color, and cell spacing and padding.

Result

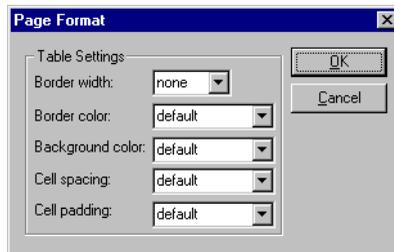
To experiment with different table settings for the “Search my Guestbook” page and see the result in the browser.

Exercise

- 1 Return to the `searchguest.taf` in Tango Editor.
- 2 Double click the Search Builder icon to open the Search Builder window, if it is not already open.
- 3 Click the Search tab.
- 4 Click the Page Format icon, or from the **Attributes** menu, choose **Page Format**.



The Page Format dialog box opens, presenting you with a list of table settings.





The options are typical for a table in HTML code, and are explained more fully in most HTML reference books.

Border width sets the width of the border around the table and around each of the cells in the table. The default setting is **none**, so no border is displayed in the browser.

Border color sets the color of the border. If no border is displayed you need not set this to any particular color. The default setting is a special value indicating that the setting will be omitted from the HTML generated by Tango, causing the browser default to be used.

Background color sets the background color of the table only. This differs from the background color of the entire Web page, identified by the tag in HTML. You may set the background color of the entire page separately, in the `<HEADER>` HTML attribute. **Default** in the Table Setting refers to the browsers' default color.

Cell spacing sets the spacing between cells.

Cell padding sets the width of the space between data in a cell and the border of a cell.

Change **Border width** to "3".

-
- 5 Change **Border color** to a color of your choice.
 - 6 Change **Background color** to a color of your choice.
 - 7 Change **Cell spacing** to "5".
 - 8 Change **Cell padding** to "4".



Note These values are arbitrary and demonstrate the varied display possibilities of these settings.

- 9 Save your changes.
- 10 Click **Build Actions** to generate the revised search actions in the application file window.
- 11 Save and replace `searchguest.taf`.
- 12 Return to your browser and reload or refresh `searchguest.taf`.

Your new table settings are displayed on the Search my Guestbook form page.

LESSON E 1 - 5

Modifying Guestbook Search Button Titles**Purpose**

To change the labels on buttons in the Search Builder.

Context

This lesson teaches you how to make more changes to the look of your Web page, specifically the button titles on your Search page.

Result

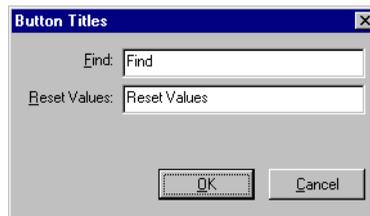
Button titles that read **Search Guestbook** (begins search for guests) and **Clear** (clears the search criteria fields if the user wants to enter or re-enter search criteria).

Exercise

- 1 Return to `searchguest.taf` in Tango Editor.
- 2 Double click the Search Builder icon to open the Search Builder, if it is not open already.
- 3 Click the Search tab.
- 4 Click the Button Titles icon, or from the **Attributes** menu, choose **Button Titles**.



The Button Titles dialog box appears.



Tango inserts two buttons on your search form page—**Find** activates the search, and **Reset Values** clears the fields of any criteria, before **Find** is clicked. You are able to change the titles of these buttons in the Button Titles dialog box.

- 5 Change “Find” to “Search Guestbook”.
- 6 Change “Reset Values” to “Clear”.

The new button titles are more suited to the Search my Guestbook model.

- 7 Click **Build Actions**.



Button titles can be changed in this manner on the Search tab and the Record Detail tab, but not on the Record List tab. The Record List is intended to display a list of matching records, with one of the field values set as a hot link to the Record Detail. It would not be practical to have a different Record Detail button for each matching record. Hot links are more appropriate.

Buttons in the builder also vary depending upon which tab you have selected. For example, Record Detail has **Save**, **Reset Values**, and **Delete** buttons, but not a **Find** button. The Record Detail buttons reflect the action options available to the end-user when on this page in the browser.

- 8 Save `searchguest.taf` and return to your browser to view the new button titles.

LESSON E 1 - 6

Modifying No Results HTML for Guestbook**Purpose**

To change the No Results HTML and Search page.

Context

The Search Builder offers a window that holds an appropriate HTML message to the user, if there are no matching records to their search.

Result

A No Results HTML page that fits the guestbook model.

Exercise

- 1 Return to `searchguest.taf` in Tango Editor.
- 2 Double click the Search Builder icon to open the Search Builder, if it is not open already.
- 3 In the search window of the Search Builder, click the No Results HTML icon.

The No Results HTML editing window opens. This window represents the entire page that is shown to the user in the event that the search produces no matching records. There are no header or footer buttons because this editing window represents the entire Web page.

- 4 Change the title to “No Matching Guests”.
- 5 Under the `<BODY>` tag, add the following heading, and two hot links that return the user to the search form, to try again, or to your Tango Tutorial Home Page:

```
<CENTER>
```

```
<H1>Sorry, no matching guests.</H1>
```

```
No guests were found that matched your search
criteria.
```

```
<BR>
```

```
<A HREF="searchguest.taf?_function=form">Search for
another guest</A> | <A HREF="default.htm">Return to
Home Page</A>
```

```
</CENTER>
```

Again, you are offering hot links here because your aim is to never have the user hit the **Back** button in the browser. Always offer the user full navigation options.



Note The vertical dash serves as a delimiter between the two links, which appear side by side in the browser. You may use any character as a separator, but the vertical dash is commonly used on Web pages.

- 6 Close the No Results HTML editing window.
- 7 Click **Build Actions** to generate a new set of actions that reflect your changes.
- 8 Save `searchguest.taf`.
- 9 Return to the browser to check your work. Search for a guest you know is *not* in the guestbook. The No Results HTML page appears with your relevant guestbook headings and hot links. Test the hot links.

LESSON E 2 - 1

Modifying Record List Page Header and Footer for Guestbook

Purpose To modify the appearance of the Record List page displaying the results of a successful database search.

Context The Search Builder offers you considerable page formatting options for the Record List tab or page. In the Header and Footer HTML windows, you can add any text or HTML code that you want to appear above and below the Record List, including body color, titles, headings, page alignment, and hot links, for example.

Result A Record List page with centered guestbook headings, white background color instead of the browser default, and guestbook hot links.

Exercise

- 1 In Tango Editor, open the Search Builder for `searchguest.taf`.
- 2 Click the Record List tab.
- 3 Click the Header HTML icon.



The Header HTML editing window opens. This window contains the text and/or HTML that appears above the guestbook search results on the Record List Web page in the browser.

- 4 Change the title to “Guestbook Search Results”.
- 5 Change the `<BODY>` tag to:

```
<BODY BGCOLOR=#FFFFFF ALIGN=CENTER>
```



Background color is a parameter or option of the `BODY` section of the Web page, as opposed to the `HEAD` section of the Web page. The body tag itself is optional; you do not need it to display your Web page successfully. However, if you are changing a parameter of the entire body, such as background color or text alignment, you do need to include it. The color `#FFFFFF` is white. Everything on the Web page is centered because `ALIGN=CENTER`.

The body tag requires an opening tag, `<BODY>` and a closing tag, `</BODY>`. The closing tag must come at the end of what you want to include in the body. Hence, you must have a closing body tag in your Footer HTML window.

6 Type the following HTML below the body line.

```
<H1>Guestbook Search Results</H1>
Your search of the Guestbook has produced the
following matching records.<BR>
Click the hot link to see the guest's full Record
Detail.<BR>
```

7 Close the Header HTML editing window.



8 While still on the Record List tab, click the Footer HTML icon.

This opens a Footer HTML editing window that contains the text and/or HTML that appears below the guestbook search results in the browser.

9 Add a horizontal rule by typing in `<HR>`, and press `ENTER` to start on a new line.

10 Add two hot links: a link back to the guestbook search form and a link back to the Tango Tutorial Home Page.

```
<A HREF="searchguest.taf?_function=form">Search for
another guest</A>
<A HREF="default.htm">Back to Home Page</A>
```



The search link offers the user a chance to conduct another search if they decide not to check a matching guest's detail. The home page link offers the user an exit from the search process altogether without clicking **Back** in the browser.

11 Close the Footer Results HTML editing window.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

12 Click Build Actions.

13 Return to your browser and execute:

`searchguest.taf?_function=form.`

14 Leaving the fields blank, click **Search Guestbook to return all the records in the database. The Record List page appears with your new guestbook headings and hot links.**

LESSON E 2 - 2

Modifying Record List Page Formatting for Guestbook

- Purpose** To set the page format of the Record List page.
- Context** The Search Builder displays database search results in table format on the Record List page, with a border width default of “1”. You can change any of the table settings for the Record List in the Page Format dialog box.
- Result** Matching records are displayed in a table with a border width of “2”, cell spacing of “2”, cell padding of “1”, and border and background colors of your choice. These table settings vary from the Tango default settings to help you gain an understanding of their Web page end result. The background color of the page remains white regardless of the background color of the table because you specified white for the body background color in the previous lesson.

Exercise

- 1 Return to the `searchguest.taf` in Tango Editor.
- 2 Double click the Search Builder icon to open the Search Builder window, if it is not open already.
- 3 Click the Record List tab.
- 4 Click the Page Format icon, or from the **Attributes** menu, choose **Page Format**.



The Page Format dialog box opens, presenting you with the same list of table settings as for the Search page.



For a summary of each of the table settings, see “Modifying Search Page Formatting” on page 99. The default table settings on the Record List page are the same as on the Search page, except for border width. The border widths on the Search page and the Record Detail page are set to “0”; on the Record List page, border width is set to “1”. Often lists of records are easier to read and understand when displayed in a visible table.

- 5 Change **Border width** to “2”.
- 6 Change **Border color** to a color of your choice.
- 7 Change **Background color** to a color of your choice.
- 8 Change **Cell spacing** to “2”.
- 9 Change **Cell padding** to “1”.



Note The above values are arbitrary to demonstrate the varied display possibilities of these settings.

- 10 Save your changes.
- 11 Click **Build Actions** to generate the revised Search Builder actions in the application file window.
- 12 Save and replace `searchguest.taf`.
- 13 Return to your browser and reload or refresh `searchguest.taf?_function=form`.
- 14 Leave all the form fields empty and click **Search Guestbook**.
All the records in the database are returned and displayed on the Record List page in a table according to your table settings.

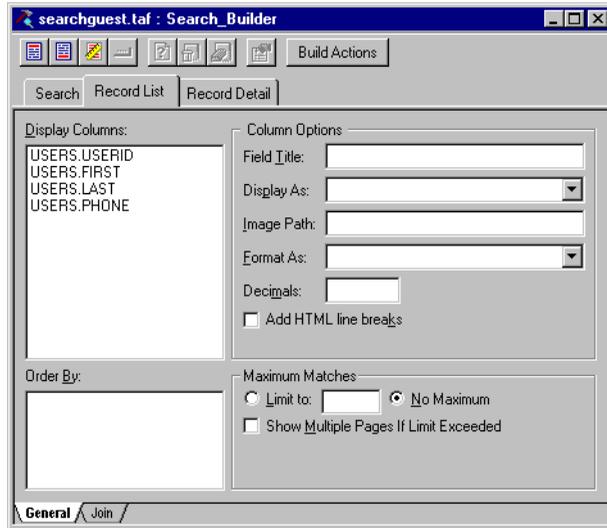
LESSON E 2 - 3

Setting the Number of Matching Records Returned

- Purpose** To set the number of matching records to return on the Record List page.
- Context** In the Search Builder, you can set the number of matching records to return to the user at one time. In some cases, the list of matching records could be lengthy. If you searched for the last name “Smith” in a phone directory database, hundreds of matching records would be found. To provide them all to the user at one time would impose an unreasonably long delay and may even crash the machine, if memory is a problem.
- In the Search Builder, you can set a reasonable number of records to return to the user, as well as decide whether to offer **Next** and **Previous** buttons.
- Result** A maximum of three matching records are displayed on the Record List page, with buttons available to display the next three or previous three records, when applicable.
- Exercise**
- 1 Return to the `searchguest.taf` in Tango Editor.
 - 2 Double click the Search Builder icon to open the Search Builder window, if it is not open already.

