Session VFP322

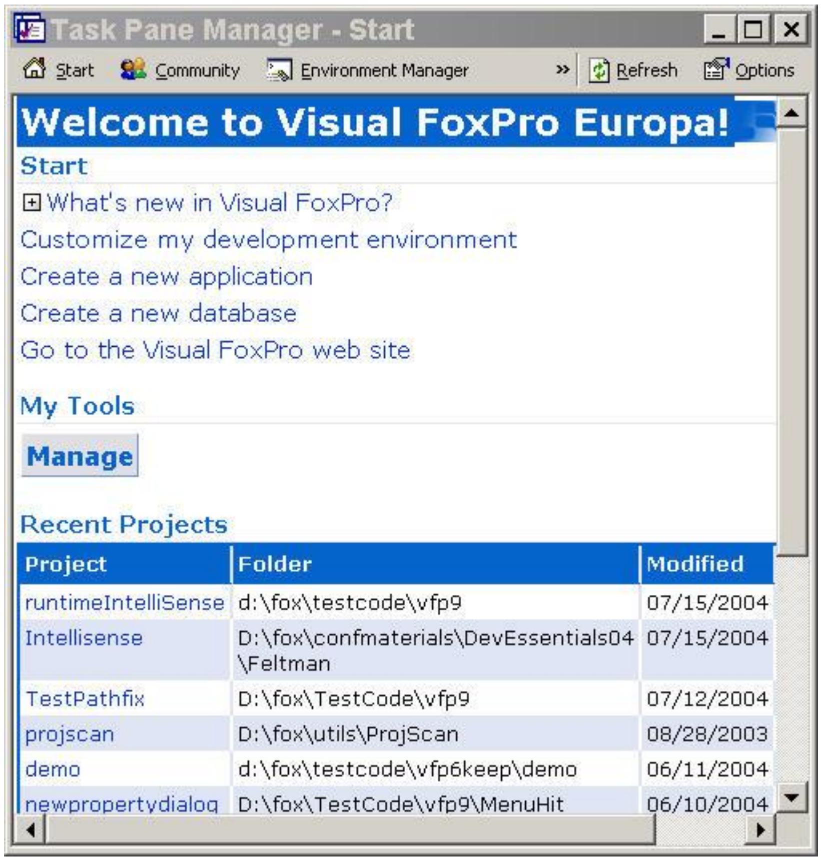
Putting the Task Pane Manager to Work

Tamar E. Granor
Tomorrow’s Solutions, LLC
tamar@tomorrowssolutionsllc.com
www.tomorrowssolutionsllc.com

The Task Pane Manager is a portal for VFP. Added in VFP 8, the tool itself is simply a container—it contains task panes. What's a task pane? Anything you want it to be. The tool comes with a number of panes (seven, in VFP 8; 8, in VFP 9) that span a wide variety of activities, but it's possible to add panes, either ones you write yourself or ones provided by others. In this session, we'll look at what you can do with the provided task panes, and then consider creating and installing custom panes.

Running the Task Pane Manager

You start the Task Pane Manager by choosing it from the Tools menu or from the Standard toolbar. The first time you open it, it opens to the Start pane (Figure 1). The button bar just below the title bar contains a list of available panes. If the list doesn't fit, a chevron button indicates that more panes are available; click it to see the list.



Welcome to Visual FoxPro Europa!

Start

⊞ What's new in Visual FoxPro?

Customize my development environment

Create a new application

Create a new database

Go to the Visual FoxPro web site

My Tools

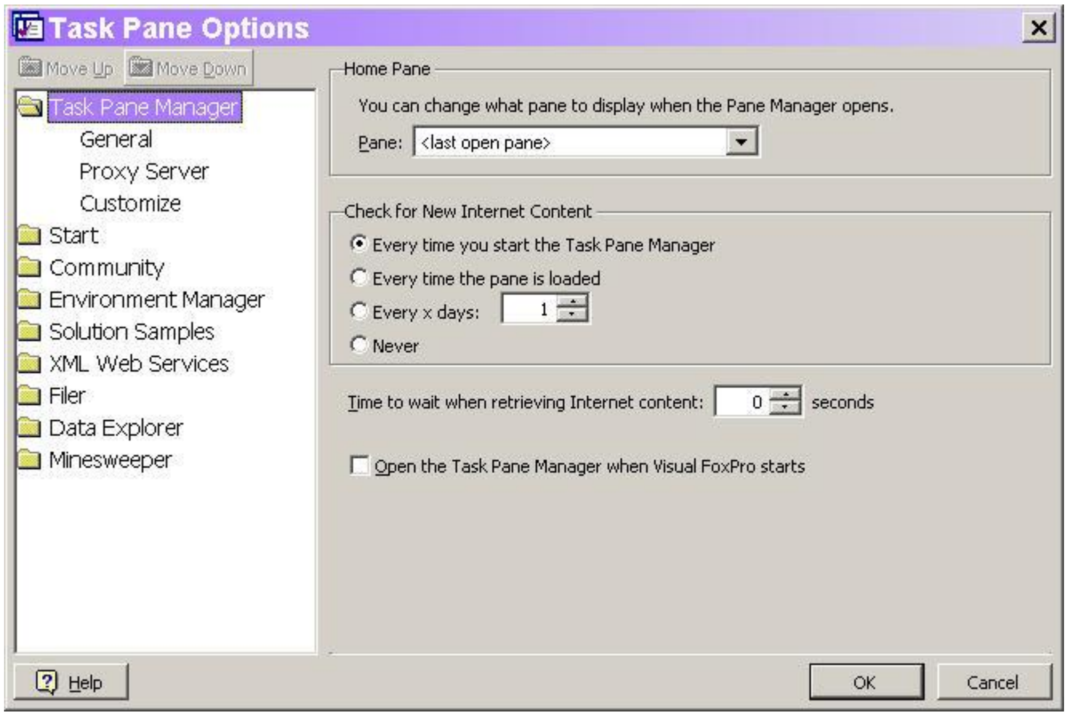
Manage

Recent Projects

Project	Folder	Modified
runtimeIntelliSense	d:\fox\testcode\vfp9	07/15/2004
Intellisense	D:\fox\confmaterials\DevEssentials04 \Feltman	07/15/2004
TestPathfix	D:\fox\TestCode\vfp9	07/12/2004
projscan	D:\fox\utils\ProjScan	08/28/2003
demo	d:\fox\testcode\vfp6keep\demo	06/11/2004
newpropertydialog	D:\fox\TestCode\vfp9\MenuHit	06/10/2004

Figure 1: The Start Pane—This pane gives you quick access to both information and tools, as well as to the projects and databases you've worked with most recently.

You can also set the Task Pane Manager to open when VFP starts; that's the default setting when you install VFP 8 and 9. To change the setting, click the Options button or, in VFP 8, click on Task Pane Manager startup options at the very bottom of the Start pane. Either choice opens the Options window (interestingly, not a dialog) shown in Figure 2. If necessary, click on Task Pane Manager in the list to see options that affect the tool as a whole. To have the Task Pane Manager open when you start VFP, check the appropriate checkbox; to prevent that behavior, clear the checkbox. (Because the tool is written in VFP, CLEAR ALL shuts it down.)



Home Pane

You can change what pane to display when the Pane Manager opens.

Pane: <last open pane>

Check for New Internet Content

(●) Every time you start the Task Pane Manager
( ) Every time the pane is loaded
( ) Every x days: 1
( ) Never

Time to wait when retrieving Internet content: 0 seconds

☐ Open the Task Pane Manager when Visual FoxPro starts

Figure 2: Setting Task Pane Options—This window lets you specify options for the tool as a whole and for individual panes.

The dropdown at the top on this page lets you specify which pane is displayed when the tool opens. By default, it returns to the pane that was displayed when you closed it, but you can choose a specific pane if you prefer.

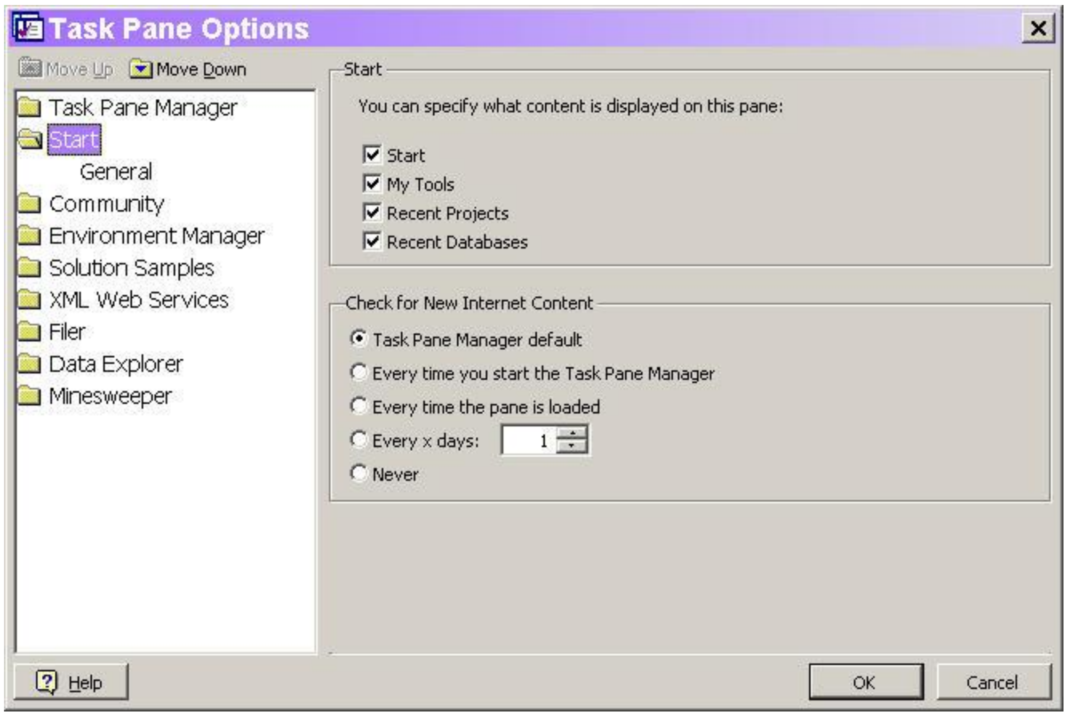
A number of the panes get some of their data from the web. The Check for New Internet Content section determines how often the tool looks for updated data. The setting on this page is a default. The Options page for each pane that has Internet data includes a similar section; there, you can override the default for that particular pane.

Some task panes call XML Web Services to get data. This presents a problem for developers using a proxy server. The Proxy Server page of the Options window, added in VFP 8 SP1, lets you specify whether or not to use a proxy server and provide the necessary settings.

The Start Pane

The Start Pane collects a variety of information and tools into a single location. It's not hard to imagine using this pane as the center of VFP development.

The pane is divided into four sections: Start, My Tools, Recent Projects, and Recent Databases. Each can be controlled individually from the Options window, so you can include as many or as few of them as you wish. (See Figure 3.)



Task Pane Manager
Start
General
Community
Environment Manager
Solution Samples
XML Web Services
Filer
Data Explorer
Minesweeper

Start

You can specify what content is displayed on this pane:

- [✓] Start
- [✓] My Tools
- [✓] Recent Projects
- [✓] Recent Databases

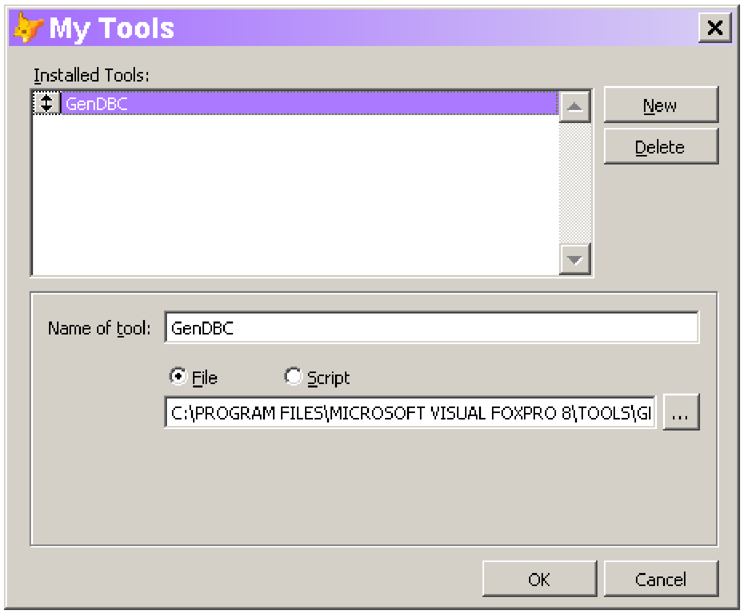
Check for New Internet Content

- [✓ Task Pane Manager default]
- Every time you start the Task Pane Manager
- Every time the pane is loaded
- Every x days: 1
- Never

Figure 3: Controlling pane contents—When a pane has multiple sections, the Options window lets you decide which sections are displayed.

At first glance, the Start section may look like it simply points to documentation. In fact, several of the links actually run tools. What's New in Visual FoxPro and its sub-items are, in fact, links to the VFP help file, and Go to the Visual FoxPro website does what its name suggests. But Create a new application and Create a new database run the Application Wizard and Database Wizard, respectively. Customize my development environment switches to the Environment Manager page, discussed later in this document.

The most intriguing section here is My Tools. When you click Manage, the My Tools dialog (Figure 4) opens. You can add a variety of items here, from VFP applications to forms to custom scripts.



Name of tool: GenDBC

[✓ File] Script

C:\PROGRAM FILES\MICROSOFT VISUAL FOXPRO 8\TOOLS\GI

Figure 4: Adding tools to the Start pane—The My Tools dialog lets you call existing applications, programs, forms and reports, or write scripts in VFP.

In Figure 4, the GenDBC utility is added to the My Tools list. You might also add a script that cleans up after running an application. The script code might look like:

```
SET SYSMENU TO DEFAULT
MODIFY WINDOW Screen
_Screen.Caption = VERSION()
```

Of course, a script could also prompt for input, so it doesn't have to do exactly the same thing every time you run it.

Once you've added at least one item to the list of tools, the My Tools section expands to include a dropdown of tools and a Run button, as shown in Figure 5. Choose a tool and click run to execute the specified tool.