3 Click the Record List tab.

To the right of **Display Columns** is the **Maximum Matches** settings area.



You have three options available: 1) a set maximum number of matches with no **Next/Previous Matches** buttons available if the limit is exceeded; 2) a set maximum number of matches with **Next/Previous** buttons available if the number is exceeded; and 3) no limit to the number of matching records returned on the Record List page.

The default builder setting limits the maximum matches returned to "100". **Show Multiple Pages If Limit Exceeded** is not enabled, so the user is not presented with **Next/Previous** buttons in the event there are more than 100 matches.

4 Click the **Limit to** field, and change "100" to "3".



You are setting the number of matches returned on the Record List page to the unusually small number of three to demonstrate the **Next/Previous** buttons. There is only a small number of guest records in the database, so the matching records number must be small to force the displaying of the **Next/Previous** buttons.

5 Click Show Multiple Pages If Limit Exceeded.

This displays the **Next/Previous** buttons when the number of records returned from a search exceeds three.



Tip If you are building a guestbook and do not anticipate an enormous number of guest records, you may set the **Maximum Matches to No Maximum**.

6 Click Build Actions to generate the revised Search Builder actions in the application file window.

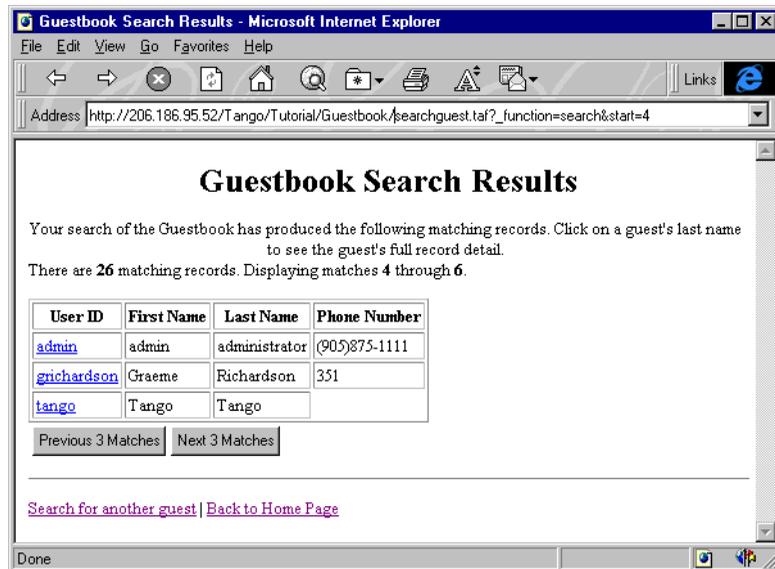
7 Save and replace `searchguest.taf`.

8 Return to your browser and reload or refresh `searchguest.taf?_function=form`.

9 Leave all the form fields empty and click Search Guestbook.

Only three records in the database are returned and displayed on the Record List page. A **Next 3 Matches** button is displayed.

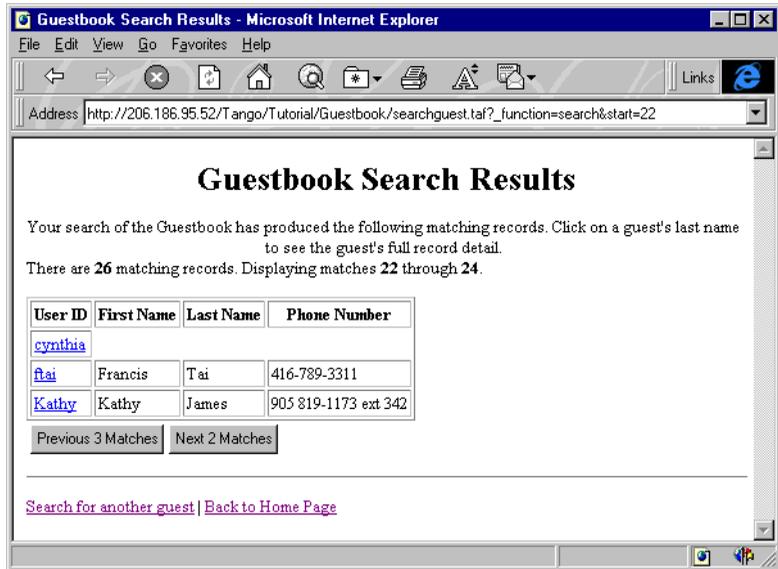
10 Click Next 3 Matches.



The next three matches are displayed, with both a **Next 3 Matches** button and now a **Previous 3 Matches** button.

11 Click either of the Next/Previous buttons to get a feel for how the user can browse through matching records with little delay.

12 Click **Next 3 Matches** until you come to the last page of matching records.



- Q.** Did you notice anything about the **Next** button?
- A.** If the number of matching records of your current search does not divide evenly by three, the **Next/Previous** buttons are adjusted accordingly on the second last page of the Record List display. For example, if only two more matching records are left to display, the **Next** button reads, **Next 2 Matches** instead of **Next 3 Matches**.

LESSON E 3 - 1

Modifying Record Detail Page Header and Footer for Guestbook

Purpose To modify the appearance of the Record Detail page displaying only one specific record from the database.

Context The Record Detail tab represents the last page in the process of drilling down into the database to display a specific record, the record's detail. In the guestbook solution, the Record Detail page displays all the records for a specific guest in the database. The Search Builder offers you the same page formatting options for the Record Detail tab or page as the Search and Record List tabs. You need to modify the Header HTML and Footer HTML so the Record Detail page suits the guestbook model you are building.

Result A Record Detail page with centered guestbook headings, white background color instead of the browser default, and guestbook hot links.

Exercise

- 1 In Tango Editor, double click the Search Builder icon in `searchguest.taf` to open the Search Builder window.
- 2 Click the Record Detail tab.
- 3 Click the Header HTML icon.



The Header HTML editing window opens. This window contains the text and/or HTML that appears above the guestbook detail search results on the Record Detail Web page in the browser.

- 4 Change the title between the `<TITLE>` and `</TITLE>` tags to "Guestbook Record Detail".
- 5 Change the body tag to the following:

```
<BODY BGCOLOR=WHITE> <CENTER>
```

6 Type in the following HTML below the body tag:

```
<H1>Guest's Record Detail</H1>
```

```
Here is the full Record Detail of the guest you  
chose from the search results.
```

```
</CENTER>
```



Note The `<BODY>` tag is opened in the Header, but closed in the Footer because there is HTML in the Footer that is a part of the body, and because between the header and the footer, Tango displays a database record that is also part of the same body.

7 Close the Header HTML editing window.

8 While still on the Record Detail tab, click the Footer HTML icon.

This opens a Footer HTML editing window that contains the text and/or HTML that appears below the guestbook Record Detail in the browser.

9 Add a horizontal rule by typing in `<HR>` at the top of the page.

10 On the next line, add the following hot links.

```
<A HREF="searchguest.taf?_function=form">Search for  
another guest</A> | <A HREF="/Tango3/TangoTutorial/  
Guestbook/default.htm">Back to Home Page</A>
```



You are adding two hot links. The link back to the guestbook search form offers the user a chance to conduct another search of the guestbook. The link back to the Tango Tutorial Home Page offers the user an exit from the search process altogether without clicking **Back** in the browser.

It is not possible in the Search Builder to make a *working* link that returns the user to the Record List page, listing all the matching records for their search. This is because simply linking to the Record List's URL is not enough; the search criteria/values must be passed to Tango as well, which then needs to conduct another search on the Tango Tutorial database.

11 Close the Footer Results HTML editing window.

12 Return to your browser and execute
`searchguest.taf?_function=form`.

For information on how to execute application files in your browser, see "Executing Application Files" on page 4.

- 13** Leaving the fields blank, click **Search Guestbook** to return all the records in the database.
- 14** Click any guest to view their full record on the Record Detail page.
 - Q.** Do you see the changes to the Record Detail page?
 - A.** Your browser displays the guest's Record Detail in the proper context of the guestbook solution.

LESSON E 3 - 2

Modifying Record Detail Page Formatting

Purpose

To set the page format or layout of the Record Detail page.

Context

The Search Builder displays the one record on the Record Detail page in a table; you can edit the table specifications under the **Page Format** option. You can also edit how each column's data is displayed within the table under the **Column Options** section on the Record Detail tab. This lesson begins by modifying the **Column Options**.

Result

A Record Detail page formatted to your specifications.

Exercise

- 1 Return to `searchguest.taf` in Tango Editor, and double click the Search Builder icon to open the Search Builder window, if it is not open already.
- 2 Click the Record Detail tab.
- 3 Identify the **Column Options** section to the right of the **Display Columns** list.

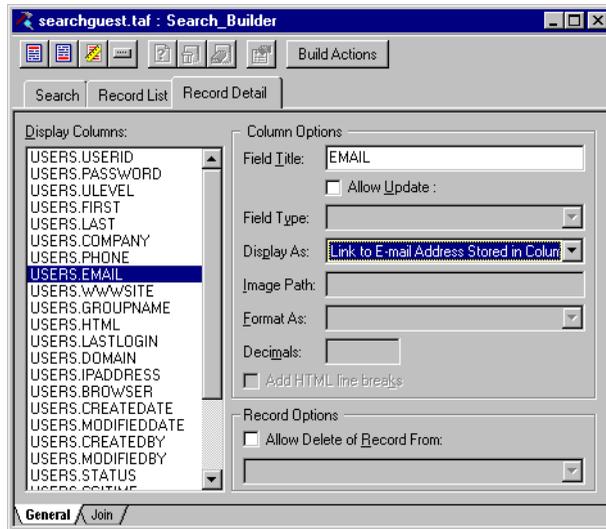


The options available are the same as those on the Record List tab, except for the addition of the **Allow Update** checkbox and its corresponding **Field Type** option when **Allow Update** is enabled. The **Allow Update** option is discussed more fully in a later lesson. All the options work as they do on the Record List tab. Currently, keep all the default settings for each of the columns except the EMAIL column.

- 4 Click USERS.EMAIL in the **Display Columns** list.

The **Column Options** settings shown are those for the EMAIL column only.

- 5 Click the **Display As** option and select **Link to E-mail Address Stored in Column** from the drop-down list.



Tango displays the EMAIL column as a hot link which, when clicked by the user, opens the browser's mail program and places the e-mail address in the **To:** line. The HTML code generated appears as follows:

```
<A HREF="mailto:(e-mail address)">
(e-mail address)</A>
```

e-mail address represents the e-mail address stored in the database. When Tango displays this column on the Record Detail page, it wraps the HTML anchor tags around the e-mail address coming out of the database.



Note There is no **Link to Record Detail** option in the **Display As** drop-down list because these options are for columns being displayed on the Record Detail page. The user would already be on the Record Detail page.



For more information on table settings, see “Modifying Search Page Formatting” on page 98.

- 6 To change the table display settings, click the Page Format icon, or from the **Attributes** menu, choose **Page Format**.

The Page Format dialog box opens, presenting you with the same list of table settings as on the Record List page. The default table settings are the same as those for the Search page.

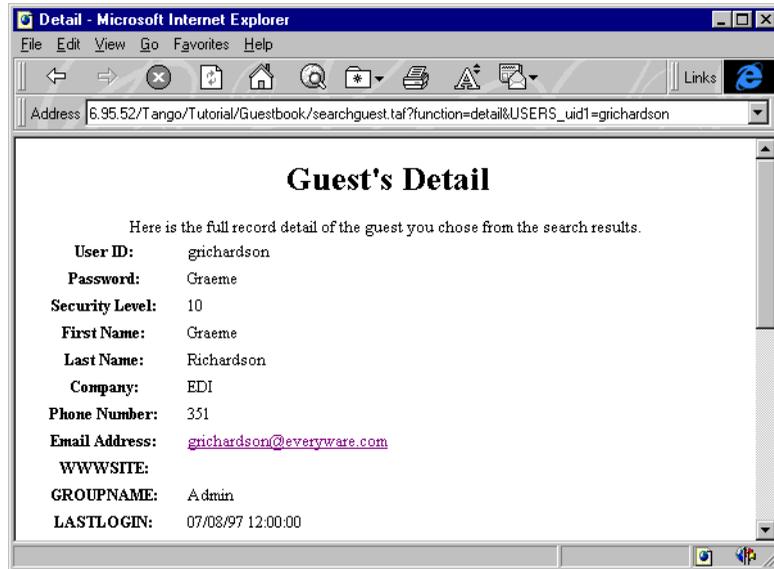
- 7 Change **Border width** to “2”.
- 8 Change **Border color** to a color of your choice, but preferably the same color as you chose on the Record List page to maintain consistency in your guestbook solution.
- 9 Change **Background color** to a color of your choice. Again, choose preferably the same color as you chose for the Record List page's table.
- 10 Change **Cell spacing** to “5”.
- 11 Change **Cell padding** to “4”.



Note These values were arbitrarily chosen in the Record List page lesson to demonstrate the varied possibilities of the table settings. You are using the same setting values to maintain some display consistency on your Web pages. Consistent designs in your Web site make for a visually more appealing Web site for the user.

- 12 Save your changes to the table settings.
- 13 Click **Build Actions** to generate the revised Search Builder actions in the application file window.
- 14 Save and replace `searchguest.taf`.
- 15 Return to your browser and reload or refresh the Search my Guestbook form page.
- 16 Leave all the form fields empty and click **Search Guestbook**.

- 17 From the list of guests, choose one to arrive at the Record Detail page for that guest.



Note The guest's Record Detail appears in a table of the same display options as the table that displays the list of guests on the previous Web page. Notice, as well, that the e-mail address for the guest is a hot link.

- 18 Click the guest's e-mail address.
 - A. Does your e-mail software open? What is placed automatically in the **To:** line?

LESSON E 4 - 1

Adding Search Functionality Using Joins**Purpose**

To incorporate the Joins function into your guestbook search, so users can search using values from more than one database table.

Context

To this point, your guestbook solution allows searches of the USERS table, and returns records from only this table. Using the *Joins* function of Tango, you can relate the USERS table to the Userlevel table, within the Tango Tutorial database. Joining these tables enables a user to conduct a search of the guestbook involving more than one table.

Result

A record list page that displays search results from the USERS table as well as the Userlevel number and description column information from the Userlevel table.

Exercise

- 1 In the Data Sources Workspace window, under the Tango Tutorial data source, double click the Userlevel table to display its columns.
- 2 With the `searchguest.taf` Search Builder open, click the Record List tab.



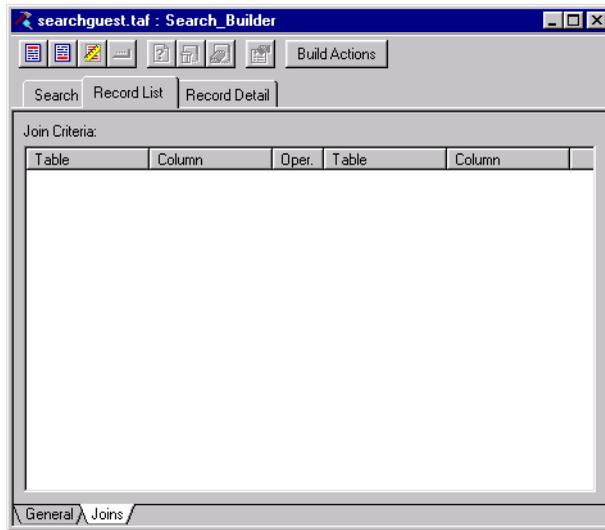
For more information on Joins, see “Joining Database Tables” in Chapter 18 of the *User’s Guide*.

To enable `searchguest.taf` to search both tables when conducting a search, instead of just hitting the USERS table, you must define a join. The join tells the database how these tables are related.

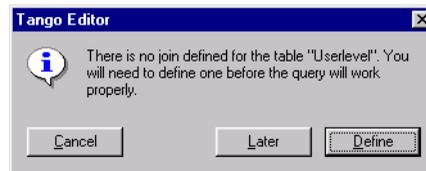
In this case, the USERS and Userlevel tables have related columns, ULEVEL and UlevelID. You relate the two columns on the Joins tab. As a result, a search of the guestbook not only returns the UserID, names and phone numbers of each guest; it can also return the Ulevel information contained in the Userlevel table that relates to that guest in the USERS table.

When you define a join on the Search page, that join will be automatically defined on the Record List page, and vice versa.

The Joins window appears.



If you add a column from a different table to a Search Builder or Search action before defining a join, Tango displays the following message:



To go to the Joins tab and define the join, click **Define**.

- 4 From the Data Source window, drag the Userlevel table into the Joins tab.

A join definition appears. By default, the Join definition shows the table you dragged from the data source window, the first table in the data source, and the first column in this table.

Join Criteria:					
Table	Column	Oper.	Table	Column	
Userlevel	ULevelD	=	Groupname	GroupKey	

- 5 Select the entry field under the right **Table** column by doing one of the following:
 - Click the field to show the drop-down list arrow, then click the arrow.
 - CONTROL click the field and choose **Edit** from the contextual menu.

Table
Groupname
Groupname
Userlevel
USERS

The field entry changes to a drop-down list box so you can choose a different entry.

- 6 Change the table from “Groupname” to “USERS”.
- 7 Select the right **Column** entry field by clicking twice in the field, and change the “Groupkey” column to “ULEVEL”.



In order to have information from the joined table appear on either page, you must drag column from that table into the **Search Columns** and/or **Display Columns** areas.

The standard join operator (=) should already be set.



A standard join (=) returns only rows where the Userlevel column in both tables match. If a guest record in the USERS table has no user level information, then that guest’s record is not returned.

- 8 Click the General tab, and drag the UlevelNum and ULevelDesc columns from the Workspace into the **Display Columns** area.

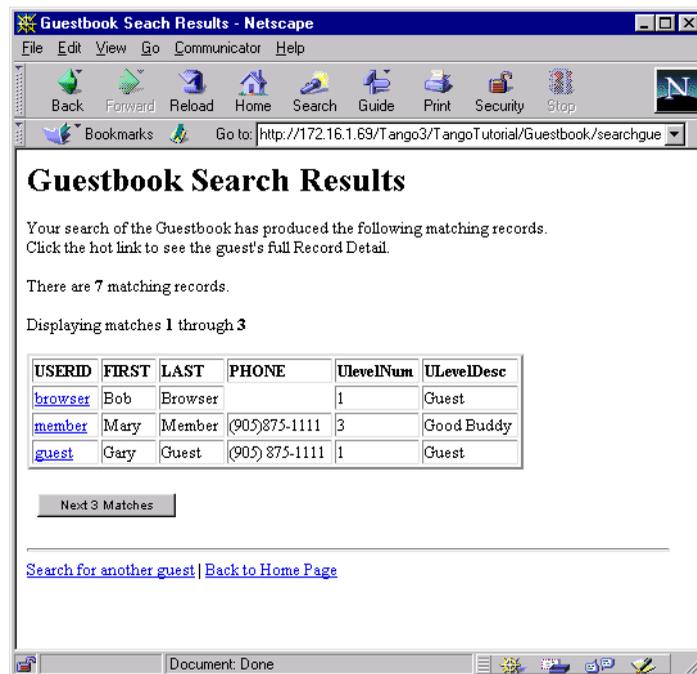
The information from these columns will now appear on your Record List page.

- 9 Click the Record Detail page, then click the Joins tab. Repeat steps 4 to 7.
- 10 Return to General tab on the Record Detail page, and drag the ULevelDesc column from the Workspace to the **Display Columns** area.

The user level description information will now appear on the Record Detail page in your Web browser.

- 11 In the Search Builder window, click **Build Actions**.
- 12 Save `searchguest.taf`, and execute the application file in your Web browser.
- 13 In your Web browser, leave form fields empty, and click **Search Guestbook**.

The Guestbook Search Results page appears, with matching user level information added to the search results table.





Defining different types of joins returns different results. As you saw in the previous steps, a standard join returns only those records from a search in which the related columns match. You can also define *outer joins*, which can be left or right.

A *left outer join* means all the rows in the left-specified table (the Userlevel table) are returned, including those with no match in the right-specified table.

A *right outer join* means that all rows in the right-specified table (the USERS table) are returned, including those with no match in the left-specified table.



14 Return to `searchguest.taf` in Tango Editor.

15 Open the Record List action within the Search Builder by double clicking its icon.

The Record List action window appears.

16 Click the Joins tab.

The Joins dialog box appears with your previously defined join visible.



17 Click the **Oper.** field, and choose `*=(Left Outer)` from the drop-down list to define a left outer join.

Any changes you make are saved when you close the Joins or Record List action window.



You use this window to add, edit or delete a join. Each entry row on the Joins tab represents a join definition. Use **Delete** and **Insert** from the contextual menu to add or remove joins.

18 Close the Record List action window, and save `searchguest.taf`.

19 Return to your Web browser, and execute `searchguest.taf`.

20 Leave form fields empty, and click **Search Guestbook**.

21 Click through the results of your search.

- Q. Do you notice any difference from the results returned by a standard join definition?
- A. When you click through the results to the last page, you see that a row is returned that contains only user level information.