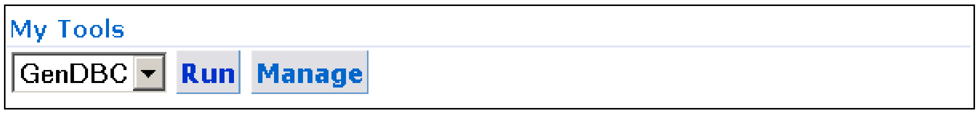


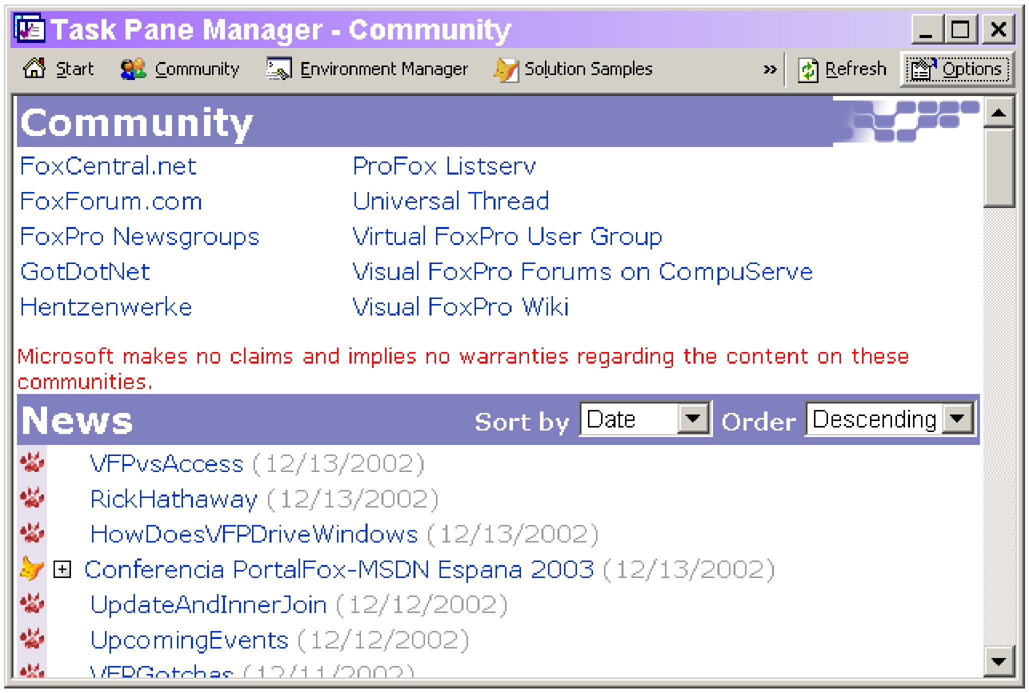
Figure 5: Working with tools--Once some tools are registered, the My Tools section expands to let you use them.

The Recent Projects and Recent Databases section draw their entries from the MRU (most recently used) lists maintained by VFP. The number of items in each section is based on the setting for MRU list size in the Tools>Options dialog. Clicking on a project or database in the list opens it, along with the appropriate tool (the Project Manager or Database Designer). Each section has buttons beneath it that let you open an item not on the list or create a new one.

The Community Pane

Community is a hot button with Microsoft these days. They're putting a lot of effort into building developer communities (something FoxPro has had for a long time). So it's not surprising that the second of the panes supplied by Microsoft focuses on community.

This pane (Figure 6) is divided into two sections, but they don't directly map to the checkboxes on the Options window page for the pane. The top section (Community) is a list of community web sites. Most of them are places where you can ask questions and get answers from your peers. A few have different structures or purposes, but all of them offer opportunities to improve your VFP skills.



FoxCentral.net
FoxForum.com
FoxPro Newsgroups
GotDotNet
Hentzenwerke
ProFox Listserv
Universal Thread
Virtual FoxPro User Group
Visual FoxPro Forums on CompuServe
Visual FoxPro Wiki

Microsoft makes no claims and implies no warranties regarding the content on these communities.

Sort by: Date Order: Descending

VFPvsAccess (12/13/2002)
RickHathaway (12/13/2002)
HowDoesVFPDriveWindows (12/13/2002)
Conferencia PortalFox-MSDN Espana 2003 (12/13/2002)
UpdateAndInnerJoin (12/12/2002)
UpcomingEvents (12/12/2002)

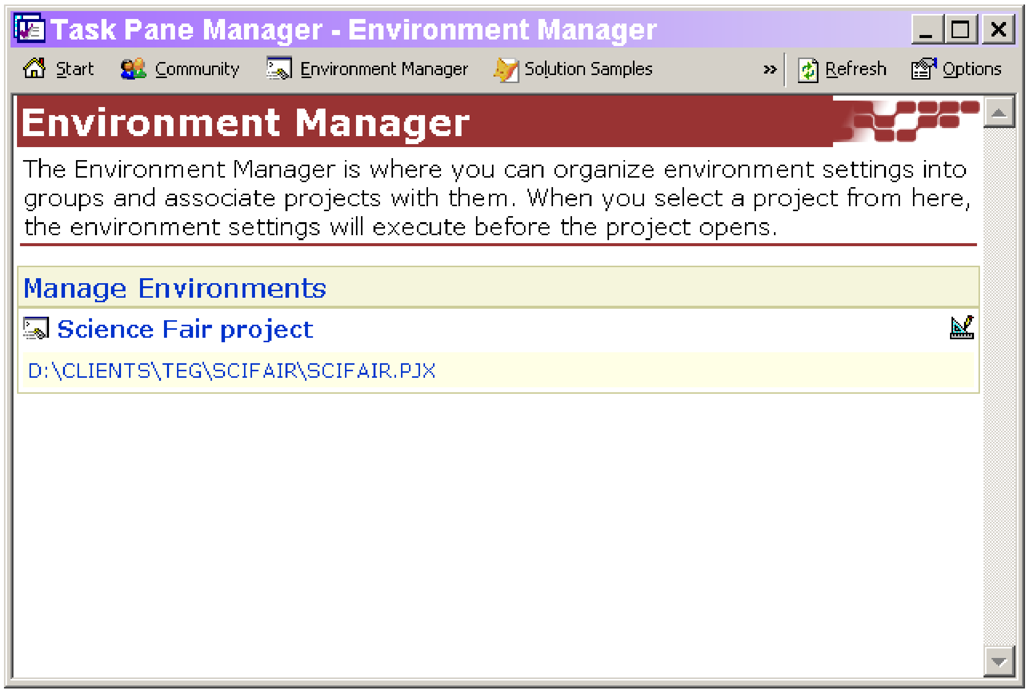
Figure 6: Joining the community—The Community pane provides quick access to VFP support sites and a listing of VFP news.

The bottom section (News) is a list of items drawn from one or more of the FoxCentral, Universal Thread and FoxPro Wiki websites. The Options window lets you specify which of them to use, as well as additional parameters for each of the three.

The Environment Manager pane

For those who work on more than one project at the same time, making sure you have the right settings in place when you work on a given project is a perennial problem. The Environment Manager pane (Figure 7) is designed to solve that problem. Click on the Manage Environments link to open the Environment Manager (Figure 8). (You can open the Environment Manager outside the Task Pane Manager by executing DO HOME() + "EnvMgr.APP".) In that dialog, you can create and edit what Microsoft calls environment sets.

Each environment set is listed in the Environment Manager pane. When you click on the name of the set (such as "Science Fair project" in Figure 7), those settings are applied. When you click on a project listed under a set, the settings are applied and the specified project is opened. Click on the design icon at the right to open the Environment Manager with this set selected for editing.



Environment Manager

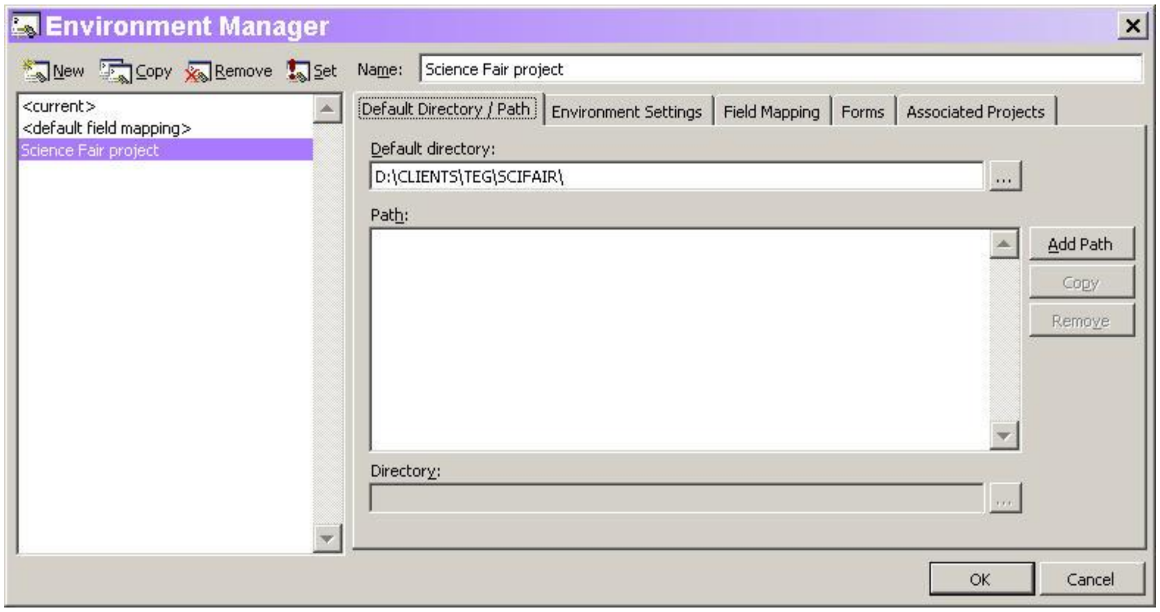
The Environment Manager is where you can organize environment settings into groups and associate projects with them. When you select a project from here, the environment settings will execute before the project opens.

Manage Environments

Science Fair project

D:\CLIENTS\TEG\SCIFAIR\SCIFAIR.PJX

Figure 7: Managing settings—The Environment Manager pane lets you name a group of settings and, optionally, associate it with one or more projects.



Name: Science Fair project

<current>
<default field mapping>
Science Fair project

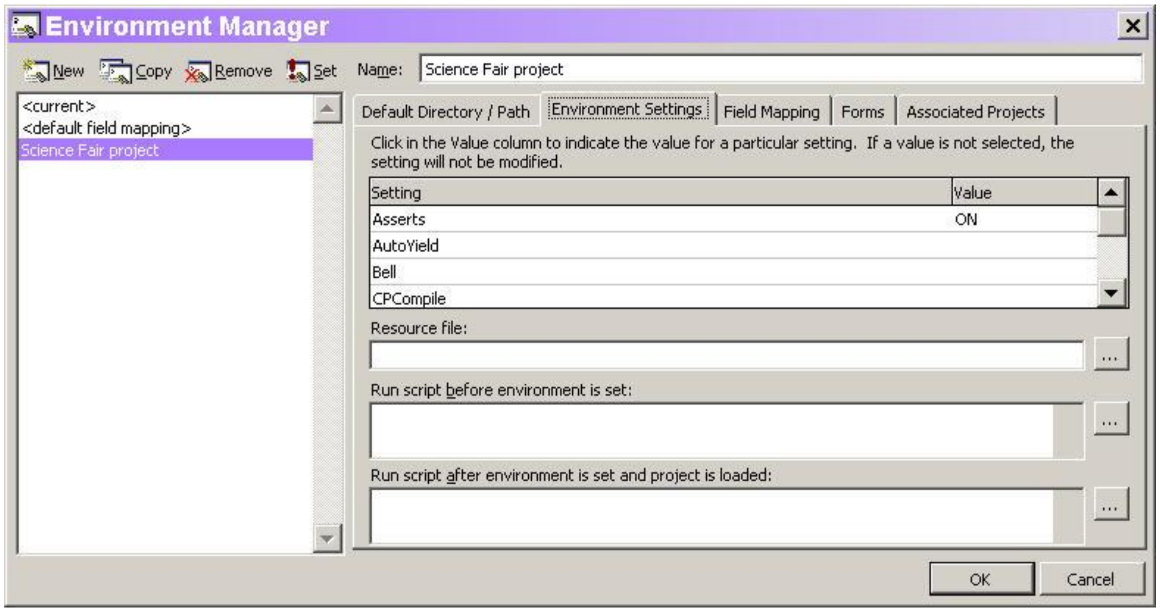
Default directory: D:\CLIENTS\TEG\SCIFAIR\

Path:

Directory:

Figure 8: The Environment Manager—This tool lets you create environment sets, which are named groups of settings, including path, default directory, and many SET commands.

To create an environment set, click New in the Environment Manager. Fill in the name, then specify the desired settings. The Default Directory / Path page shown in Figure 8 includes only the default directory and path, which default to the current settings. The Environment Settings page (Figure 9) lets you specify values for the most common SET commands, indicate the resource file (by default, FoxUser.DBF) to use, and specify scripts to run before and after applying the other settings. In each case, the script can be any VFP code, allowing you to handle items not directly included in the Environment Manager. (For example, you might open the Toolbox and point it to a specific data file or issue additional SET commands.) The ellipsis button next to each script opens a larger editing window to make it easier to enter your script code.



Name: Science Fair project

<current>
<default field mapping>
Science Fair project

Click in the Value column to indicate the value for a particular setting. If a value is not selected, the setting will not be modified.

Setting	Value
Asserts	ON
AutoYield	
Bell	
CPCompile	

Resource file:

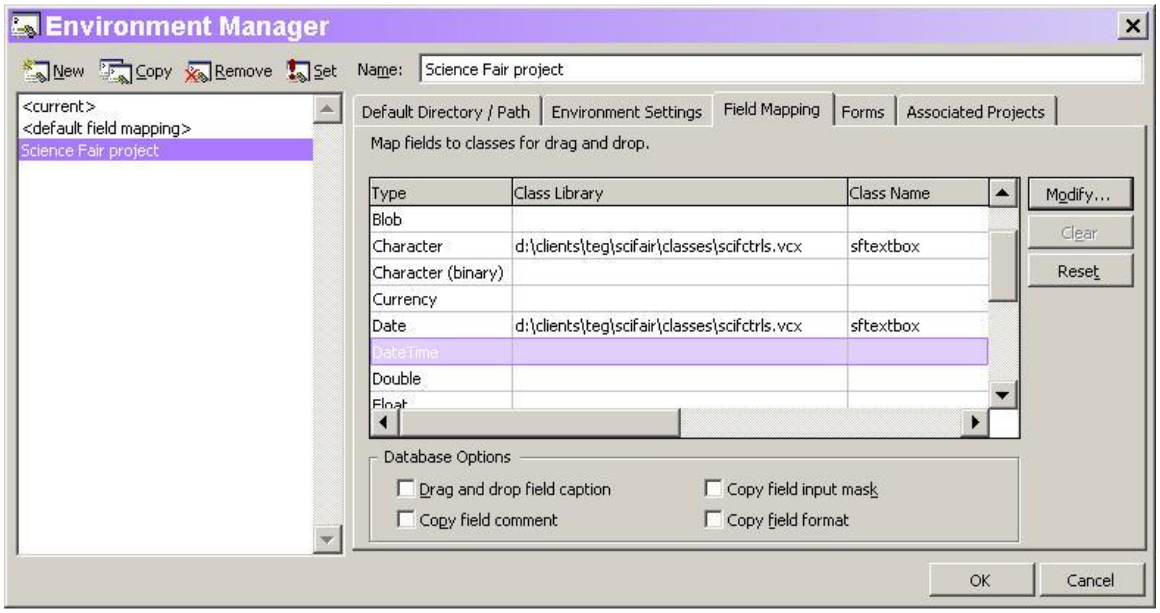
Run script before environment is set:

Run script after environment is set and project is loaded:

Figure 9: Specifying settings—This page of the Environment Manager includes a number of SET commands, plus the resource file and scripts to run before and after applying the other settings.