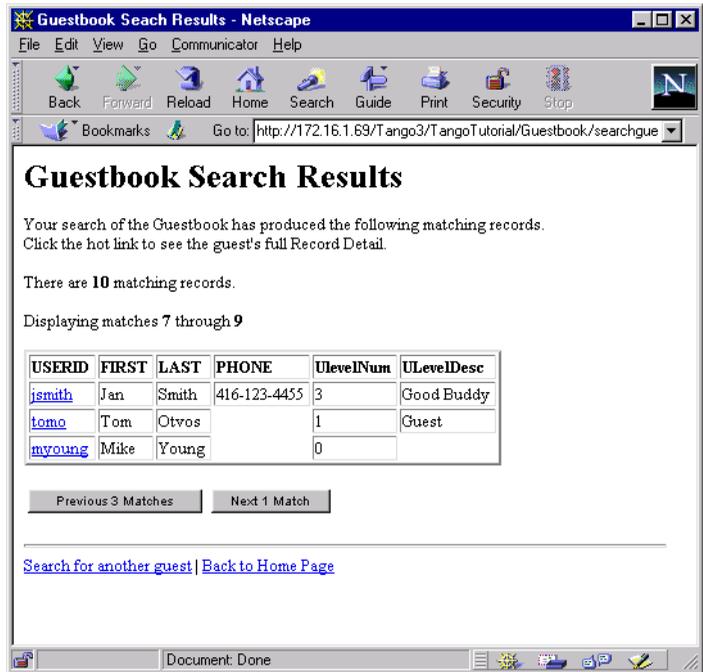
With a left outer join defined, Tango searches both database tables and returns all the UlevelNum and ULevelDesc rows from the Userlevel table (the left table), but only the rows from the USERS table which were related to existing Userlevel rows.

Additional Exercise

Now see what happens when you search the guestbook using a right outer join.

- 1 Repeat steps 13 to 15 from the previous exercise to return to the Joins tab.
- 2 Click the **Oper.** field, and choose `=*(Right Outer)` from the drop-down list to define a right outer join.
- 3 Close the Record List action window, and save `searchguest.taf`.
- 4 Execute `searchguest.taf` in your Web browser, and search the guestbook leaving the form fields blank.

- Q. What do you notice when you click through the search results?
- A. You see that some rows are returned that contain no user level information.



Tango searches both database tables and returns all rows from the USERS table (the right table), whether or not the row contains user level information in the Userlevel table. It returns only the UlevelNum and ULevelDesc columns that match columns in the USERS table.

In this example, a right outer join is most useful, because you are searching primarily for guest information. The right outer join returns all guest rows, whether or not they contain any user level information. A left outer join does not return any guest information that does not have related user level information. A standard join returns only rows that contain both guest ID and user level rows.

If you were mainly interested in searching for user level information, a left outer join would be more appropriate.

Guestbook Project



Creating a Tango Project for the Guestbook Solution

This tutorial introduces you to a larger organization in Tango, the *project*. A Tango project comprises any number of Tango application files and represents a larger organization within your Tango site.

Tutorial F includes one lesson, creating a Guestbook project.

LESSON F - 1

Creating a Guestbook Project

Purpose

To create and use Tango Editor projects.

Context

You now have two Guestbook application files: `signguest.taf` and `searchguest.taf`. Tango Editor allows you to group related files such as these into a *project*.

A Tango Editor project file contains information about a project you create, including a list of the files, snippets, and data sources grouped in this project.

The purpose of a project is to allow you to logically group and work on a set of application files that are related.

Result

The Guestbook project file. There are no changes in the Guestbook application files, or the Web pages they generate.

Exercise



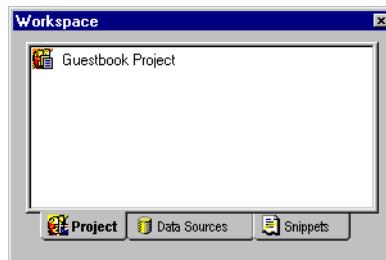
- 1 Open Tango Editor.
- 2 From the **Project** menu, choose **New**.



Note You cannot create an *untitled* project. When choosing **New**, Tango prompts you for a file name to save the project under.

- 3 Navigate to the `Guestbook` folder you are using in this tutorial.
- 4 Name the project `Guestbook`, and click **Save**.

Tango automatically adds the file extension `.tep`, short for Tango Editor Project, to the project file.





A Project tab is added to the Workspace window and your new Guestbook project is displayed; you can only open and work on one Tango Editor project file at a time.

To create a project, it does not matter if you have any application files currently open, and you do not need to open, or have open the files that will be assigned to the project.



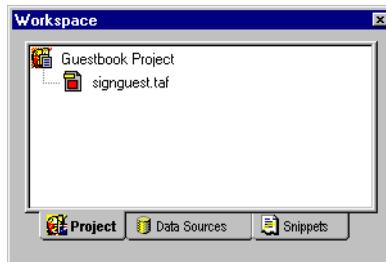
- 5 To add your Guestbook application files to the Guestbook project, from the **Project** menu choose **Add Files**.

The **Add Files into Project** dialog box appears.

- 6 Ensure you are in the Guestbook folder, and choose `signguest.taf` from the list of files.

- 7 Click **Open**.

The **Add Files into Project** dialog box closes and `signguest.taf` now appears as an item within the Guestbook project.



Note `signguest.taf` is not opened.

- 8 Repeat steps 5 through 7 for `searchguest.taf`, thereby adding it to your Guestbook project.

- 9 From the **Project** menu, choose **Close** to save your changes to the Guestbook project, namely the addition of `signguest.taf` and `searchguest.taf`.



The project file is written to disk. In the future, you can simply open the Guestbook project file, and open any Guestbook application files you want to work on from there by selecting the one you want in the Project workspace window, clicking the right mouse button, and choosing **Open** from the menu. You can remove files from a project in the same manner. Again, the purpose of the Guestbook project is to organize logically a set of Guestbook application files.

Guestbook Administration



Allowing Updates and Deletions to the Guestbook

This tutorial adds more components to your dynamic Web site. You add the ability to update or delete a guest from your guestbook.

The following lessons make up Tutorial G:

- allowing an update to your guestbook
- allowing a deletion from your guestbook
- adding a Guestbook Administration link to the Tutorial Administration Page.

LESSON G - 1

Allowing Update of Guest

Purpose

To generate a series of actions in a new application file using the Search Builder that will update existing records in the database.

Context

The current Record Detail page of your guestbook does not allow any updating or modifying of the guest's record in the database. As owner of your guestbook, you may want to be able to update these records if and when necessary, just as you would want to do with the records of any database. You also do not want to be limited to updating the records only in the database file itself.

You can create an application file that allows the user to modify or update database records. Allowing the update of records is a column option on the Record Detail tab in the Search Builder.

You may want to limit some users' ability to modify guests' records, since they have access to all the guests' records. In this lesson, you create a guestbook administration file for your use only, which allows the update of guestbook records. Only you will know about this file, so others may not execute it. It will eventually be a link on your Tango Tutorial Administration Page.

Results

An application file called `guestadmin.taf` that allows guest records in the Tango Tutorial database to be edited. The Web pages will look exactly as they do in `searchguest.taf`, except the Record Detail will have editable fields for the guest's record.

Exercise

- 1 Open Tango Editor. If it is not open already, open your guestbook project by choosing **Open** from the **Project** menu.
- 2 In the Project workspace, double click `searchguest.taf` to open it.
- 3 Double click the Search Builder icon to open it.



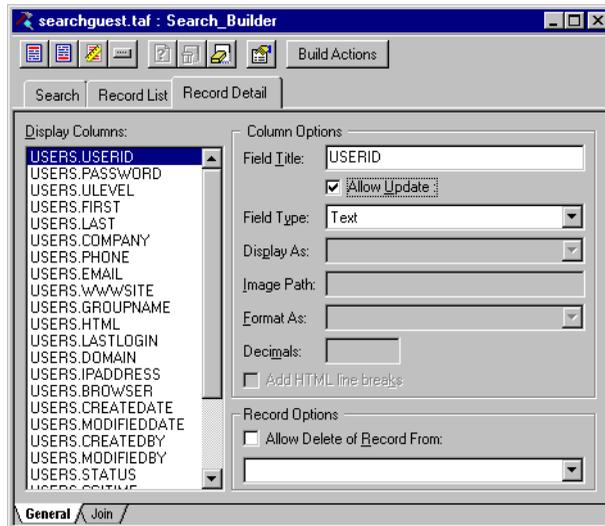
You continue to work with the Search Builder in this guestbook model. Your `guestadmin.taf` can look exactly the same as your `searchguest.taf`. Hence, you can enable the **Allow Update** option in the `searchguest.taf` and save the application file as `guestadmin.taf`.

- 4 Click the Record Detail tab in the Search Builder.
- 5 Click the first column in the **Display Columns** list.



In the **Column Options** section, there is an option to allow update of the selected column. If **Allow Update** is not enabled, Tango displays the data from the column as text that cannot be modified in the browser, as is the current case for `searchguest.taf`. If **Allow Update** is enabled, Tango inserts the data from the column in an editable form field, and adds **Save** and **Reset Values** buttons at the bottom of the Web page.

- 6 Select **Allow Update** for the second column, `USERS.USERID`.





When you enable **Allow Update**, **Field type** becomes active, and all of the formatting options below it are disabled. Because the column data is displayed in a form field, it cannot be displayed in any other format, such as a hot link. **Field type** becomes active because you have the option to create a drop-down list, or some other such entry field type for the selected database column. In this lesson, keep the field type default of **Text**. This means the data appears as editable text in the form field, allowing the user to type anything in the form field. When you click **Save**, Tango updates all the column data that is within a form field, regardless of whether or not it has changed.

For more information on primary keys, see "Using User Keys" in the *User's Guide*.



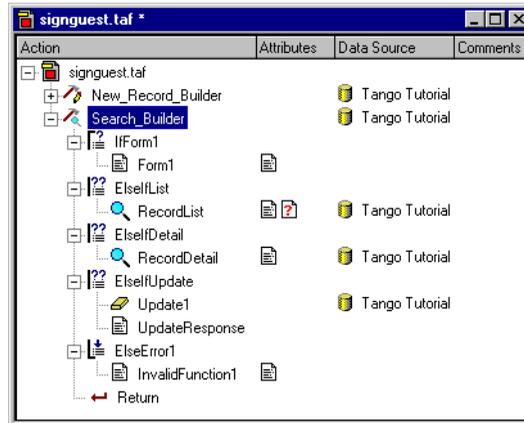
Caution For the purpose of this tutorial, do not enable **Allow Update** for the `USERS.UserKey` column. This column is the primary key for the data source.

In general, be careful in allowing update of a primary key column in your database. You need to ensure any column updates of the primary key are unique values, thereby maintaining the integrity of your database. Individually select the rest of the columns in the **Display Columns** list, and enable **Allow Update** for each.

- 7 Individually select the rest of the columns in the **Display Columns** list, and enable **Allow Update** for each.
- 8 Click **Build Actions** to generate a new series of actions in the application file that reflect your changes.

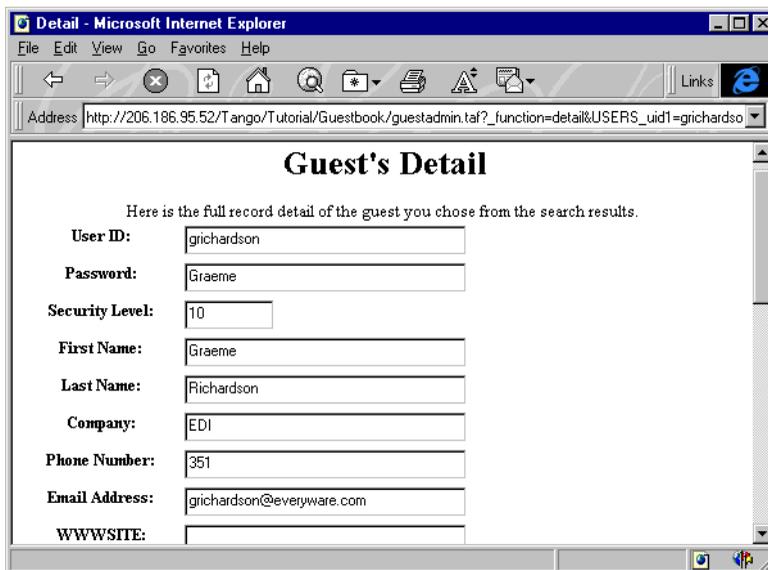


The new set of actions under the Search Builder now contain an Update action, along with an Else If action that specifies the criteria by which Tango executes the Update action.



- 9 From the **File** menu, choose **Save As**. Save the new application file as `guestadmin.taf` in the Guestbook folder.
- 10 Add `guestadmin.taf` to the Guestbook project by choosing **Add Files** from the **Project** menu. In the Guestbook folder, choose `guestadmin.taf` and click **Open**.
- 11 Save the guestbook project with the addition of `guestadmin.taf` by choosing **Close** from the **Project** menu.
- 12 Return to your browser and execute `guestadmin.taf`.
- 13 Click your way to the Record Detail page for a guest you entered during the Sign my Guestbook lesson.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.



Q. What do you notice about the column data displayed?

- 14** Make a change to some of the data, and click **Save**.
- 15** Return to the Search page and search for the guest you just updated.

Q. On the Record Detail page, do your changes appear? You may update them again here.

Additional Exercise

Upon updating a guest's record, a new Web page is displayed stating, "The record was updated successfully".

- 1** Return to the Record Detail window of `guestadmin.taf` in Tango Editor.
- 2** Find and open the Update Response HTML window, and modify the message to suit your guestbook solution. For example, you may want to say, "The guest's record was updated successfully".



Modifying the Update Response HTML is similar to modifying the No Results HTML in the Search window. There are no Tango header or footer sections in either; the windows for each represent the entire Web page that is displayed to the user. Hence, all HTML you want displayed must be inserted in the one window.

LESSON G - 2

Updating Hot Links Using the Replace Operation

Purpose

To introduce the replace operation by updating hot links that reference `searchguest.taf`. Instead, these links will reference `guestadmin.taf`.

Context

In the previous lesson, you replaced `searchguest.taf` with `guestadmin.taf`. If you return to the Tango Tutorial Home Page from `guestadmin.taf` and try to search the guestbook again, the updates you made to the Record Details page no longer exist. This is because the links in `default.htm` (the Home Page), point to `searchguest.taf`. To continue using `guestadmin.taf`, you must change all references that point to `searchguest.taf` to `guestadmin.taf`. You can efficiently update these hot links using the replace operation in Tango.

Result

When you navigate through the links of your Guestbook, all of the links will reference `guestadmin.taf`, instead of redirecting searches through the older application file, `searchguest.taf`.

Exercise

- 1 In your text editor, choose **Open** from the **File** menu.
- 2 Select `searchguest.taf` from the Guestbook folder and press **DELETE** on your keyboard.

A dialog box appears, asking if you want to send `searchguest.taf` to the recycle bin.

- 3 Click **Yes**.

You have deleted the `searchguest.taf` application file. All the information it contained still exists in `guestadmin.taf`.



For more information on find and replace operations, see “Finding and Replacing Text” in the *User's Guide*.

The find, and find and replace operations in Tango allow you to search many different types of windows in Tango for the text *string* you want to modify or replace. Tango will only search for text that you have explicitly entered, but this can include values in criteria lists, action parameters, custom column definitions, and HTML. You can perform find, and find and replace operations in open application files, action editing windows, HTML editing windows, and projects.

4 Open `signguest.taf` from the Guestbook project.

5 From the **Edit** menu, choose **Replace**.

The Replace dialog box appears.

6 In the **Find** field, type “searchguest.taf”.

7 In the **Replace With** field, type “guestadmin.taf”.

8 Under **Find In**, enable “File signguest”.



The **Find In** options allow you to specify the range of your search. You can search an entire application file or you can limit your search to the current window.

If you enable the **Start at top** option, the operation begins from the top of the search range. Otherwise, it begins the search from the current cursor position.

9 Click **Find Next**.

The New Record Builder Response window of `signguest.taf` opens, with “searchguest.taf” highlighted.

10 Click **Replace**.

Tango searches the entire `signguest.taf` application file for instances of “searchguest.taf”. Each time it finds this string, that window opens, showing you the highlighted string. The operation waits for you tell it to replace the string or find its next instance.

11 Continue replacing, until Tango gives you the message “Finished searching for ‘searchguest.taf’”.

12 Click **OK**, and close the Replace dialog box.

- 13 Save the changes you made to `signguest.taf`.
- 14 Open `guestadmin.taf` from your Guestbook project, and repeat steps 5 to 13.



Note In the Replace dialog box, you can choose **Replace All** to change all instances of a given string in one step.

- 15 In your text editor, open `default.htm` in the Guestbook folder.

Because `default.htm` is not a Tango file, you must update its `seachguest.taf` hot link manually.

- 16 Change the hot link surrounding “Search my Guestbook” to read:

```
<A HREF="/Tango3/TangoTutorial/Guestbook/guestadmin.taf">Search my Guestbook.</A>
```

- 17 Return to your browser and execute `guestadmin.taf`. Check the links to and from the Tango Tutorial Home Page to ensure it links properly to the correct application file.

LESSON G - 3

Allowing Deletion of Guest

Purpose To delete a record from the database by using the Search Builder to generate the series of actions that can accomplish this.

Context For the purposes of database administration, you need the ability to delete records from the database. The Search Builder gives you the option to allow the delete of a record when on the Record Detail page.

Result Modification to `guestadmin.taf` so that a button to delete a record appears on the guest Record Detail page, in addition to the option to update a record.

Exercise

- 1 In Tango Editor, open the Search Builder in `guestadmin.taf`.
- 2 Click the Record Detail tab.

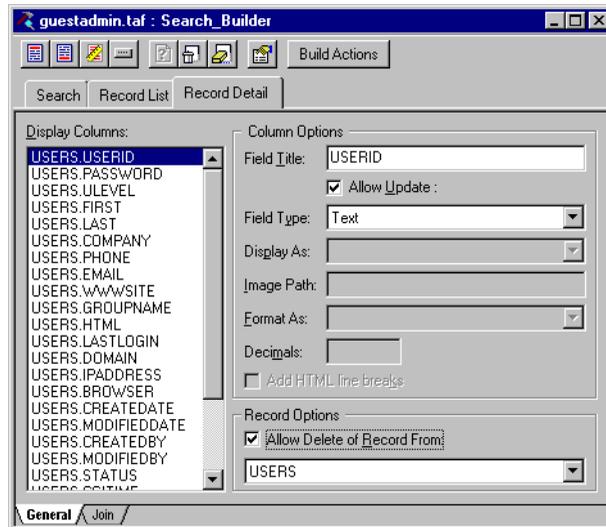


Under the **Column Options** section, there is a **Record Options** section. Whereas the column options are specific to each column listed in the **Display Columns** list, the options you set in the **Record Options** section apply to the entire record that is returned and displayed on the Record Detail page. Hence, it is not necessary to select a column to set the record options.

In **Record Options** you find the option to allow deletion of a record. Selecting **Allow Delete of Record From** adds a **Delete** button to the Record Detail Web page.

When you choose this option, the drop-down list below it is enabled, listing the different tables in the event the columns in the **Display Columns** list come from different tables. You are working with only one table to build your guestbook, so only that table appears in the drop-down list.

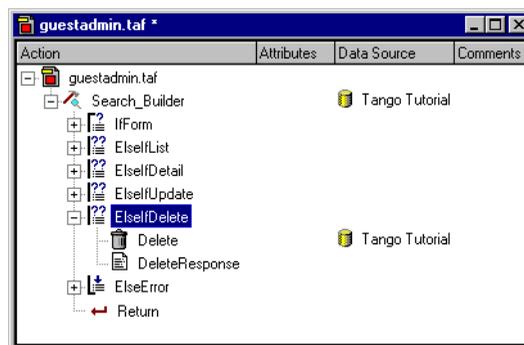
3 In the **Record Options** section, enable **Allow Delete of Record From**.



4 Click **Build Actions** to generate a new series of actions in the guestadmin.taf application file.



The new set of actions under the Search Builder now contain a Delete action, along with an Else If action that specifies the criteria by which Tango executes the Delete action.



5 Save guestadmin.taf.

For information on how to execute application files in your browser, see “Executing Application Files” on page 4.

- 6 Return to your browser and execute `guestadmin.taf`.
- 7 Navigate to the Record Detail page for a particular guest, specifically a guest you would like to delete.
- 8 Click the new **Delete** button.

The record is deleted from the Tango Tutorial database, and the message, “The record was successfully deleted” appears.

- 9 Click **Back** in your browser.

Q. Why is the deleted record still displayed?

A. The record is still displayed because you are looking at a browser-cached version of the Record Detail page. If you reload or refresh the page, Tango returns a message stating “No matching records were found”. This is because Tango conducts another search of the database upon reload, using the primary key identified in the URL line as the search criteria. It cannot find the record this time, so it returns the message. This is one way to test if the record was deleted.

- 10 Navigate back to the Search my Guestbook search form page.
- 11 Leave all the search fields blank, and click **Search Guestbook**.

The Record List page displays all the records in the database. The guest you just deleted is not listed, indicating the deletion was successful.



Caution Deleting a record from a database is permanent. There is no way to retrieve a deleted record.

Additional Exercises

Upon deleting a guest's record, a new Web page is displayed with the message, "The record was deleted successfully".

- 1 Return to the Record Detail and modify this page in the Delete Record Response HTML window.
- 2 Change headings and messages to reflect your guestbook solution, and add hot links to the Tango Tutorial Home Page and to the Search my Guestbook form page. By offering these hot links, you decrease the chances of the user hitting the **Back** button in the browser and seeing the deleted record still on the screen.
- 3 Check your changes in the browser.

The Search Builder allows you to change the titles of the **Update** and **Delete** action buttons in the Record Detail window.

- 1 Return to the Record Detail window of the `guestadmin.taf` Search builder, and open the Button Titles dialog box by clicking the Button Titles icon, or by choosing **Button Titles** from the **Attributes** menu.
- 2 Change the titles to the following titles, which are more appropriate for the guestbook solution.
 - **Save** becomes **Update Guest**.
 - **Reset Values** becomes **Clear**.
 - **Delete** becomes **Delete Guest**.
- 3 Check your changes in the browser.



Modifying the Delete Response HTML is done in the same way as modifying the No Results HTML in the Search window and the Update Response HTML in the Record Detail window. Changing button titles is also done in the same way as in the Search window. (The Record List page has no available buttons, so **Button Titles** is disabled.)

- 4 Modify the heading of the Search page from "Search my Guestbook" to "Guestbook Administration". This is a more appropriate heading for the first page of the `guestadmin.taf` file execution.

LESSON G - 4

Adding a Guestbook Administration Link to the Tutorial Administration Page

Purpose To integrate application files with standard HTML Web pages, and to learn the purpose of having an administration page.