VFP 5 introduced the idea of field mapping, specifying the class and class library to use for each data type when you drag and drop in the Form Designer. Ever since, making sure you're using the right mappings for a given project has been a problem. In VFP 9, the Environment Manager has been enhanced to handle the problem for you. Use the Field Mapping page (Figure 10) to specify mappings for this environment set.

Be aware that the current field mappings are stored in the Registry. If you have two instances of VFP open, and set an environment in one, the field mappings for the other change, as well.


Environment Manager

New Copy Remove Set Name: Science Fair project

<current>
<default field mapping>
Science Fair project

Default Directory / Path | Environment Settings | Field Mapping | Forms | Associated Projects

Map fields to classes for drag and drop.

Type	Class Library	Class Name
Blob		
Character	d:\clients\teg\scifair\classes\scifctrls.vcx	sftextbox
Character (binary)		
Currency		
Date	d:\clients\teg\scifair\classes\scifctrls.vcx	sftextbox
DateTime		
Double		
Float		

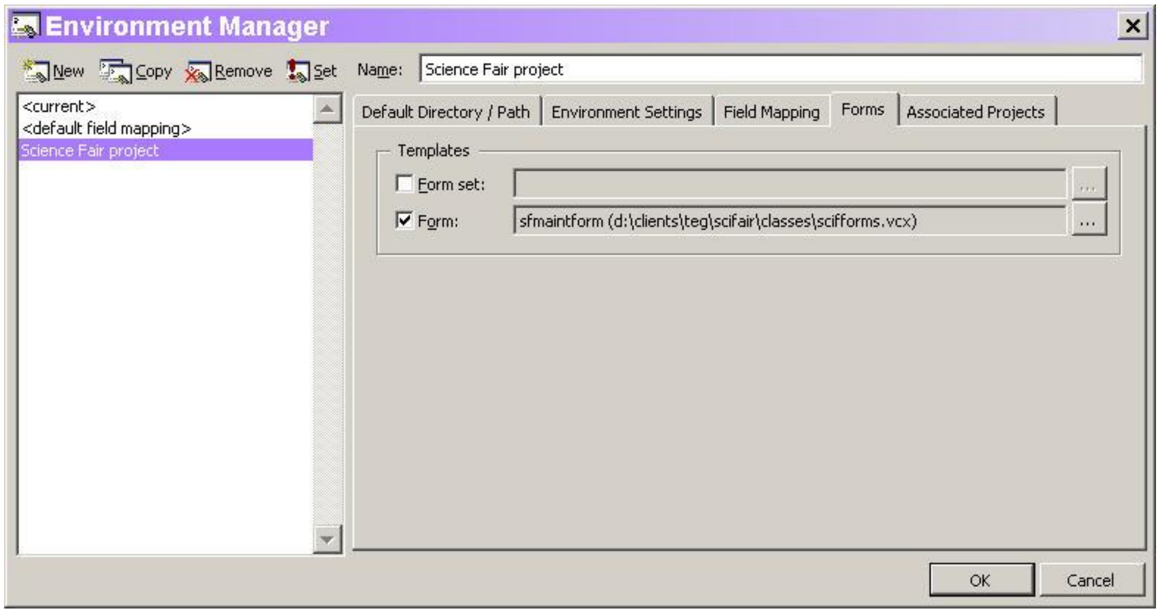
Modify... Clear Reset

Database Options
- Drag and drop field caption
- Copy field comment
- Copy field input mask
- Copy field format

OK Cancel

Figure 10. Choosing classes—The Field Mapping page lets you indicate what class to use for each data type.

The Forms page (Figure 11), also new in VFP 9, handles the mapping problem at the form level. You can specify a class to use each time you create a new form. Interestingly, forms do not suffer from the same problem as fields; two instances of VFP can have different default forms set.


Environment Manager

New Copy Remove Set Name: Science Fair project

<current>
<default field mapping>
Science Fair project

Default Directory / Path | Environment Settings | Field Mapping | Forms | Associated Projects

Templates
- Form set:
- [✓] Form: sfmaintform (d:\clients\teg\scifair\classes\sciforms.vcx)

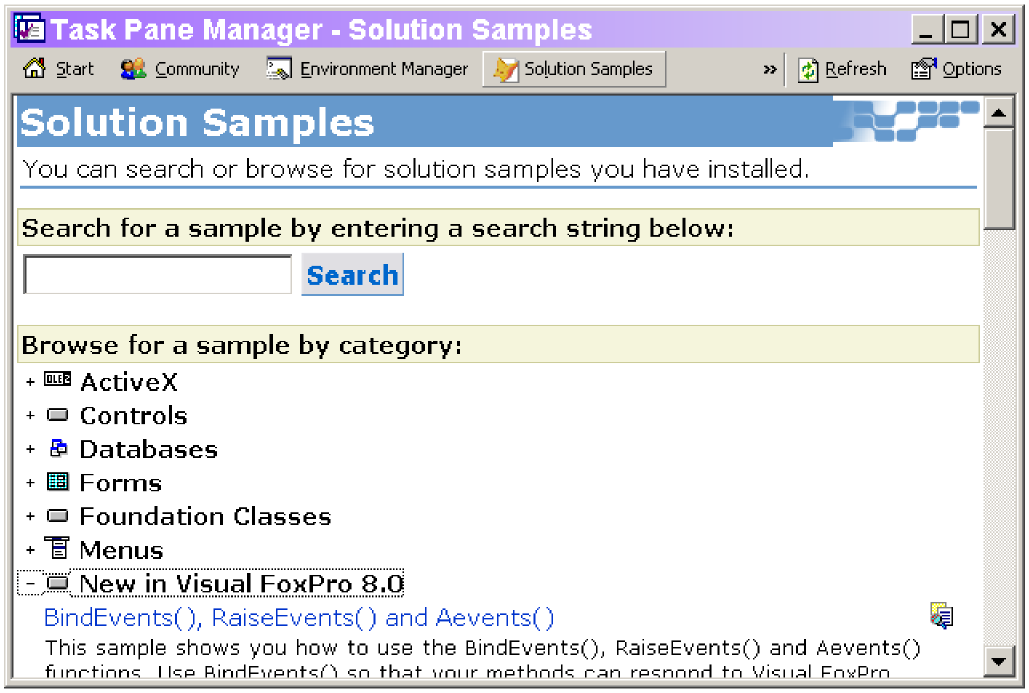
OK Cancel

Figure 11. Specifying a form class—The Forms page of the Environment Manager lets you indicate what class to use for new forms with this environment set.

Use the Associated Projects page to list the projects for which this environment set applies. All the projects associated with a specified environment set are listed beneath it in the Environment Manager pane.

The Solution Samples pane

Since VFP 5, Microsoft has included a fairly comprehensive set of granular examples, each designed to show a single feature. The hard part has been getting people to pay attention to these Solution Samples. So, starting in VFP 8, they get a task pane of their own. This pane (shown in Figure 12) includes searching, the ability to install new samples as they become available, and a section to show links to websites that offer additional samples.



Solution Samples

You can search or browse for solution samples you have installed.

Search for a sample by entering a search string below:

Browse for a sample by category:

+ ActiveX
+ Controls
+ Databases
+ Forms
+ Foundation Classes
+ Menus
- New in Visual FoxPro 8.0

BindEvents(), RaiseEvents() and Aevents()

This sample shows you how to use the BindEvents(), RaiseEvents() and Aevents() functions. Use BindEvents() so that your methods can respond to Visual FoxPro

Figure 12: Lots of examples—The Solution Samples pane offers quick access to well over 100 examples that demonstrate VFP features.

From the pane, you can both run the individual examples and display the underlying code. The samples include more than a dozen that demonstrate new features of VFP 8 and nearly 20 for new features of VFP 9.

One secret for this pane: When you search, the results are listed beneath the search criteria. To remove the listing, clear out the textbox and click Search.

From time to time, new samples may become available (most likely from Microsoft's VFP website). They'll be supplied as a file called Manifest.XML. To add them to the pane, click the Install Sample button and point to the file. (Sadly, as of this writing, no websites are offering additional samples in the specified format. When Microsoft did make additional VFP 8 samples available, they didn't package them for this pane.)

The XML Web Services Pane

Web Services are a key technology direction for Microsoft, and VFP is well-equipped to go in this direction, with the ability to both create and consume (use) web services. The XML Web Services pane (Figure 13) provides access to a variety of tools and resources to help you in building and using web services.

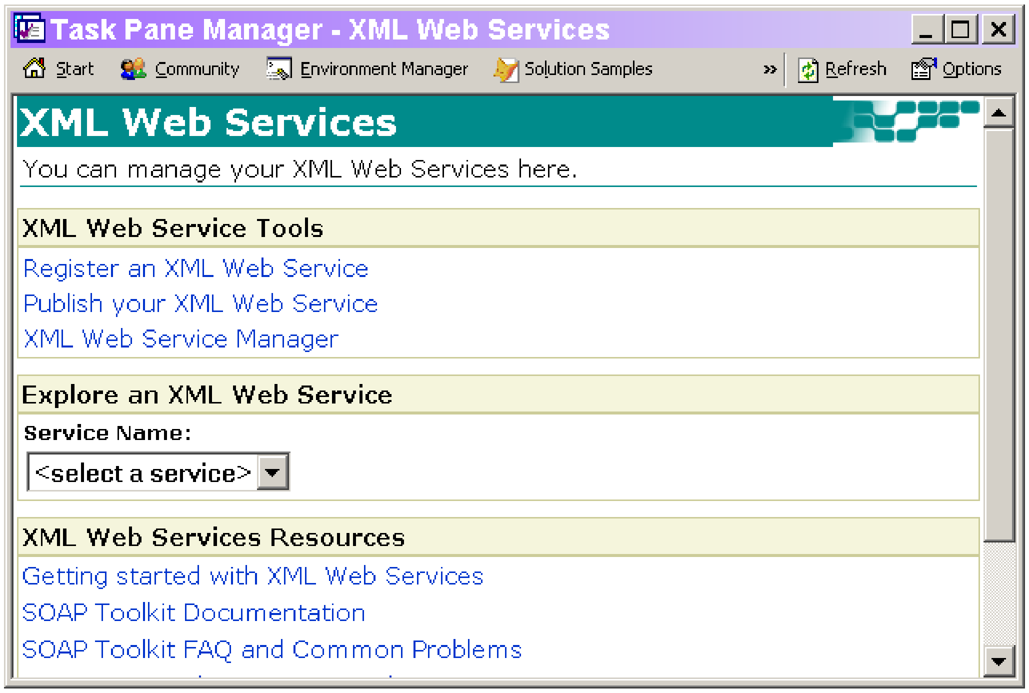

Task Pane Manager - XML Web Services
Start
Community
Environment Manager
Solution Samples
Refresh
Options
XML Web Services
You can manage your XML Web Services here.
XML Web Service Tools
Register an XML Web Service
Publish your XML Web Service
XML Web Service Manager
Explore an XML Web Service
Service Name:
<select a service>
XML Web Services Resources
Getting started with XML Web Services
SOAP Toolkit Documentation
SOAP Toolkit FAQ and Common Problems

Figure 13: Working with web services—This pane is a one-stop shop for web services tools and resources.

This pane has three sections. The XML Web Service Tools section has links to three key tools for working with web services. The first is the XML Web Services Registration dialog that lets you register a web service. Doing so provides IntelliSense for it and adds it to the Toolbox. Second is the XML Web Services Publisher, which lets you make web services you create available to others. The third is the XML Web Service Manager (shown in Figure 14), a new tool that lists registered web services and lets you add and delete them, as well as determine whether they're listed in the Toolbox. Each of these tools is quite capable, but they’re beyond the scope of this session.