Context Now that you have a working `guestadmin.taf` application file, which allows you to maintain guests in the database, add a hot link to it from your Tango Tutorial Administration Page.

The process for adding the hot link is the same as for making live links on your Tango Tutorial Home Page, except you are now making a link on the Tango Tutorial Administration Page, `admin.htm`.

Result Your Tango Tutorial Administration Page with a working link to Guestbook Admin that when clicked executes `guestadmin.taf`.

Exercise

1 Open `admin.htm` in your text editor.

This file is found in the `Guestbook` folder you have been working in.

2 Find the text "Guestbook Admin".

3 Wrap a hot link or anchor tag around it as follows:

```
<A HREF="guestadmin.taf?function=form">Guestbook  
Admin</A>
```

4 Save `admin.htm` in the `Guestbook` folder.

5 Return to the browser and call up the Tango Tutorial Administration Page.

6 Test the **Guestbook Admin** link.

7 Click through to the Record Detail page and update or delete a guest.

Q. Does everything work as it should?

Login Solution

H

Creating a Login Solution With Search Builder and Customizing It

One of the most common Tango solutions is the login model. The login model involves a page where users enter their user ID and password. If these values match values in the database, the user is allowed into the *system*, and a welcome screen is displayed. The menu options are other application files, which are executed, or available, if the user has the appropriate user level. The Tango login model is easy to build and secure.

The following lessons make up Tutorial H:

H1: Login Framework Using Search Builder

- using the Search Builder to create a framework for the login application file

H2: Modify Login Application

- modifying the password field
- setting the business logic of the login solution
- creating a personalized welcome page

H3: Logged-in User Information

- storing and tracking user information
- preventing unauthorized access to application files in the options menu.

LESSON H 1 - 1

Using Search Builder to Create a Framework for the Login Application File

Purpose

To use the Search Builder to build a large part of the functioning login solution.

Context

In general terms, the login model is simply a search of a database. The database contains all the valid users in the system, including their user ID and password.

When a user logs in by entering a user ID and password, Tango searches to see if these values match any of those in the database. If a match is found, indicating the user is valid, the user is permitted to see the next page, the welcome screen, which presents a list of menu options. The menu options are links that allow the user to perform other functions or activities, such as registering guests/members, looking up members, and updating or deleting members. If the user ID and password do not have a match in the database of valid users, an invalid login response is displayed.

Outcome

The login screen, presented first to the user, containing two form fields into which the user enters a user ID and password. If the login is successful, a personalized welcome screen appears with a set of menu options. If the login is unsuccessful, an invalid login message appears.

Exercise



- 1 Open a new application file in Tango Editor.
- 2 Drag the Search Builder icon into the application file.
The Search Builder window opens.

- 3 Use the USERS table in the Tango Tutorial data source you created in Tutorial C.

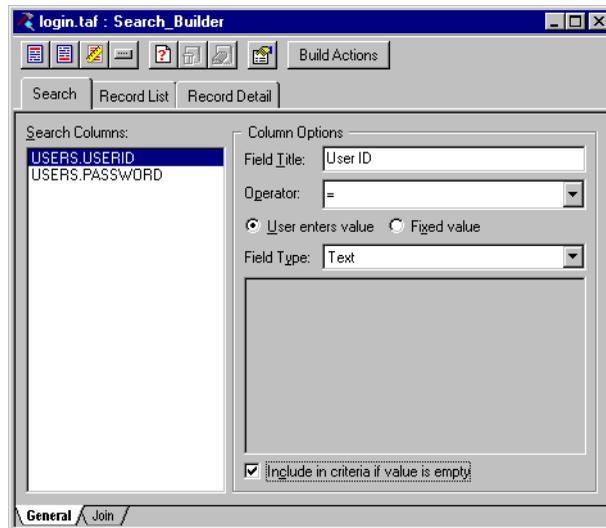


Note The USERS table holds all the guests that sign your guestbook, which you built in previous lessons. These guests will now become valid users in the login system, and many may not have passwords because the guestbook does not ask guests to enter a password. This tutorial provides you with users with valid user IDs and passwords where necessary. They already exist in the database.

- 4 Drag the USERID and PASSWORD columns to the **Search Columns** list.
- 5 Select the USERS.USERID column in the **Search Columns** list.
- 6 In **Field Title**, change the default field title to “User ID”.
- 7 In **Oper.**, select =, if not already selected.

The checkbox for **Include in criteria if value is empty** now becomes available.

- 8 Enable **Include in criteria if value is empty**.





The **Include in criteria if value is empty** option determines whether or not Tango returns all the records in a database if the search criteria is left empty by the user. You may recall in the Search my Guestbook search page that leaving the form fields empty returned all the guests from the database. That is because this option was disabled. There were no criteria specified—it had an empty value—so Tango ignored it and returned all records. It did not include the criteria of the Search action. Enabling **Include in criteria if value is empty** forces Tango to use the criteria and the operator no matter what the search criteria value is, even if it is a blank or empty. If the criteria is blank, Tango goes off into the database and searches for a blank record that matches. If it does not find one, it returns a no matching records message; if one is found, Tango returns the blank record.

In the login model, then, you do not want a user to leave the user ID and password form fields blank, and suddenly see a listing of all the valid users and their passwords in the database. If the user leaves it blank, they should receive a no matching records message.

It is your responsibility as database administrator to ensure there are no blank records in the database. Because of the nature of the **Include in criteria if value is empty** option, it is only available or applicable when you set the search operator to =. For example, if you specify a **contains** search and the user leaves the search criteria blank, Tango cannot search for a blank record.

-
- 9 Repeat steps 5 to 8 for the USERS.PASSWORD column, changing the columns field title to “Password”.
 - 10 Click the Header HTML icon, or choose **Header HTML** from the **Attributes** menu.



The Header HTML editing window opens.

- 11 Replace the existing HTML with the following:

```
<TITLE>Login</TITLE>
<H2>Please Log In</H2>
Enter your User ID and Password to gain access to
the system.<BR> <BR>
```

- 12 Close the window to save this information.
- 13 Click the Button Titles icon, or choose **Button Titles** from the **Attributes** menu.



The Button Titles dialog box appears.

- 14 In the **Find** field, type “Login”.
- 15 In the **Reset Values** field, type “Reset”.
- 16 Click the No Results HTML icon, or from the **Attributes** menus choose **Responses**, then **No Results HTML**.
- 17 Replace the existing HTML with the following:



```
<TITLE>Invalid Login</TITLE>

<H3>The User ID and/or Password entered are
incorrect.</H3>

Please <A HREF="<@CGI><@APPFIL>?_function=form">
try again.</A>
```



This HTML is displayed if no records match the user ID and password entered and provides a link back to the login form so the user can try again. In the hot link code, `<@CGI>` is a Tango meta tag that resolves to the file path and the file name of the Tango CGI, for example `/scripts/t3CGI.exe`, which you must specify if the Tango plug-in is not being used. The `<@APPFIL>` meta tag resolves to the file name of the current application file, including the path to that file. You will save this application file as `login.taf` in the `\Tango3\TangoTutorial\Login\` folder. Hence, `<@APPFIL>` resolves to `/Tango3/TangoTutorial/Login/login.taf`.

The purpose of using both these meta tags is to avoid hard coding these paths. The meta tags allow you to easily change the file path or the file names of either the Tango CGI or the current application file without the additional task of having you change any links that followed the old paths and names. A good exercise is to create a new application file, drag in a Results action, and type out the above meta tags. Save the file in some folder in the Tutorial folder and execute the application file in the browser. The meta tags should return the values for them at that moment in time in the browser screen. Experiment with placing the file in different folders and see what is returned.

- 18 Close the No Results HTML window to save the information.
- 19 Click the Record List tab.
- 20 From the USERS table, drag the USERID, FIRST, and ULEVEL columns into the **Display Columns** list.



You need the values for these columns to come out of the database upon a user's successful login. For instance, once a user logs in, you want Tango to retrieve the user's User ID (which is obviously the same as what the user entered, but it is confirmed as being in the database now), their first name (FIRST), and their user access level (ULEVEL). You use these elements to create a personalized welcome page for the user. You do not need to do any page formatting or field title formatting as you will modify the Record List page considerably to be the welcome screen and none of the settings will matter.

- 21 Click **Build Actions** to generate the Search Builder actions.
- 22 Close the Search Builder.
- 23 Save the application file as `login.taf` in the `Login` folder.
- 24 Return to your browser and execute the file you just created.

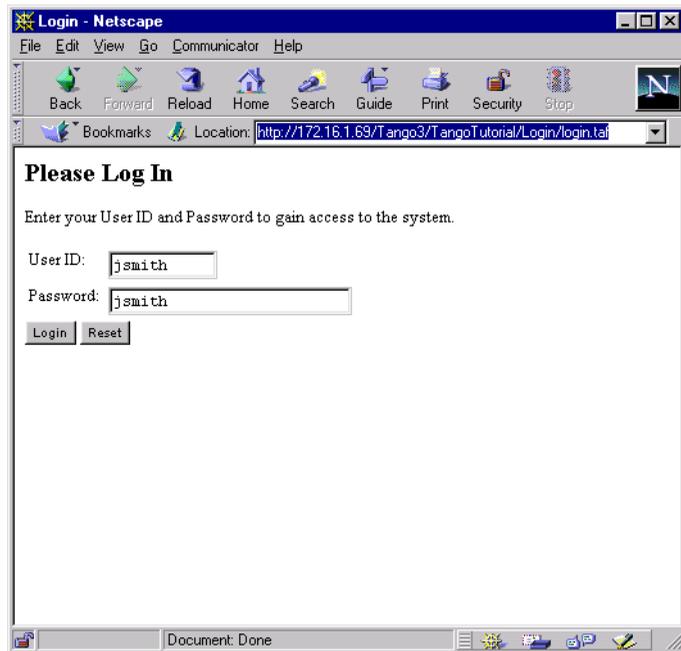
For information on how to execute application files in your browser, see "Executing Application Files" on page 4.



You must send the search argument `_function` with the value of `form` to `login.taf` because it determines which set of actions—and therefore which HTML pages—will be executed in `login.taf`.

The opening page of the login appears, with form fields for entering **User ID** and **Password**. Test the login by logging in as the following valid user in the database.

User ID: jsmith
Password: jsmith



Notice the password is not encrypted with asterisks or dots as you type it into the **Password** form field. This is because the `<INPUT>` tag that produced it is of a text type, not a password type. You will change this in the next lesson.

The Record List page appears with the Search Builder default formatting. You will modify this page to suit the login model shortly. It will contain a personalized message to the user, and it will list a set of menu options.

- 25 Click **Back** in your browser to return to the login page.
- 26 Try logging in as a user that does not exist in the database.

The invalid login message is displayed. Test your hot link back to the login form page to ensure it works.

- Q.** If **Include in criteria if value is empty** is enabled, and a user leaves the search criteria blank, Tango returns which of the following: all the records in the database; a database error; all blank records that exist in the database, if any exist; or the first record in the database?
- A.** All blank records that exist in the database, if any exist.