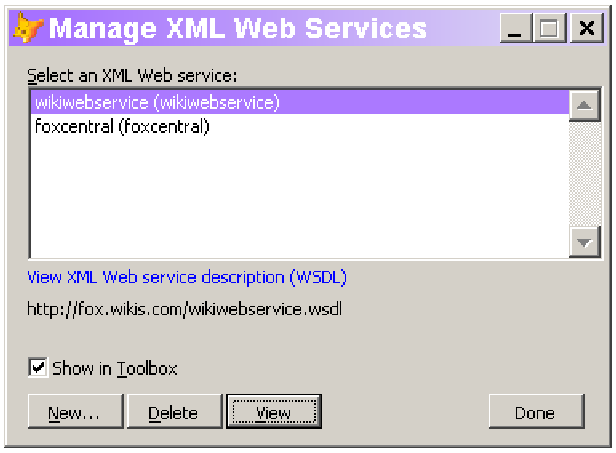
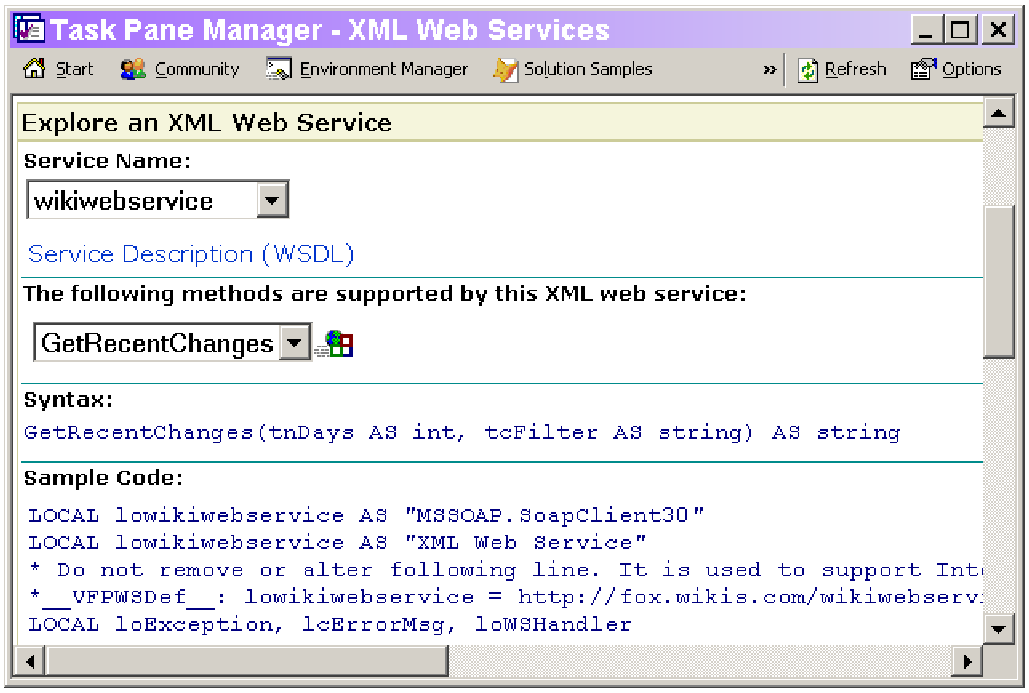

Manage XML Web Services
Select an XML Web service:
wikiwebservice (wikiwebservice)
foxcentral (foxcentral)
View XML Web service description (WSDL)
http://fox.wikis.com/wikiwebservice.wsdl
Show in Toolbox
New...
Delete
View
Done

Figure 14: XML Web Services Manager—This new tool offers quick access to all registered web services.

The second section of the XML Web Services pane provides help in using the web services you've registered. When you choose a web service from the dropdown (which appears only after you've registered at least one web service), the section expands (Figure 15) to include a dropdown of that web service's methods and sample code for using the web service.

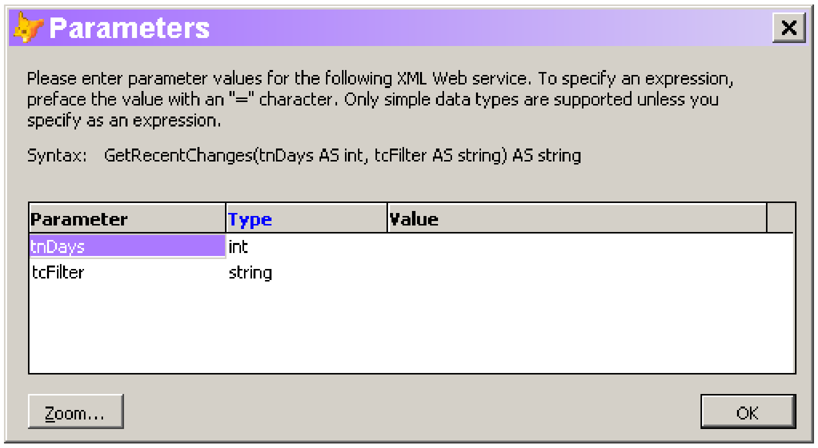

Task Pane Manager - XML Web Services
Start
Community
Environment Manager
Solution Samples
Refresh
Options
Explore an XML Web Service
Service Name:
wikiwebservice
Service Description (WSDL)
The following methods are supported by this XML web service:
GetRecentChanges
Syntax:
Sample Code:

```
GetRecentChanges(tnDays AS int, tcFilter AS string) AS string
```

```
LOCAL lowikiwebservice AS "MSSOAP.SoapClient30"
LOCAL lowikiwebservice AS "XML Web Service"
* Do not remove or alter following line. It is used to support Int
*__VFPWSDef__: lowikiwebservice = http://fox.wikis.com/wikiwebserv
LOCAL loException, lcErrorMsg, loWSHandler
```

Figure 15: Exploring a web service—When you choose a web service from the dropdown, the Explore an XML Web Service section expands. Click the icon next to the methods dropdown to test that method.

The icon next to the dropdown listing the methods lets you test the chosen method. When you click it, a dialog opens (Figure 16) for you to specify the parameters to pass to the method. Once you do so, another window opens showing the results from the method call. You can see the results either as XML or in a grid.



Please enter parameter values for the following XML Web service. To specify an expression, preface the value with an "=" character. Only simple data types are supported unless you specify as an expression.

Syntax: GetRecentChanges(tnDays AS int, tcFilter AS string) AS string

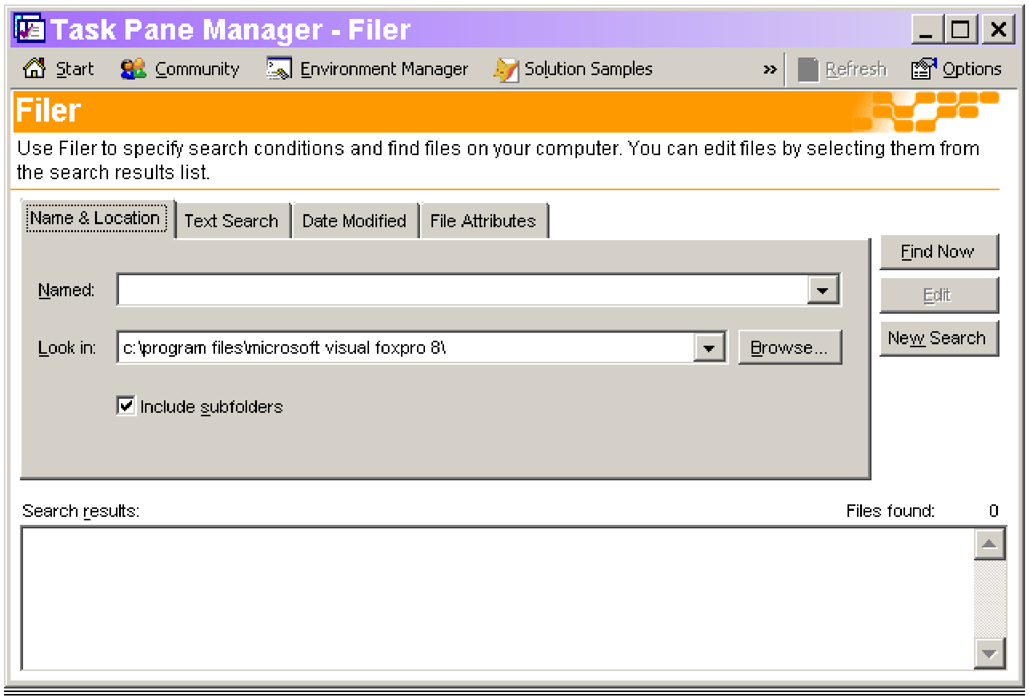
Parameter	Type	Value
tnDays	int	
tcFilter	string	

Figure 16: Testing a web service method—When you test a method, this dialog allows you to specify the parameter values to pass. Click OK to pass the specified values and execute the method.

The final section of the XML Web Services pane provides links to a variety of resources for building and using web services. They include relevant sections of the VFP Help file, documentation for the SOAP toolkit, some MSDN articles, and more.

The Filer pane

The Filer tool dates back to FoxPro 2.0, where it was introduced to provide an easy way to find files meeting various criteria. The functionality hasn't changed much over the years, but the form and face of the tool have. VFP 8 added another twist by housing Filer in a task pane (Figure 17). While the task pane version looks different than the stand-alone version (which is still available), the functionality is the same.



Use Filer to specify search conditions and find files on your computer. You can edit files by selecting them from the search results list.

Named:

Look in: c:\program files\microsoft visual foxpro 8\

☑ Include subfolders

Search results:

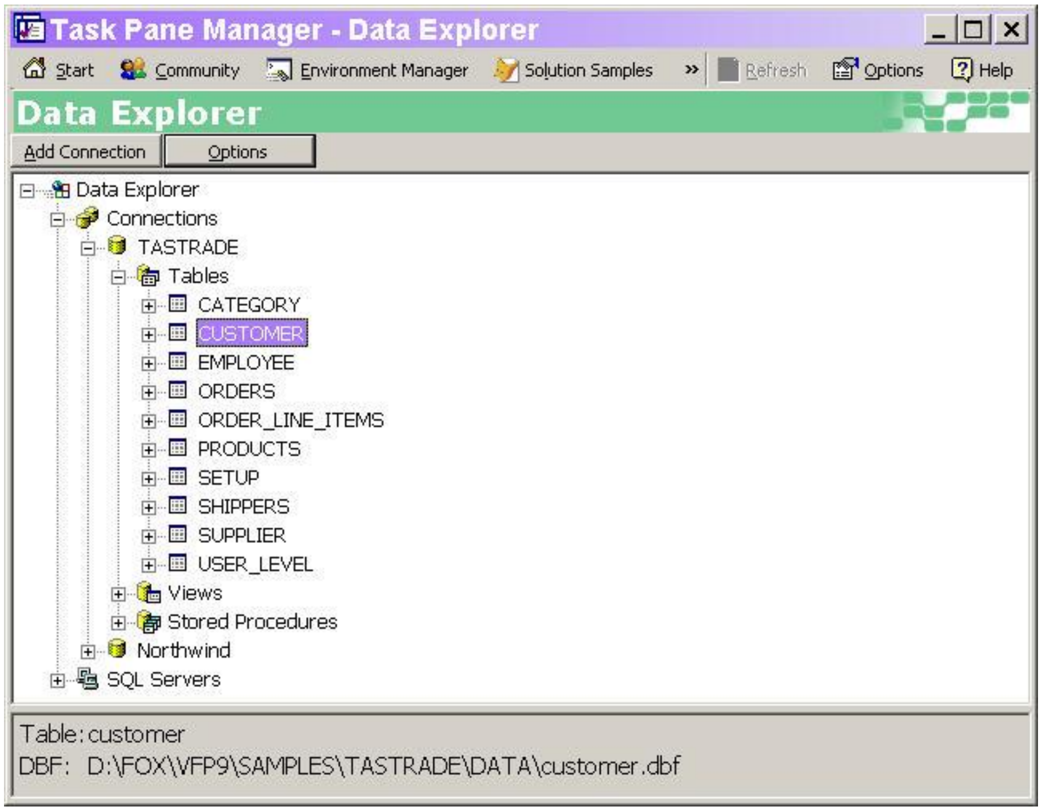
Files found: 0

Figure 17: Searching for files—The Filer tool can be accessed through the Task Pane Manager to search for files meeting specified criteria.

VFP 8 also introduced the Code References tool that provides much of the same functionality as the Filer, but better suited to VFP development.

The Data Explorer pane

With more and more VFP applications accessing data from other sources, the Data Explorer pane was added in VFP 9 (it’s the only new built-in pane in VFP 9). This tool (Figure 18) lets you collect information about all the databases on your network. For each database listed, you can see the tables, views and stored procedures, and even run queries created on the fly. (Like the Environment Manager, the Data Explorer can be run outside the Task Pane Manager: DO HOME() + "DataExplorer". The stand-alone Data Explorer can be docked.)

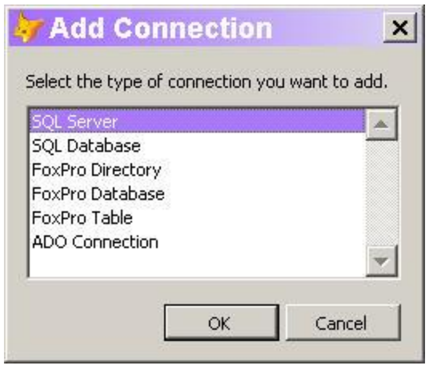


```
Data Explorer
  Connections
    TASTRADE
      Tables
        CATEGORY
        CUSTOMER
        EMPLOYEE
        ORDERS
        ORDER_LINE_ITEMS
        PRODUCTS
        SETUP
        SHIPPERS
        SUPPLIER
        USER_LEVEL
      Views
      Stored Procedures
    Northwind
  SQL Servers
```

Table: customer
DBF: D:\FOX\VFP9\SAMPLES\TASTRADE\DATA\customer.dbf

Figure 18: Examining data—The Data Explorer lets you drill into the databases available on your network.

To add an item (a database, a VFP table, or all the VFP databases in a directory), click the Add Connection button or right-click on the Connections node and choose Add Connection. The Add Connection dialog (Figure 19) appears. Choose the appropriate type. If you specify any FoxPro object, you're then prompted to point to the object (directory, database or table) using a standard dialog.



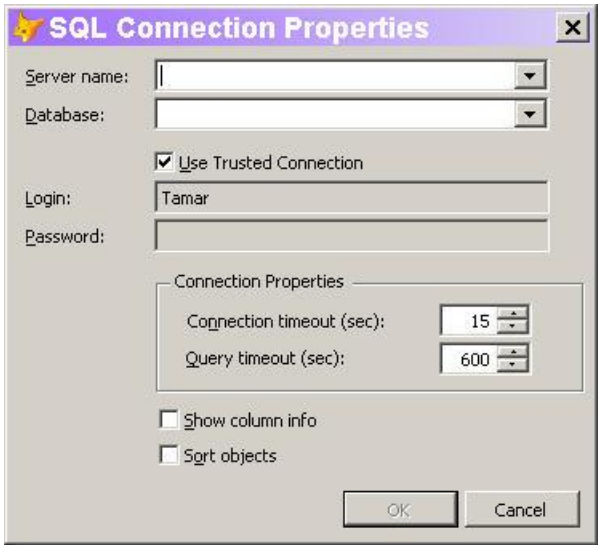
Select the type of connection you want to add.

- SQL Server
- SQL Database
- FoxPro Directory
- FoxPro Database
- FoxPro Table
- ADO Connection

Figure 19. Adding databases—This dialog appears when you click the Add Connection button. Specify the type of database you want to add.

If you specify SQL Server or SQL Database in the Add Connection dialog, the SQL Connection Properties dialog appears. (For SQL Server, the Database textbox is omitted.)

Use the dialog to specify the SQL Server and, if appropriate, database to add, along with the login and connection information.



[✓] Use Trusted Connection

Login: Tamar

Password:

Connection timeout (sec): 15

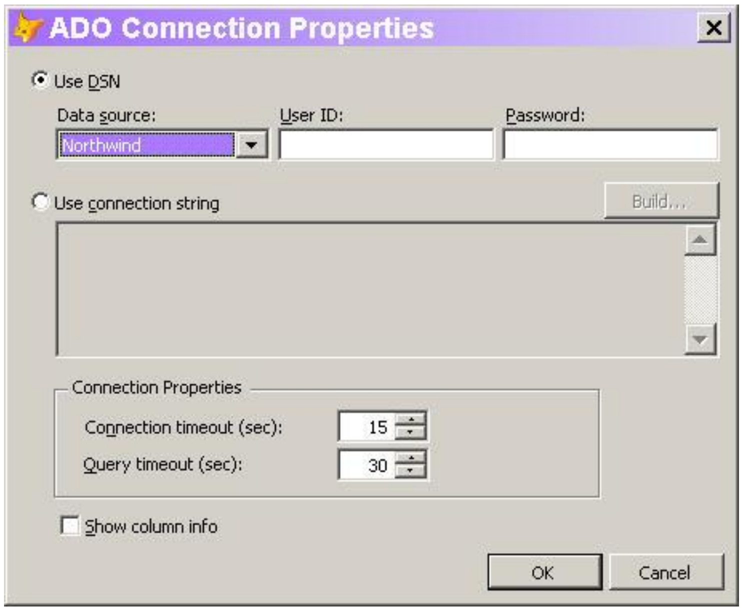
Query timeout (sec): 600

[ ] Show column info

[ ] Sort objects

Figure 20. Adding a SQL database—When you choose to add a SQL Server or SQL Database to the Data Explorer, the SQL Connection Properties dialog opens.

If you specify ADO Connection, the ADO Connection properties dialog (Figure 21) opens, allowing you to point to an existing connection using the DSN, or to specify a connection string.



(●) Use DSN

Data source: Northwind

( ) Use connection string

Connection timeout (sec): 15

Query timeout (sec): 30

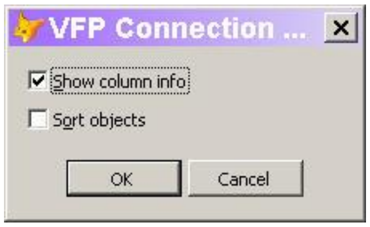
[ ] Show column info

Figure 21: Adding an ADO connection—You can specify either an existing data source or provide a connection string.

Once a database has been added, you can expand it to see Tables, Views and Stored Procedures sections, as in Figure 18. Each section has a number of options, accessible through its context menu. Most are self-explanatory, but a few are worth discussing.

Choose the Properties item for a FoxPro database, table or folder to open the VFP Connection Properties dialog. This dialog (shown in Figure 22) includes two check boxes: checking Show column info displays the type and size for the fields in each table; checking Sort objects organizes each section alphabetically. The SQL Connection Properties and the ADO Connection Properties dialogs also include the Show column info checkbox. For SQL data, you can also sort alphabetically.

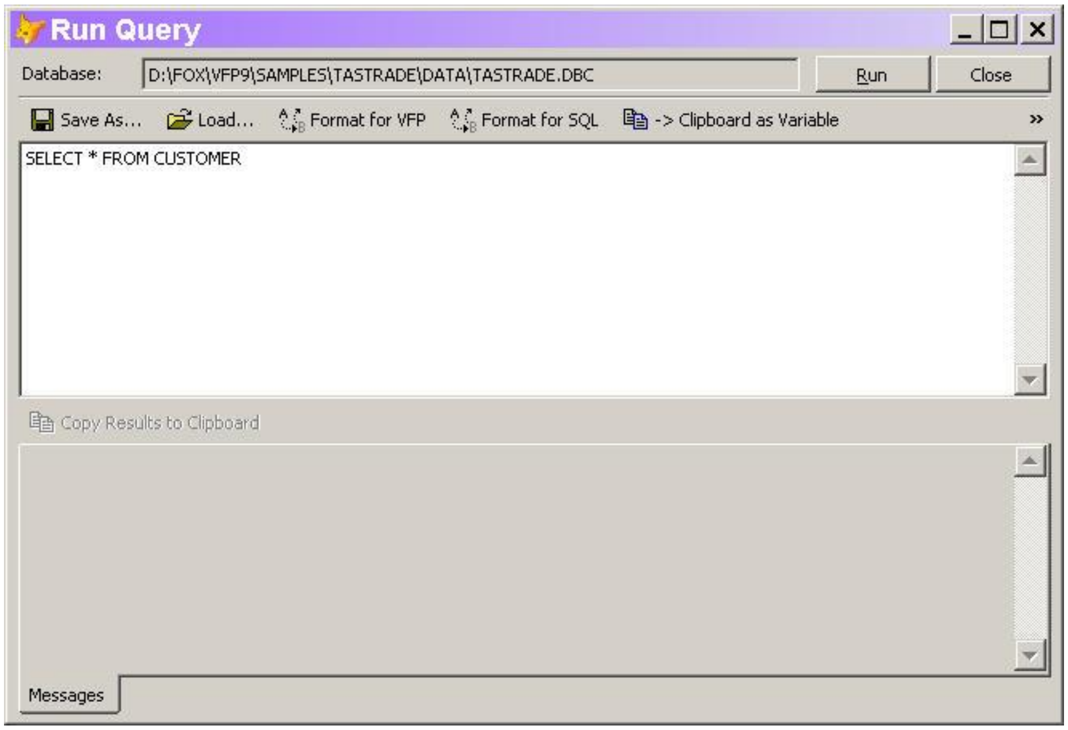
Note that in all cases, these settings apply to the entire item that is below Connections in the treeview, whether it's a database, table or directory. That is, you can't turn on column info for one table in a database and not the others, and if you've added a whole directory of VFP databases, these settings are all or nothing for the items in that directory.



☑ Show column info
☐ Sort objects

Figure 22: Customizing database information—The VFP Connection Properties dialog lets you determine whether fields show their type and size, and whether items are sorted alphabetically.

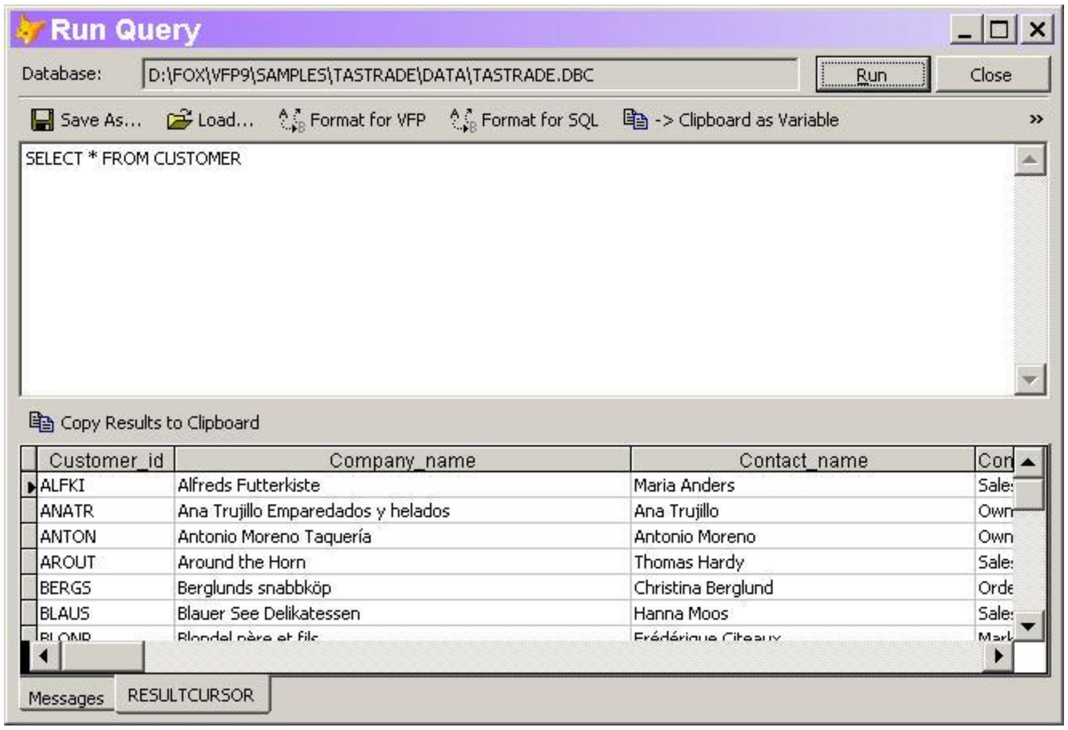
Many items include a Run Query item in their context menu. Choosing that item opens a window (Figure 23) that lets you create and run a query on the fly. Depending on the item selected when Run Query is chosen, the window may include a query to start with. For example, in Figure 23, Run Query was chosen from the shortcut menu of the Customer table. If a field's shortcut menu is used, the query selects only that field. Choosing Run Query for a stored procedure of a SQL database generates a command to execute that procedure.



```
SELECT * FROM CUSTOMER
```

Figure 23: Examining data—When you choose Run Query from a context menu, this window appears to let you create a query. It's initially populated with a query based on the item selected.

You can modify the query (or replace it entirely). When you click the Run button, the query is executed, and a page is added to the window, showing the results (as in Figure 24).



```
SELECT * FROM CUSTOMER
```

Customer_id	Company_name	Contact_name	Con
ALFKI	Alfreds Futterkiste	Maria Anders	Sale
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Own
ANTON	Antonio Moreno Taquería	Antonio Moreno	Own
AROUT	Around the Horn	Thomas Hardy	Sale
BERGS	Berglunds snabbköp	Christina Berglund	Orde
BLAUS	Blauer See Delikatessen	Hanna Moos	Sale
BLONP	Blondel père et fils	Frédérique Citeaux	Mark

Figure 24: Query results—When you choose the Run button, the query is executed. The ResultCursor pages shows the results of the query.

The Run Query window includes a button bar of what are referred to as "add-ins." These little tools let you do things like storing the query or the query results to the clipboard. You can modify the behavior of add-ins or create your own using the Add-in Manager described later in this section.

The SQL Servers section is populated automatically with the list of available servers. Expand that node to see the databases on that server. The items listed offer the same collection of options that SQL servers and databases include in the Connections node.

The context menu for the SQL Servers node includes Properties. When you choose it, you get a cut-down version of the SQL Connection Properties dialog (Figure 20) that allows you to set defaults.

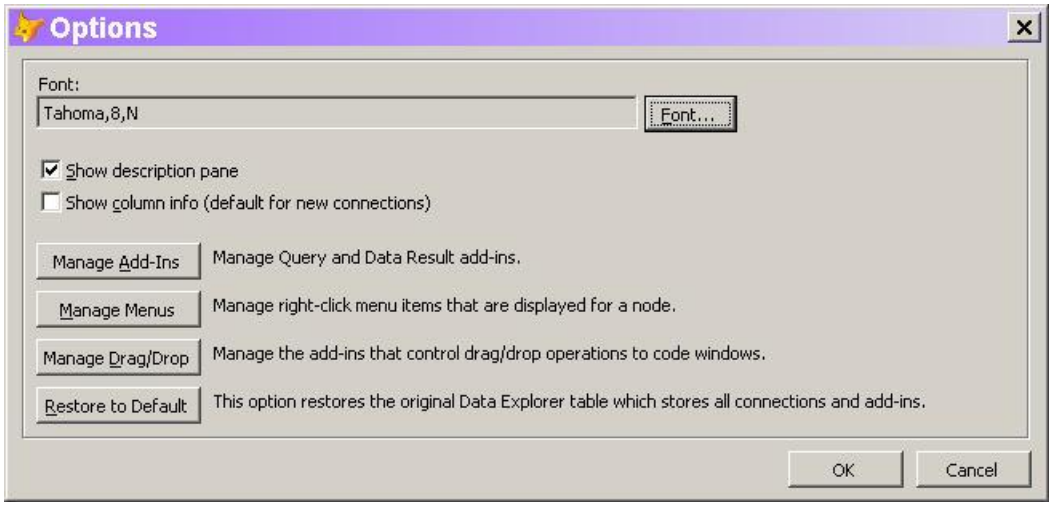
Most of the nodes support drag and drop into code windows and onto design surfaces. The code that's dropped varies depending on the node and whether you drop onto a code window or a designer.

When you drop into code windows, you generally get code to retrieve the appropriate object. For example, when you drag a VFP table into a code window, a query listing all fields of that table is dropped. When you drag a VFP field, you get a query for just that field. When you drag a SQL table or a table from an ADO connection, you get code that creates a cursoradapter, as well as the necessary ADO connection and recordset objects, and then uses the cursoradapter to retrieve all the data in the table.

Fewer of the nodes support dropping onto the design surfaces (such as the Form or Class Designer). Drop a non-VFP table onto a design surface and you get a grid. In the Form Designer, a cursoradapter is added to the data environment, as well.

Some of the drag and drop behavior is less useful—dragging the name of a VFP database drops that name and dragging the Tables node for any database drops the word "Tables." However, you can change any of the drag and drop behavior as well as adding additional behaviors using the Drag/Drop Manager described later in this section.

Finally, the Data Explorer has a built-in Options button that opens the dialog shown in Figure 25. The Task Pane's Options window doesn't include anything for the Data Explorer. No doubt the options are offered separately because the Data Explorer can be run outside the Task Pane Manager (by issuing DO HOME() + "DataExplorer.APP").