LESSON H 2 - 1

Modifying the Password Field

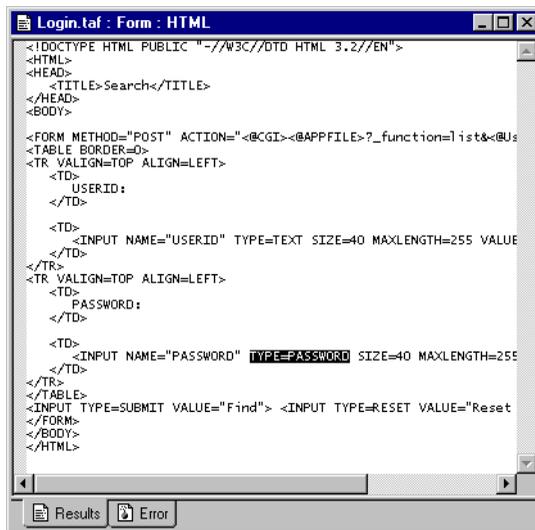
Purpose To change the **Password** form field on the login form page so that it displays asterisks or bullets when the user types it in.

Context A good login system displays the password typed in by the user as a series of bullets or asterisks, depending on your browser, to prevent the viewing of the password as it is entered.

Result On the login page, the form field for **Password** displaying a series of bullets or asterisks instead of letters.

- Exercise**
- 1 Return to `login.taf` in Tango Editor.
 - 2 Open the Results action of the IfForm action in the Search Builder by double clicking its icon, or by selecting it and choosing **Results HTML** from the **Attributes** menu.
 - 3 Find the HTML that reads:


```
<INPUT NAME="PASSWORD" TYPE=TEXT SIZE=40
MAXLENGTH=255 VALUE=" ">
```
 - 4 Replace `TYPE=TEXT` with `TYPE=PASSWORD`.
 - 5 Close the Results HTML editing window.



```

Login.taf : Form : HTML
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
  <TITLE>Search</TITLE>
</HEAD>
<BODY>
  <FORM METHOD="POST" ACTION="<@CGI><@APPFILE>?_function=list<@US
  <TABLE BORDER=0>
  <TR VALIGN=TOP ALIGN=LEFT>
    <TD>
      USERID:
    </TD>
    <TD>
      <INPUT NAME="USERID" TYPE=TEXT SIZE=40 MAXLENGTH=255 VALUE
    </TD>
  </TR>
  <TR VALIGN=TOP ALIGN=LEFT>
    <TD>
      PASSWORD:
    </TD>
    <TD>
      <INPUT NAME="PASSWORD" TYPE=PASSWORD SIZE=40 MAXLENGTH=255
    </TD>
  </TR>
  </TABLE>
  <INPUT TYPE=SUBMIT VALUE="Find"> <INPUT TYPE=RESET VALUE="Reset
  </FORM>
</BODY>
</HTML>

```

- 6** Save `login.taf`.
- 7** Return to the browser and execute `login.taf` with `_function=form` to display the login form page. If it is on the screen already, refresh or reload the page.
- 8** Type a password in the **Password** field.
The text you type now appears as bullets or asterisks.

LESSON H2 - 2

Setting the Business Logic of the Login Solution

Purpose

To learn where and how to modify the actions generated by the Search Builder to build a proper login solution.

Context

At present, the soon-to-be welcome page and the invalid login message page are Results HTML and No Results HTML attributes of the Search action titled RecordList, which searches the database for the user ID and password entered by the user.

In this lesson you separate these components from the Search action and display each according to whether there are results from the search or not. By separating the welcome page and invalid message page you leave the opportunity to add other actions that may be required in either a valid or invalid login.

Result

The same Web page results as in the previous lesson, but with the welcome page returning the words “Welcome Page”, which will be replaced in the following lesson.

Exercise

- 1 Return to `login.taf` in Tango Editor.



First, you drag in all the actions you need for the logic that determines whether Tango sends the welcome page or the invalid login page to the browser. You drag in the actions all at once but set them after, to help you visualize the flow and business logic of the application file when it executes.



- 2 Hold down the CTRL key and drag an If action into the ElseIfList action. Ensure it appears in sequence after the RecordList action.

- 3 Rename the new If action, `If_Invalid_Login`.



- 4 Hold down the CTRL key and drag a Results action into the `If_Invalid_Login` action so that it becomes a sub-action of `If_Invalid_Login`.

- 5 Rename the Results action, `Invalid_Login_Msg`.



6 Hold down the CTRL key and drag an Else action into the ElseIfList action.

7 Rename the Else action, Else_Valid_Login.



8 Hold down the CTRL key and drag a Results action into the Else_Valid_Login action so that it becomes a sub-action of Else_Valid_Login.

9 Rename the Results action, Welcome_Page.

10 Select the RecordList Search action, and choose **Results HTML** from the **Attributes** menu to open the Results HTML editing window.

11 Delete the HTML shown there and close the window. (You do not need this HTML because its presentation does not suit the login model.)

12 Select the RecordList action, if not already selected, and choose **No Results HTML** from the **Attributes** menu to open the No Results editing window.

13 Select and cut the HTML shown there. This is the HTML you typed in the No Results HTML window in the Search Builder for the login model. However, it should now go in the Invalid_Login_Msg action.

14 Close the window.

The No Results Attribute icon disappears for the RecordList action.

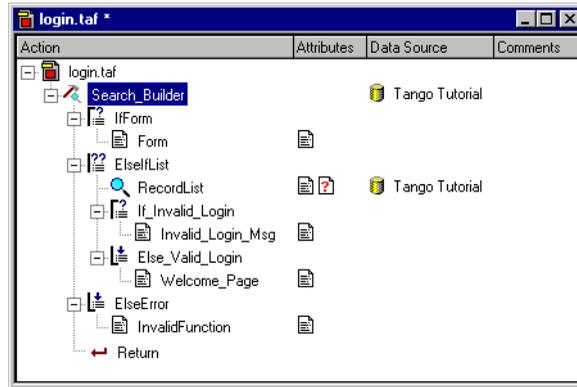
15 Double click the Invalid_Login_Msg action.

A blank HTML editing window opens.

16 Paste in the HTML you cut from step 13 and close the window.



You now have an outline of the login model's logic that takes place after the search for a user ID and password. You now return to set the If and Elseif actions.



The criteria that determines if Tango displays the invalid login message or the welcome message is whether there are any results from RecordList, the search action that searches the database for the user ID and password. If there are results from the search, it means the user's User ID and Password were found in the database; it was a successful login. If there are no results from the search, it means the user's User ID and Password were not found in the database; it was not a successful login.

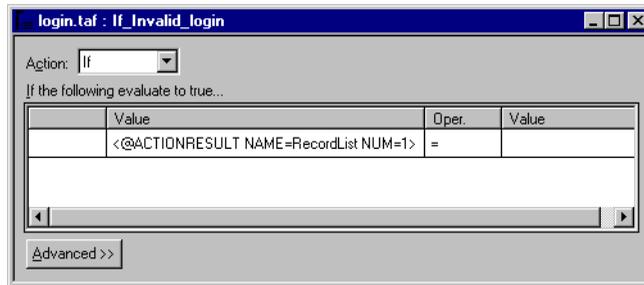
The Tango meta tag that represents or references the search results is `<@ACTIONRESULT NAME=actionname NUM=number [FORMAT=format]>`.

17 Double click the If_Invalid_Login icon.

18 In the first **Value** field, enter `<@ACTIONRESULT NAME=RecordList NUM=1>`. This resolves to the first column value that is retrieved from the database in the event of the search finding matching records.

- 19** From the **Oper.** drop-down list, select **=**.

Leave the second **Value** column empty.



If there is not at least one column value returned from the database, then <@ACTIONRESULT NAME=RecordList NUM=1> is empty. If <@ACTIONRESULT NAME=RecordList NUM=1> is empty, no matching records were found and the Invalid Login page is returned.

- 20** Close the If_Invalid_Login action.



Else_Valid_Login is the corresponding Else action to If_Invalid_Login. You do not need to set an expression to evaluate in this action because it evaluates the opposite expression found in If_Invalid_Login. In other words, if <@ACTIONRESULT NAME=RecordList NUM=1> is *not* empty, the Welcome_Page is displayed.

- 21** Double click Welcome_Page to open the HTML editing window.
- 22** Type “Welcome Page”. This will be a placeholder for the real welcome message you enter in the next lesson.
- 23** Save login.taf.
- 24** Return to your browser and execute login.taf.
- 25** Login with the following user ID and password:

User ID: jsmith
 Password: jsmith

The plain welcome message appears.

- 26** Login with a user you know is not in the database.
The “Invalid Login” message appears.

LESSON H 2 - 3

Creating a Personalized Welcome Page

- Purpose** To modify application file actions to suit the model you are trying to build, in this case the login model.
- To create a welcome page that is personalized for the user that has just logged in successfully.
- Context** At present, the page the user sees after a successful login simply displays the words “Welcome Page”. You need to modify this page so that it resembles a login welcome page and presents some menu options.
- Result** A welcome page that states, “Welcome, (*user’s name*)!” and lists the following three menu options:
- **Guest Registration** allows the user to enter a guest in the database.
 - **Member Look-up** allows the user to look-up, but not modify, a member in the database.
 - **Member Update** allows the user to look-up and modify a member's record.
- Exercise**
- 1 Return to `login.taf` in Tango Editor.
 - 2 Double click `Welcome_Page`.
The HTML editing window opens.
 - 3 Delete the HTML shown there.
 - 4 Type the following HTML:

```
<TITLE>Welcome</TITLE>  
<H2>Welcome, !</H2>
```



To personalize the welcome statement in the <H2> heading, you must reference the value for the column FIRST that Tango retrieves from the database upon successful login. This value is the third column value retrieved from the database by the RecordList action; the second and fourth are USERID and ULEVEL.

The value is referenced using the <@ACTIONRESULT> meta tag: <@ACTIONRESULT NAME=RecordList NUM=3>. “RecordList” is the name of the Search action that is retrieving the column value, and “3” represents numeric place of the column you want to reference, in this case FIRST.



Note Note that the value you use to reference FIRST here is slightly different than that referenced in the last lesson. In the RecordList action, the column USERS.UserKey is automatically inserted before the columns you inserted from the database. Double click the RecordList action to see this change.

5 Place your cursor between the comma and the exclamation mark.

6 From the **Edit** menu, choose **Insert Meta Tag**.

The **Insert Meta Tag** dialog box appears. There is a drop-down menu with meta tag selections.

7 From the **Category** drop-down list, select **Action Result Item**.



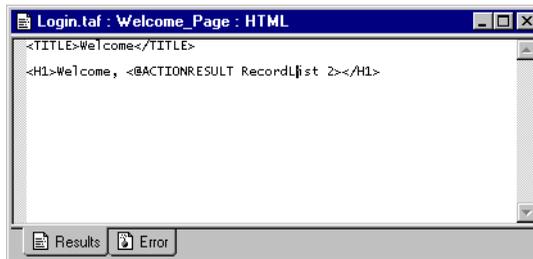
The appropriate actions are listed in the action drop-down list. In the current login application file, only RecordList returns database results, so it is the only action listed.

- In the **Item Number** field, type “2” because FIRST.COLUMN is the second item retrieved from the database.



- Click **Insert**.

The meta tag is entered at your cursor position.



When login.taf is executed,
<@ACTIONRESULT NAME=RecordList NUM=2> resolves to the value
that came out of the database for the user’s first name.

- Close the HTML editing window of Welcome_Page.



The menu items that are displayed on the welcome page will be typed in another Results action dragged in below `Welcome_Page`. You are doing that primarily for educational purposes, to learn how Tango seamlessly joins together HTML found in two or more actions. In reality, you may want to place the HTML for the menu items within `Welcome_Page`.