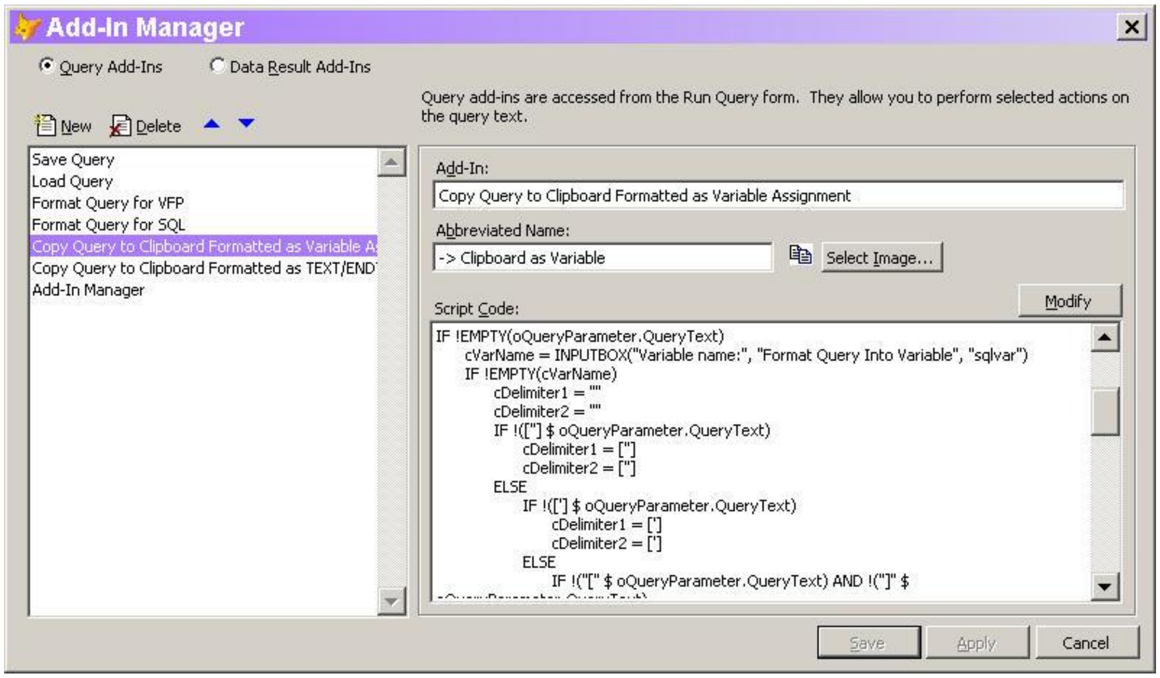
Font:
Tahoma,8,N

[✓] Show description pane
[ ] Show column info (default for new connections)

Manage Add-Ins	Manage Query and Data Result add-ins.
Manage Menus	Manage right-click menu items that are displayed for a node.
Manage Drag/Drop	Manage the add-ins that control drag/drop operations to code windows.
Restore to Default	This option restores the original Data Explorer table which stores all connections and add-ins.

Figure 25. Data Explorer Options—This dialog lets you change the font for the Data Explorer, customize the tool's behavior, and remove all the items you've added.

The Data Explorer is extremely extensible. You can change the behavior of many of its features as well as add new behaviors through the various "Manage" items in the Options dialog. For example, Figure 26 shows the dialog for managing the add-ins that appear in the Run Query window. They're all written in VFP code, so it's easy to change behavior if it doesn't give you what you want. Click New to create a new add-in, and then specify the name and script.



(●) Query Add-Ins ( ) Data Result Add-Ins

Query add-ins are accessed from the Run Query form. They allow you to perform selected actions on the query text.

Save Query
Load Query
Format Query for VFP
Format Query for SQL
Copy Query to Clipboard Formatted as Variable A
Copy Query to Clipboard Formatted as TEXT/END
Add-In Manager

Add-In:
Copy Query to Clipboard Formatted as Variable Assignment

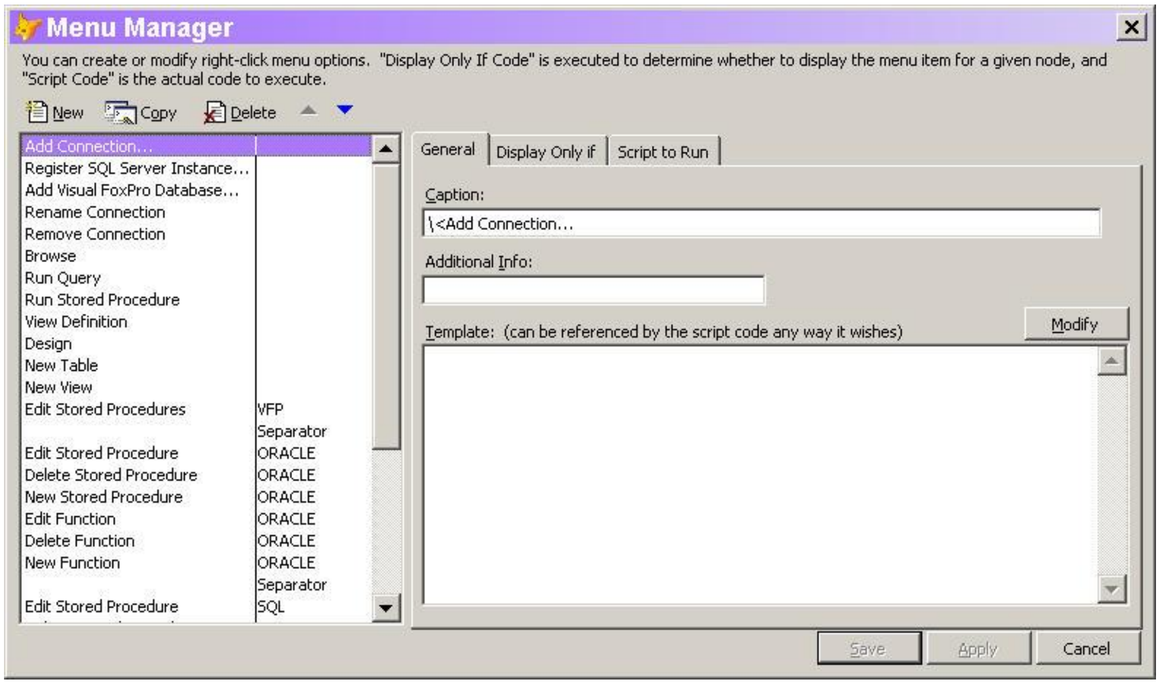
Abbreviated Name:
-> Clipboard as Variable

Script Code:

```
IF !EMPTY(oQueryParameter.QueryText)
    cVarName = INPUTBOX("Variable name:", "Format Query Into Variable", "sqlvar")
    IF !EMPTY(cVarName)
        cDelimiter1 = ""
        cDelimiter2 = ""
        IF !(["] $ oQueryParameter.QueryText)
            cDelimiter1 = ["]
            cDelimiter2 = ["]
        ELSE
            IF !(['] $ oQueryParameter.QueryText)
                cDelimiter1 = [']
                cDelimiter2 = [']
            ELSE
                IF !("[" $ oQueryParameter.QueryText) AND !("]" $
```

Figure 26. Customizing the Data Explorer—The various Manage buttons in the Options dialog open tools that let you change built-in behaviors and add new options.

Click Manage Menus to customize the shortcut menus in the tool. The Menu Manager (Figure 27) opens. For each item, you can specify the menu item, which shortcut menus contain the item, and code to run when the menu item is chosen. You can have different versions of the same item for different contexts. (See, for example, Edit Stored Procedures in the figure.)


Menu Manager

You can create or modify right-click menu options. "Display Only If Code" is executed to determine whether to display the menu item for a given node, and "Script Code" is the actual code to execute.

Add Connection...	
Register SQL Server Instance...	
Add Visual FoxPro Database...	
Rename Connection	
Remove Connection	
Browse	
Run Query	
Run Stored Procedure	
View Definition	
Design	
New Table	
New View	
Edit Stored Procedures	VFP
	Separator
Edit Stored Procedure	ORACLE
Delete Stored Procedure	ORACLE
New Stored Procedure	ORACLE
Edit Function	ORACLE
Delete Function	ORACLE
New Function	ORACLE
	Separator
Edit Stored Procedure	SQL

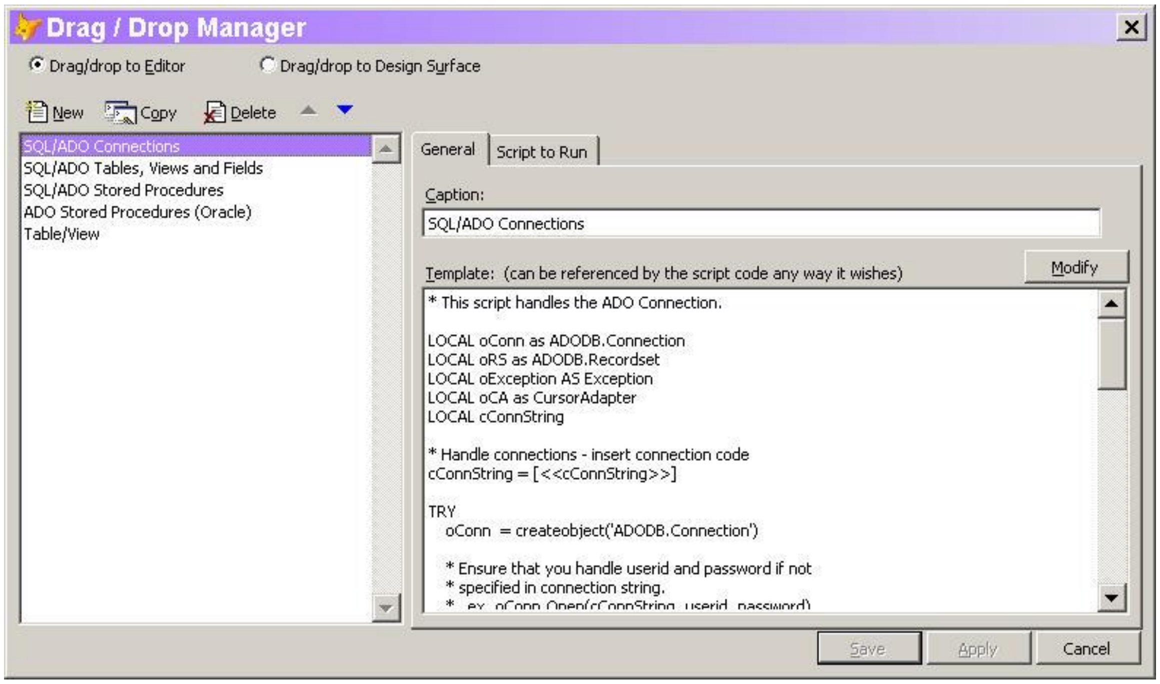
Caption: \<Add Connection...

Additional Info:

Template: (can be referenced by the script code any way it wishes)

Figure 27. The Menu Manager lets you control the contents of the Data Explorer's shortcut menus.

Manage Drag/Drop lets you decide what happens when you drag items from the Data Explorer into code windows. The Drag/Drop Manager (Figure 28) lets you specify which nodes the action applies, and template code and a script for those nodes. You can specify code window and design surface behavior separately.


Drag / Drop Manager

(•) Drag/drop to Editor ( ) Drag/drop to Design Surface

New Copy Delete

SQL/ADO Connections
SQL/ADO Tables, Views and Fields
SQL/ADO Stored Procedures
ADO Stored Procedures (Oracle)
Table/View

General | Script to Run

Caption: SQL/ADO Connections

Template: (can be referenced by the script code any way it wishes) Modify

```
* This script handles the ADO Connection.

LOCAL oConn as ADODB.Connection
LOCAL oRS as ADODB.Recordset
LOCAL oException AS Exception
LOCAL oCA as CursorAdapter
LOCAL cConnString

* Handle connections - insert connection code
cConnString = [<<cConnString>>]

TRY
   oConn = createobject('ADODB.Connection')

   * Ensure that you handle userid and password if not
   * specified in connection string.
   *   ex. oConn.Open(cConnString, userid, password)
```

Save Apply Cancel

Figure 28. Use the Drag/Drop Manager to control behavior when you drag and drop nodes into code windows or onto design surfaces.

The Minesweeper pane

The final pane supplied by Microsoft is just for fun. It's a VFP version of Minesweeper. It's missing some of the functionality of the Windows version, but includes the ability to set the size of the grid.

Actually, this pane has one other use. The code uses the Collection class added in VFP 8. You may want to examine the code to see how it works. See "VFP Controls panes" later in this document to learn how.

Extending the Task Pane Manager

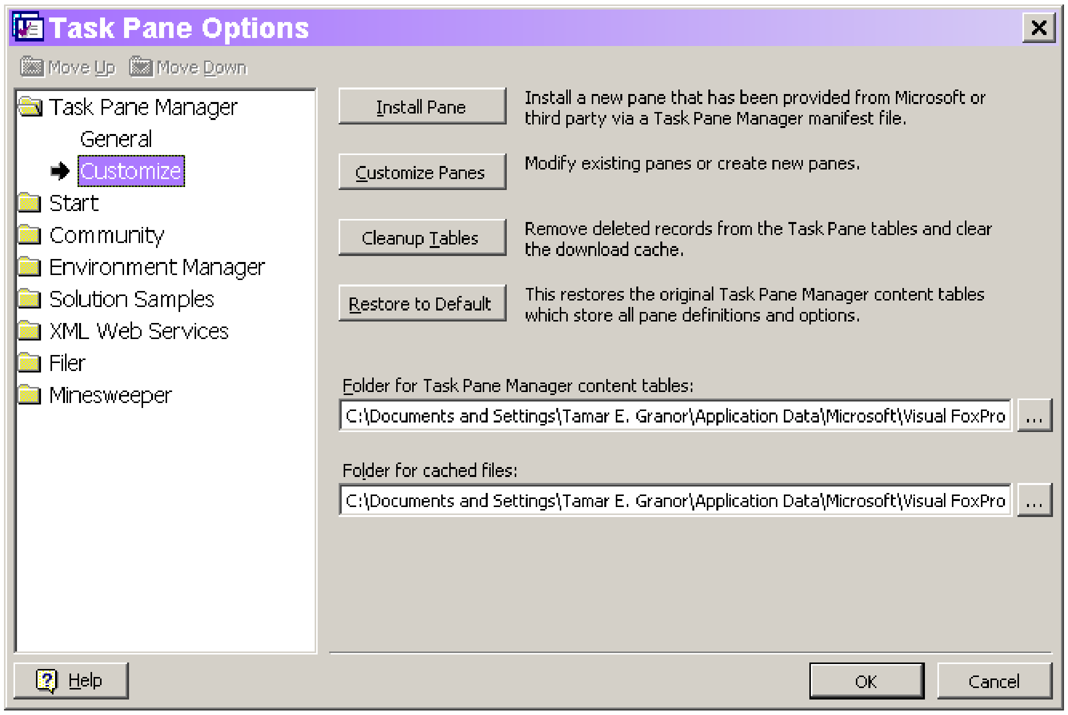
While the panes provided by Microsoft are useful (or fun), the real power of the Task Pane Manager is in the ability to add panes. There are two ways to do so. The simpler, by far, is to install a pane supplied by someone else. The hard way is to create your own.

Installing third-party panes

Before VFP 8 was released, I expected most of the vendors of tools for VFP to create task panes for their products. Each framework and developer tool has its own portal, with some of them modal. The Task Pane Manager provides an opportunity to unify all the tools we use into a single interface. However, to date, only one framework has done so.

Nonetheless, you'll find some task panes available for download at http://taskpane.leafe.com/.

Installing a pane supplied by someone else is a breeze. Open the Task Pane Manager and click the Options button. In the Options window, open the Task Pane Manager folder. Then choose Customize. On the page that appears (Figure 29), click Install Pane.



Install a new pane that has been provided from Microsoft or third party via a Task Pane Manager manifest file.

Modify existing panes or create new panes.

Remove deleted records from the Task Pane tables and clear the download cache.

This restores the original Task Pane Manager content tables which store all pane definitions and options.

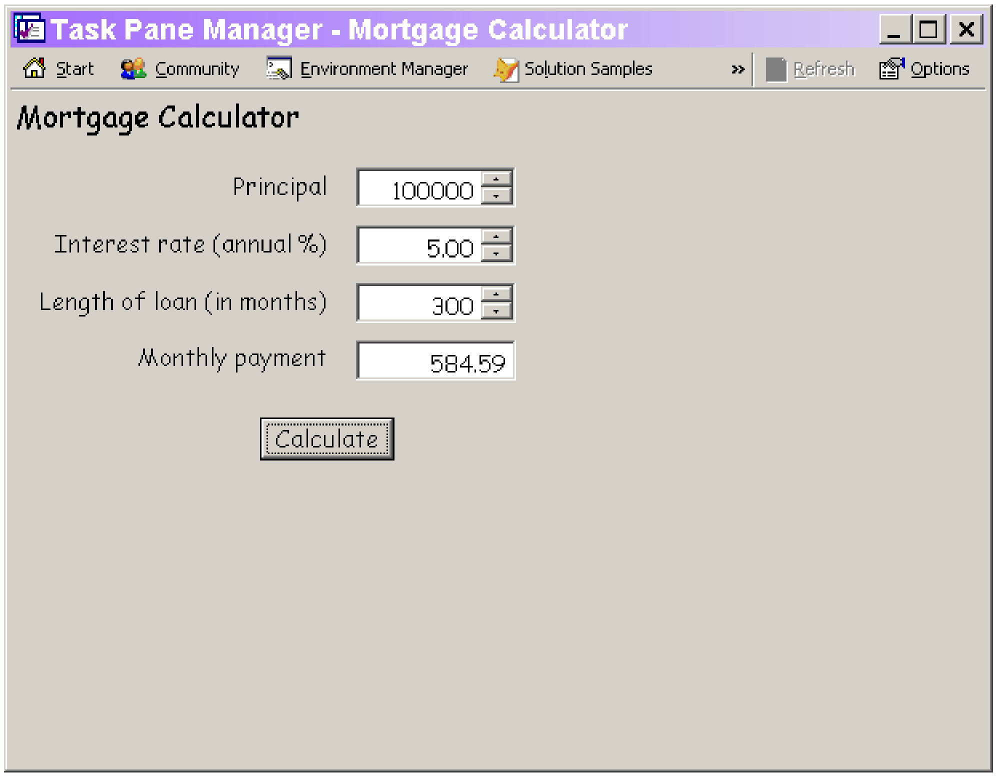
Folder for Task Pane Manager content tables: C:\Documents and Settings\Tamar E. Granor\Application Data\Microsoft\Visual FoxPro

Folder for cached files: C:\Documents and Settings\Tamar E. Granor\Application Data\Microsoft\Visual FoxPro

Figure 29: Adding and creating panes—This page of the Task Pane Options window lets you install third-party panes and create your own panes.

Task panes are distributed as XML files, so you're prompted to point to the appropriate XML file. Once you do so, it appears in the Task Pane Options window, and focus lands on that pane, so you can set any options it may have. The pane is also added to the Task Pane Manager, of course.

The materials for this session include Mortgage_Calculator.XML, the installation file for a simple pane that computes mortgage payments (Figure 30).



Mortgage Calculator

Principal 100000

Interest rate (annual %) 5.00

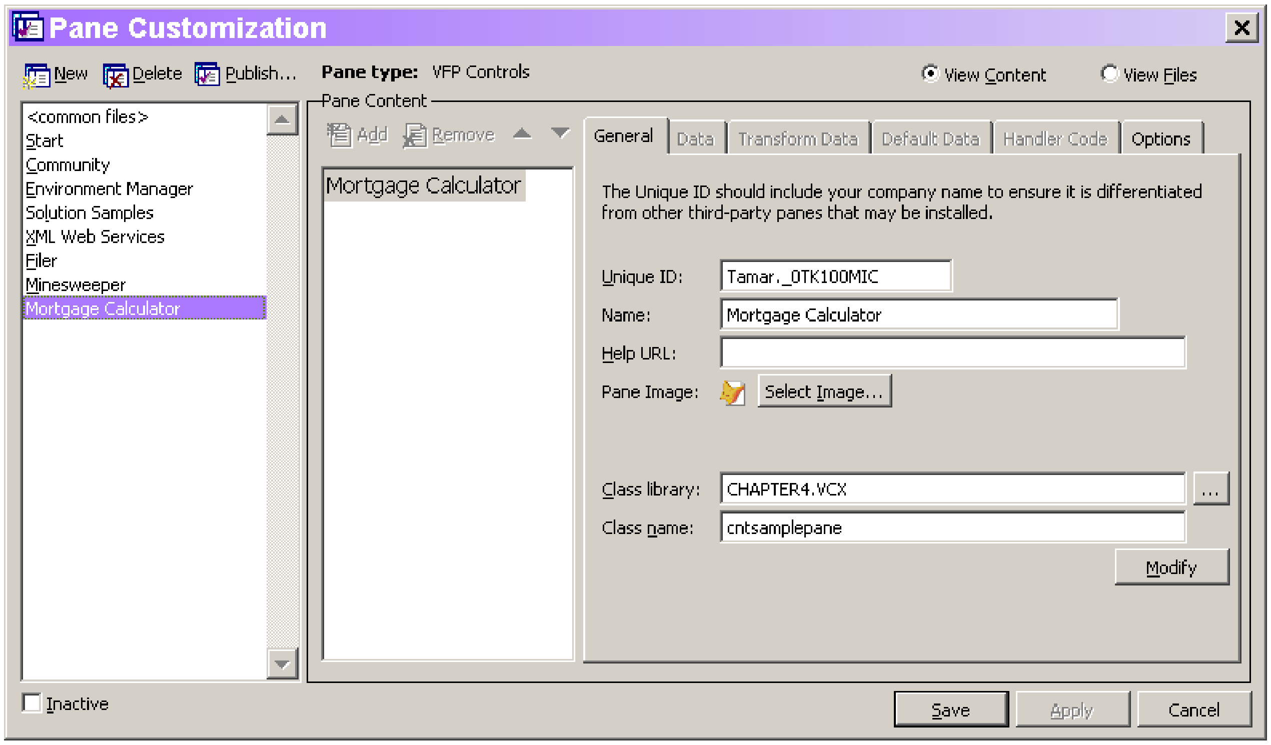
Length of loan (in months) 300

Monthly payment 584.59

Figure 30: Mortgage Calculator pane—This pane, built with VFP code, computes loan payments.

Creating custom panes

You use the same page of the Task Pane Options window to start the process of creating your own panes, as well as to modify existing panes. In this case, click Customize Panes, which opens the Pane Customization window (Figure 31).



Pane type: VFP Controls

<common files>
Start
Community
Environment Manager
Solution Samples
XML Web Services
Filer
Minesweeper
Mortgage Calculator

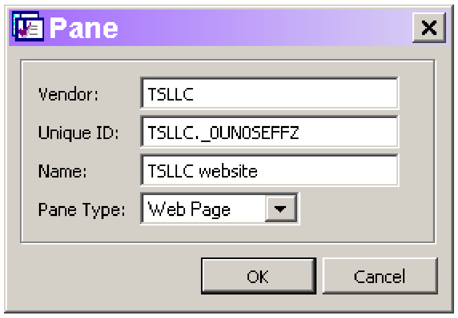
Mortgage Calculator

The Unique ID should include your company name to ensure it is differentiated from other third-party panes that may be installed.

Unique ID:	Tamar._0TK100MIC
Name:	Mortgage Calculator
Help URL:	
Pane Image:	
Class library:	CHAPTER4.VCX
Class name:	cntsamplepane

Figure 31: Creating and modifying panes—The Pane Customization window lets you modify existing panes and create new ones.

To create a new page, click the New button on the button bar. The Pane dialog (Figure 32) appears. In this dialog, you specify a "vendor" (your name or your company name), the name for the new pane, and the type of pane you're building.



Vendor:	TSLLC
Unique ID:	TSLLC._0UN0SEFFZ
Name:	TSLLC website
Pane Type:	Web Page

Figure 32: Creating a new pane—This dialog lets you specify basic information for a new task pane.

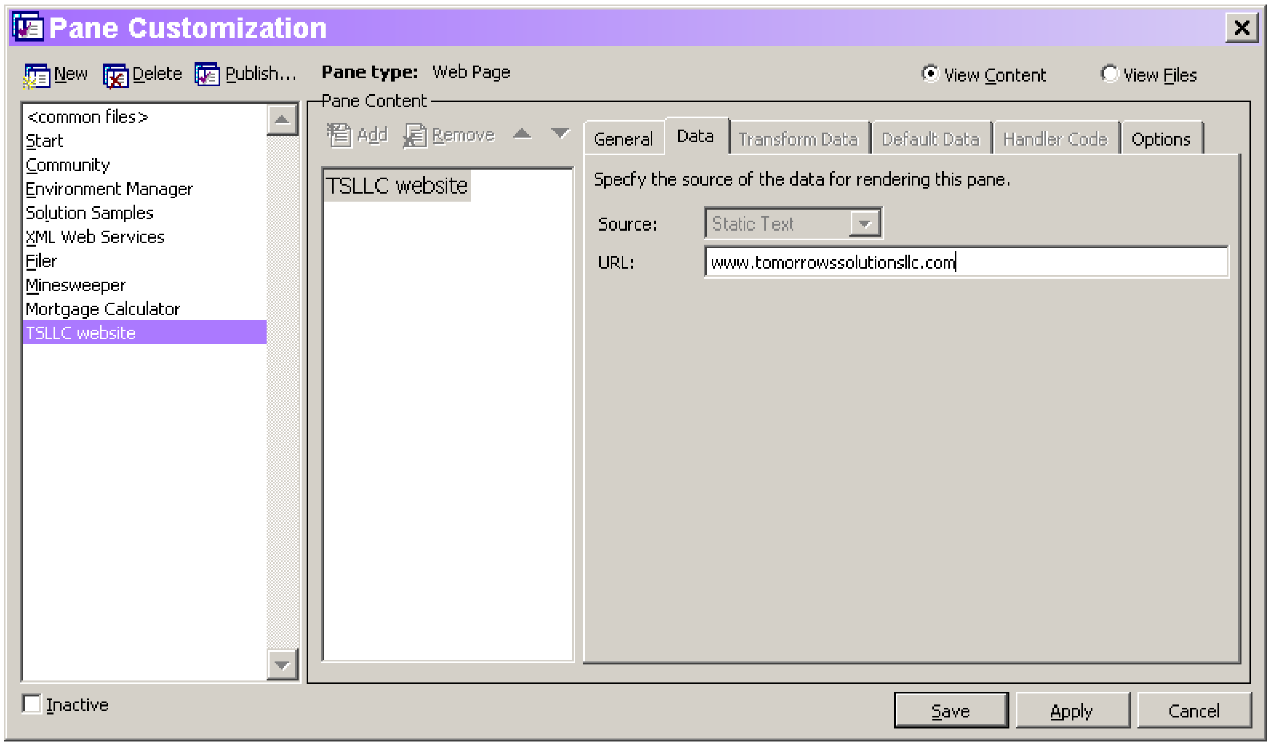
By default, the name you specify for Vendor is concatenated with the Unique ID to form the complete identifier for the pane. This combination increases the chances that every pane a user installs has a completely unique identifier. (The default value for Unique ID is generated using SYS(2015), but you can specify a different value if you want.)

The string you specify for Name is the pane's "friendly name." It appears in the Task Pane's button bar, in the title bar when the pane is displayed, and in the Task Pane Options window.

The tool supports four pane types: Web Page, HTML, XML, and VFP Controls. The type of a pane appears in the button bar of the Pane Customization window. (In Figure 31, the Mortgage Calculator pane is a VFP Controls pane.) We'll look at each type.

Web Page panes

The simplest type of pane to create is a Web Page. Web Page panes simply display a specified website in the Task Pane Manager. The only thing you need to specify for a Web Page pane is the URL of the website. You do so on the Data page of the Pane Customization dialog (Figure 33). The appearance of this page changes depending on the type of pane you're creating and on choices you make on the page.



Pane type: Web Page

<common files>
Start
Community
Environment Manager
Solution Samples
XML Web Services
Filer
Minesweeper
Mortgage Calculator
TSLLC website

TSLLC website

Specfy the source of the data for rendering this pane.

Source: Static Text

URL: www.tomorrowssolutionsllc.com

Figure 33: Specifying a Web Page pane—For this type of pane, the only required information is the URL of the website to display.

There's one other item you may want to specify for this type of pane, as well as for other panes—that's the image to associate with the pane. The image appears next to the pane name in the button bar. You specify it on the General page of the Pane Customization window (see Figure 31).

The pane defined in Figure 33 is included in the materials for this session as TSLLC_website.XML.

VFP Controls panes

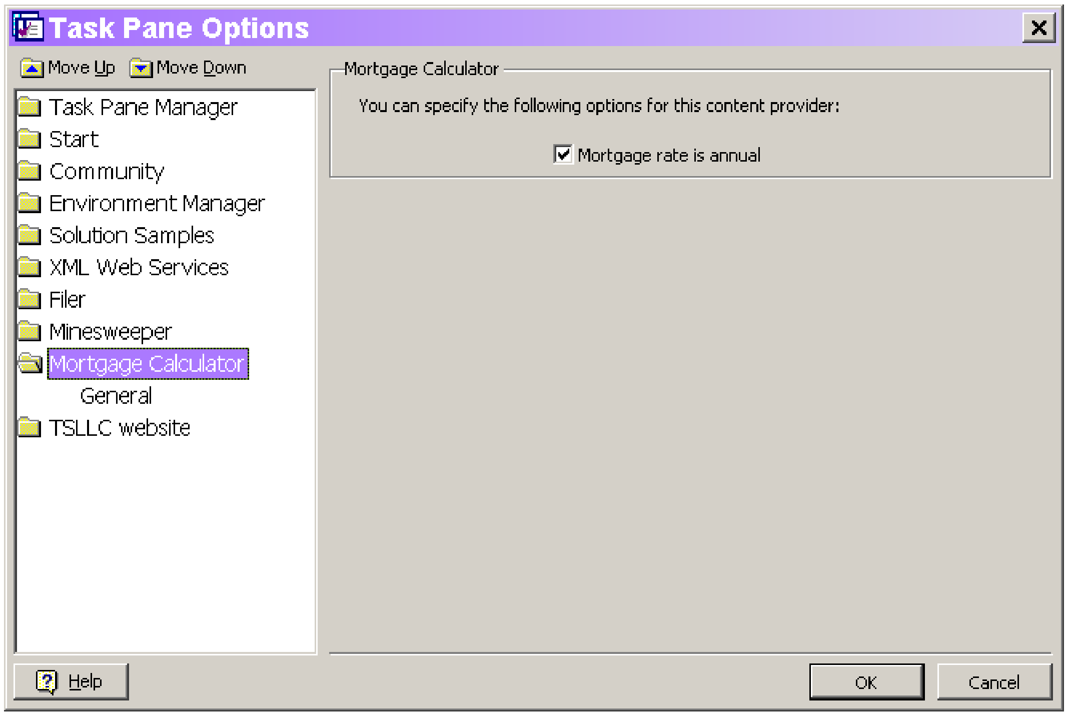
For VFP developers, panes based on VFP code are clearly the next easiest to create. The Mortgage Calculator in the session materials is a VFP Controls pane—we'll look at its code to see how to build this sort of pane.