11 Drag in a Results action below `Welcome_Page`.

The HTML editing window opens.

12 Type the following HTML

```
<P>Here are your options:
<P><A HREF="<@CGI>
<@APPPATH>guestregister.taf?_function=form">Gue
st Registration</A>
<P><A HREF="<@CGI>
<@APPPATH>memberlookup.taf?_function=form">
Member Look-up</A>
<P><A HREF="<@CGI>
<@APPPATH>memberupdate.taf?_function=form">
Member Update</A>
```



Note The hot links connect to other application files that are installed in your `Login` folder; you do not need to create them.

13 Close the HTML editing window.

14 Rename the Results action “Menu”.

15 Save `login.taf`.

16 Return to the browser and execute `login.taf` to get the login form page.

17 Login using the following user ID and password:

User ID: `jsmith`
Password: `jsmith`

The welcome page appears, with a message personalized for Jan.

18 Click each of the menu hot links to ensure they work properly.

LESSON H 3 - 1

Storing and Tracking User Information

Purpose

To keep track of a user after a successful login.

Context

In many Tango applications you create, you may need to keep track of information associated with a particular user's *session* on your Web site and have it available to use in many different places—either across multiple executions of the same application file or in many different application files. In the login solution, for example, you want to retrieve a valid user's user level, so that throughout the rest of the user's session you can control which application files the user is allowed to execute or not execute.

Tango has variables to help you perform this kind of *user state* tracking. *Variables* are name-value pairs that can be associated with a particular user's visit to your site. They are stored by Tango Server and can be accessed from any application file the user executes.

Result

There are to be no changes to the Web pages the user sees. However, the login application file will have an additional Assign action that assigns user information to Tango variables for tracking.



During a database search for the user's user ID and password, certain values exist: two post arguments named UserID and Password with the values the user submitted. If the database produces search results there are three action results: UserID, First, and ULevel.

However, once the welcome page is displayed and execution of `login.taf` ends, all of these values are lost. The only way to retrieve them is to have the user re-submit the post arguments, and have the RecordList action search the database and produce results again. These, however, will only last for one execution of the application file. If you want to use these values elsewhere during the user's login session, particularly the values from the database, you must assign them to Tango variables. How long Tango variables last is based on the time period set by the Tango developer (the default time period in Tango is 30 minutes). In this lesson, you assign the results from the database to Tango variables using the Assign action.

Exercise

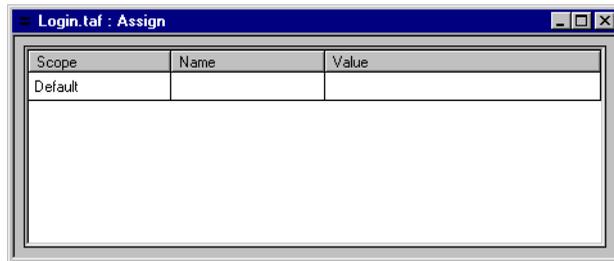
- 1 Return to `login.taf` in Tango Editor.



Note You want to assign the Tango variables only after a successful login, so the Assign action must be inserted in the `Else_Valid_Login` action.

- 2 Drag an Assign action to be the first sub-action of `Else_Valid_Login`.

The Assign action window opens.



You have three column values returned as results from the database, `USERID`, `FIRST`, and `ULEVEL`. Assign all three search results to Tango variables in the order they are retrieved from the database. You are particularly concerned with assigning the `Ulevel` value to a Tango variable because `Ulevel` holds the value of the user's login system access level, and is used to determine what menu items the user is allowed to execute on the welcome page.

- 3 Click the **Scope** field twice, and in the drop-down list select **User**.



Variables with **User** scope let you store and access values associated with a particular user, which is relevant to the login solution you are building.

For more explanation of variable scope, see the *User's Guide*.

- 4 In the **Name** field, type "`UserID`". This names the variable `UserID`; it is logical to name your variable the same name as the column in the database.

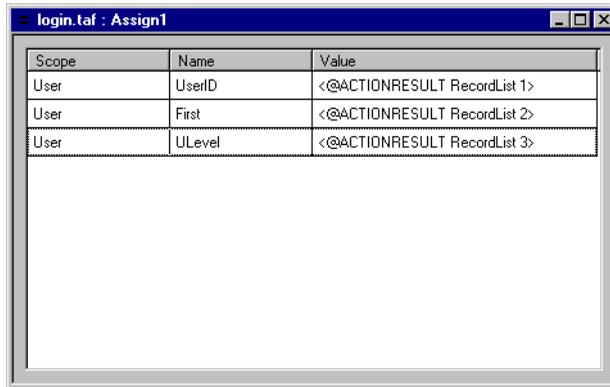
5 In the **Value** field, type `<@ACTIONRESULT NAME=RecordList NUM=1>`.

6 Add another row by right clicking under the **Scope** column, and choosing **Insert Assignment**.

An empty value row appears.

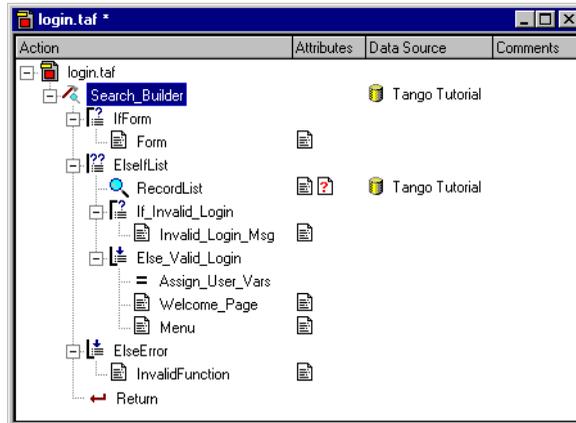
7 Repeat steps 3 to 6 using the following names and values:

“First” `<@ACTIONRESULT NAME=RecordList NUM=2>`
 “ULevel” `<@ACTIONRESULT NAME=RecordList NUM=3>`



8 Close the Assign action editing window.

9 Rename the Assign action, `Assign_User_Vars`.





The user's specific variables are now set, and can be referenced using the `<@VAR>` meta tag as follows:

```
<@VAR NAME=UserID>
<@VAR NAME=First>
<@VAR NAME=ULevel>
```

Because the variable assignments take place before the display of the welcome message in the execution of `login.taf`, you could now replace `<@ACTIONRESULT NAME=RecordList NUM=3>` in `Welcome_Page` with `<@VAR NAME=First>`. Both represent the same value during the execution of `login.taf`.

In the next lesson, you use the variable `ULevel` to determine whether other application files—specifically the welcome menu items, can be executed by the current user. The `ULevel` is checked at the top of these other application files, and the execution is ended if the user's access level `ULevel` is not high enough.

LESSON H 3 - 2

Preventing Unauthorized Access to Application Files in the Options Menu

Purpose

To create a user access-level check at the top of application files to determine whether the files are executed or not for that particular user.

Context

In the previous lesson, you assigned the user's user access level to a Tango variable called ULevel. In this lesson, you insert three actions at the top of three application files that check ULevel and execute the rest of the actions in the file if the user's access level is of a sufficient level. The three actions can be implemented at the top of any application files. The user's access level has been pre-determined and is a value in the database.

Result

A welcome page with three menu options is presented to a user upon successful login: Guest Registration, Member Look-up, and Member Update. The access levels necessary to execute these application files are "1", "5", and "10", respectively. If a user's access level is "1", he or she is able to get to the Guest Registration form, but receive an "Access Denied" message when they execute Member Look-up or Member Update. A user with an access level of "5" can get to the first form page for the first two menu options successfully, but not the last. A user with an access level of "10" can get to the first form page for all three menu options.

Exercise

- 1 Return to `login.taf` in Tango Editor.
- 2 Open the application files that are the three menu options on the welcome page. They are found in the `\Tango3\TangoTutorial\Login\` folder and are called `guestregister.taf`, `memberlookup.taf`, and `memberupdate.taf`.
- 3 Open `guestregister.taf`.
- 4 Hold down the CTRL key and add an If action to the beginning of `guestregister.taf`.



- 5 Hold down the CTRL key and add a Results action to the If action, executed only if the expression in the If action resolves to true. Make sure the Results action is within the If action, that is, indented and below.
- 6 Add a Return action to the If action below the Results action. Make sure the Return action is within the If action, that is, indented and below.
- 7 Rename the three new actions as follows:

Rename ...	To ...
If	Check_User_Level
Results	Access_Denied
Return	Abort

Now that you have the file logic outlined, you return to each action to set it or its attributes.

- 8 Double click the Check_User_Level action.
The If action window appears.
- 9 In the first **Value** field, enter `<@VAR NAME=ULevel>`, which is the value for the current user's access level.
- 10 From the **Oper.** drop down list, select `<` (less than).
- 11 In the second **Value** field, enter `"1"`.



The actions within the If action will be executed only when the user's access level is less than one, which it specifically would be if the user has not logged in at all—it would be 0.

- 12 Close the If action window.
- 13 Double click the Access_Denied action to open the HTML editing window.

14 Type the following:

```
<TITLE>Access Denied</TITLE>
<H1>Access Denied</H1>
```

Either you have not logged in, or your user level does not permit access to this function.

To log in, click

```
<A HREF="@CGI>
<@APPFILEPATH>login.taf?_function=form">
here.
</A>
```



This is the HTML the user sees if the If action above resolves to true.

15 Close the HTML editing window.

16 Save `guestregister.taf`.



It is easy to drag these three actions—`Check_User_Level`, `Access_Denied`, and `Abort`—and use them at the top of any application file. The only item that needs to be adjusted is the access level you want to set for the application file in question.

17 Highlight the first three actions of `guestregister.taf`: `Check_User_Level`, `Access_Denied`, and `Return`.

18 Hold down the CTRL key and drag the three actions to the top of `memberlookup.taf`.

19 In `memberlookup.taf`, open the `Check_User_Level` action.

20 Change the value in the second **Value** field to “5”, so that an access level of five or greater is necessary to continue execution of the application file.

21 Save `memberlookup.taf`.

22 Repeat steps 18 to 22 for `memberupdate.taf`, setting the access level to 10.

23 Return to your browser and execute `login.taf`.

24 Log in as different users with different access levels.

The following users exist in the database. The passwords are the same as the user IDs for each user.

Access Level	Name	User ID
1	Gary Guest	gguest
5	Jan Smith	jsmith
10	Anna Administrator	admin

25 On the welcome page, click through the menu options. The menu options available to you depend directly on the access level of the user you logged in as.

What's Next

Concluding Remarks

Now that you have completed the Tango Enterprise tutorials, you should have a good grasp of the basics required to create your own Web-based applications with Tango.

To use Tango to its fullest potential, refer to the *User's Guide* and the *Meta Tags and Configuration Variables* manual.

For more information on some of the concepts covered in these tutorials, refer to the following chapters in the *User's Guide*:

- Chapter 7, "Working With Variables"
- Chapter 9, "Configuring the Search Builder"
- Chapter 10, "Configuring the New Record Builder"
- Chapter 11, "Using Actions"
- Chapter 13, "Using Basic Database Actions"
- Chapter 14, "Using Control Actions"
- Chapter 18, "Using Advanced Database Actions".

For information on new and updated Tango features, see *What's New in Tango 3*.