For a VFP Controls pane, you specify a class and class library. The class should be subclassed from the PaneContainer class in the FoxPane.VCX class library that's part of the TaskPane project. (Unzip XSource.ZIP in the Tools folder, and then look in the VFPSource\TaskPane folder to find that class library.) While you probably can create a pane based on a different class, using PaneContainer ensures that all the necessary methods are present and that default behavior is provided.

To the subclass of PaneContainer, add the controls and code necessary for the functionality you want the pane to have. The Mortgage Calculator pane has a few labels and spinners to represent the principal, interest and length of the loan, plus a textbox to contain the payment amount and a button to perform the calculation. There's one custom method, ComputePayment, which does the calculation (using VFP's Payment() function). The Init method of the pane calls ComputePayment, as does the Click method of the Calculate button.

Defining Options

Panes can have options, items the user specifies that change the behavior of the pane. For example, the Minesweeper pane has an option for the size of the matrix. (To see it, open the Task Pane Options window, and click on Minesweeper.) The Mortgage Calculator pane has an option to indicate whether the interest rate specified is annual or monthly. Figure 34 shows this option in the Task Pane Options window.



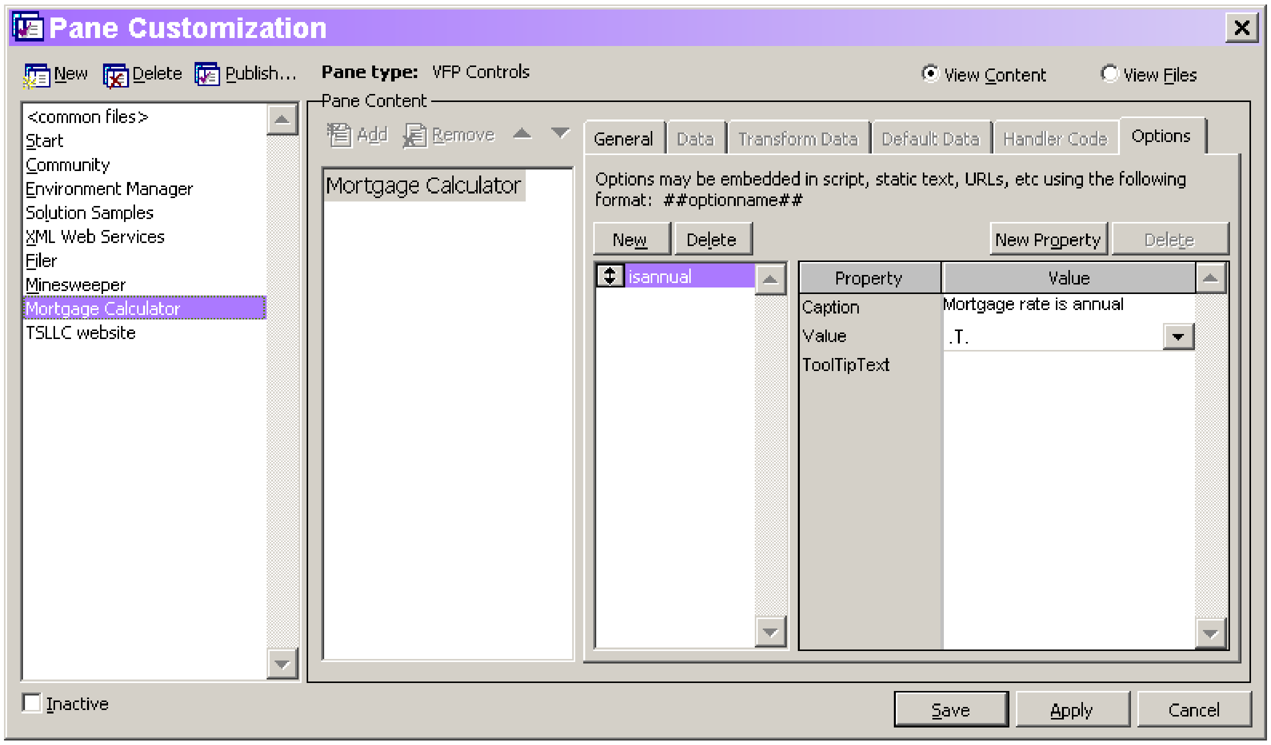
Mortgage Calculator

You can specify the following options for this content provider:

☑ Mortgage rate is annual

Figure 34: Pane option—When creating a custom pane, you can define options that the user can specify when using the pane.

Options are specified on the Options page of the Pane Customization window (Figure 35). To add an option item, click New; the New Option dialog (Figure 36) appears. Specify a name for the option, the type of control it should use (textbox, checkbox, spinner, or password textbox) and a caption for that control.



Pane type: VFP Controls

[✓ View Content] View Files

<common files>
Start
Community
Environment Manager
Solution Samples
XML Web Services
Filer
Minesweeper
Mortgage Calculator
TSLLC website

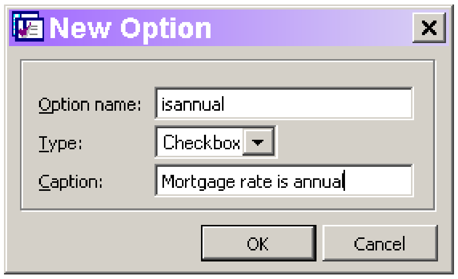
Mortgage Calculator

Options may be embedded in script, static text, URLs, etc using the following format: ##optionname##

isannual

Property	Value
Caption	Mortgage rate is annual
Value	.T.
ToolTipText	

Figure 35: Pane options—The Options page of the Pane Customization dialog defines options for the pane. The options specified here appear on the Options page for the specified pane.



Option name: isannual
Type: Checkbox
Caption: Mortgage rate is annual

Figure 36: Adding options—The New Option dialog lets you specify the name of an option and the control used for it.

Once an option has been defined, you can modify the control used to display it by setting properties of the control. The properties grid on the right-hand side of the Options page is initially populated with a few properties you're likely to want to change. You can add other properties to the list by clicking the New Property button above the grid. Then specify the name of the property and it's added to the grid. The Value column of the grid is editable, so you can set the properties.

Using Options

Of course, defining options wouldn't be very useful if there weren't a way to use them in figuring out what appears in a pane. There are actually two ways to access the value of a property; which one you use depends on the situation.

VFP code that you write for a pane (whether in a subclass of PaneContainer or in one of several other places used only for HTML and XML panes) generally receives one or more parameters. The one you need for accessing option values is oContent, a reference to an object based on the Content class defined in FoxPaneContent.PRG. The Content class has a GetOption method—pass it the name of the option and it returns the current value. (Option values are stored between VFP sessions.)

In the Mortgage Calculator pane, the OnRender method (inherited from PaneContainer) uses this code to get and apply the value of the IsAnnual option. Note that all option values are character, so the code must convert to the appropriate type:

```
LPARAMETERS oPane, oContent

LOCAL cResult

cResult = oContent.GetOption("isannual")

This.lIsAnnual = IIF(cResult = ".T.", .T., .F.)

IF This.lIsAnnual
  This.lblInterest.Caption = "Interest rate (annual %)"
ELSE
  This.lblInterest.Caption = "Interest rate (monthly %)"
ENDIF
```

The second way to use an option's value applies when you're using the option in the specification for the pane; delimit the option name with "##". For example, you might define a generalized Web Page pane that lets you specify a URL and then displays that page. To do so, you'd add an option called url to the pane. Then, on the Data page, in the URL textbox, specify:

```
##url##
```

You can see an example of this approach in the Community pane. Look at the Data page for the Wiki section of the page. The call to the Wiki's web service passes the parameter ##daysold##.

HTML and XML Panes

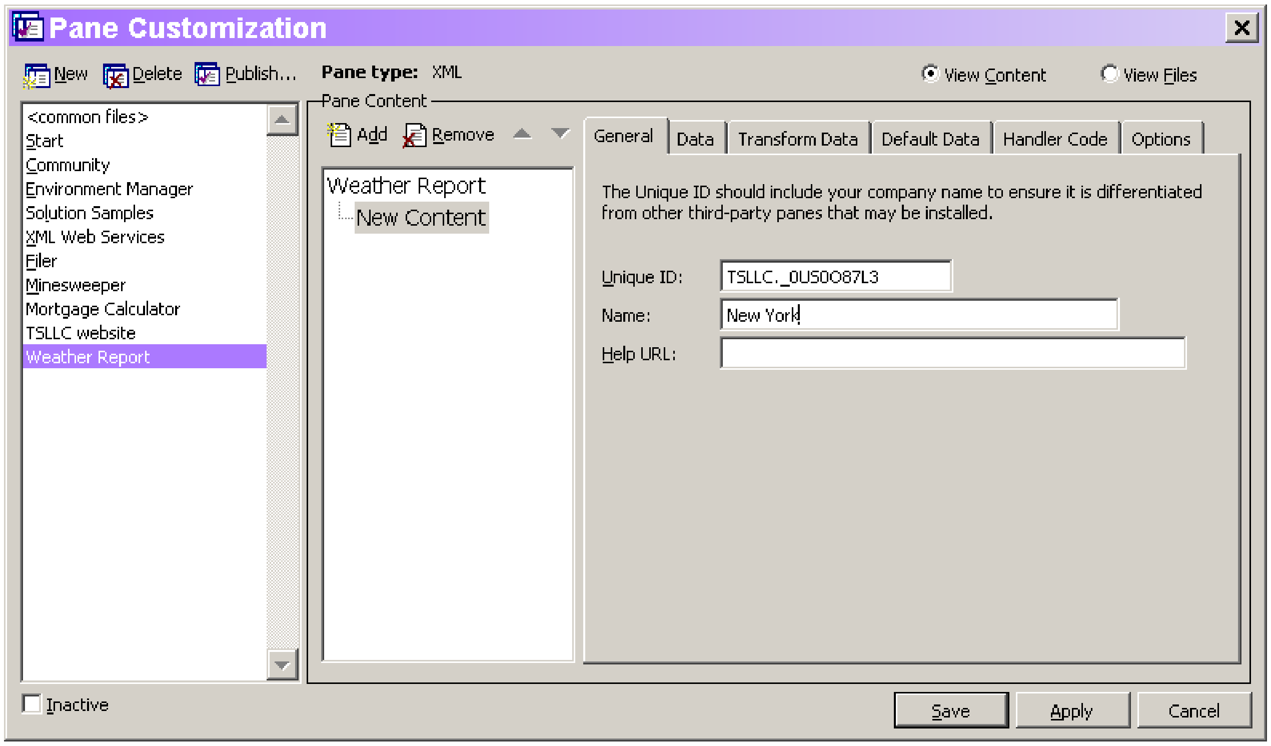
The last two types of panes, HTML and XML, have a lot in common. Both display their content in an Internet Explorer ActiveX control. The principal difference between the two is that XML panes must include an XSL style sheet to convert the content to HTML.

HTML and XML panes generally have most of their content in sections beneath the main pane. Web Page and VFP Controls panes don't support sections. You can tell whether a pane has sections by looking at the Task Pane Options window. Each section appears as a checkbox on the General options page for the pane, so you can control each separately. For example, the Solution Samples pane contains two sections: Solution Samples and Add-in Links.

As an example, we'll build a pane that displays the weather for a number of different cities. The forecast information will be retrieved using a web service whose definition is found at http://www.ejse.com/WeatherService/Service.ASMX?WSDL. (You need to register this web service before creating or using the pane. See

RegisteringWebServices.txt in the session materials for instructions.) The Weather Report pane is included in the session materials for you to install as described earlier in these notes.

Once you've created the pane, you add sections by clicking the Add button in the Pane Content section of the Pane Customization window. The new section shows up as a leaf node under the pane; you can change its ID and name on the General page, as in Figure 37. In the Weather Report pane, we'll use a section for each city we're interested in.



Pane type: XML

<common files>
Start
Community
Environment Manager
Solution Samples
XML Web Services
Filer
Minesweeper
Mortgage Calculator
TSLLC website
Weather Report

Weather Report
New Content

The Unique ID should include your company name to ensure it is differentiated from other third-party panes that may be installed.

Unique ID: TSLLC._0US0O87L3
Name: New York
Help URL:

Figure 37: Adding subsections—The Add button adds a section to the pane you're editing. Each section has its own unique ID. Display of sections can be controlled individually through the Task Pane Options window.

For an XML or HTML pane, the Data page specifies the raw source of the pane. It may be HTML or XML. When a pane has sections, the Data page for the pane itself ("Weather Report" in Figure 37) contains information for the pane as a whole. Include the string "<!-- CONTENT -->" to indicate where the content from the sections should be placed. The data for the sections is concatenated and substituted for the Content comment.

The data for the pane as a whole and for each section can come from a variety of sources, shown in Table 1. The Help topic "Data Tab, Pane Customization Dialog Box" has more information on these choices.

Table 1. Sources for XML and HTML data and transformations—The data for a pane or its sections can be specified in a variety of ways. The same choices apply for specifying transformation of data.

Source Type	**Description**
Static Text	The content or transformation is specified as XML, HTML (or, for transformations, XSL) in an edit box on the page. A Modify button appears to let you open an editing window.
URL	A URL to the content or transformation is specified in a text box on the page.
Script	The content or transformation is returned from VFP code. The VFP code is specified in an edit box on the page. A Modify button appears to let you open an editing window.
File	The content or transformation is contained in a file, which you specify in a text box on the page.
Web Service	The content or transformation is supplied by calling a web service. You specify the details for the call on the page.

For the Weather Report pane itself, the default data for an XML pane works with just a small addition. Here's the Static Text specified:

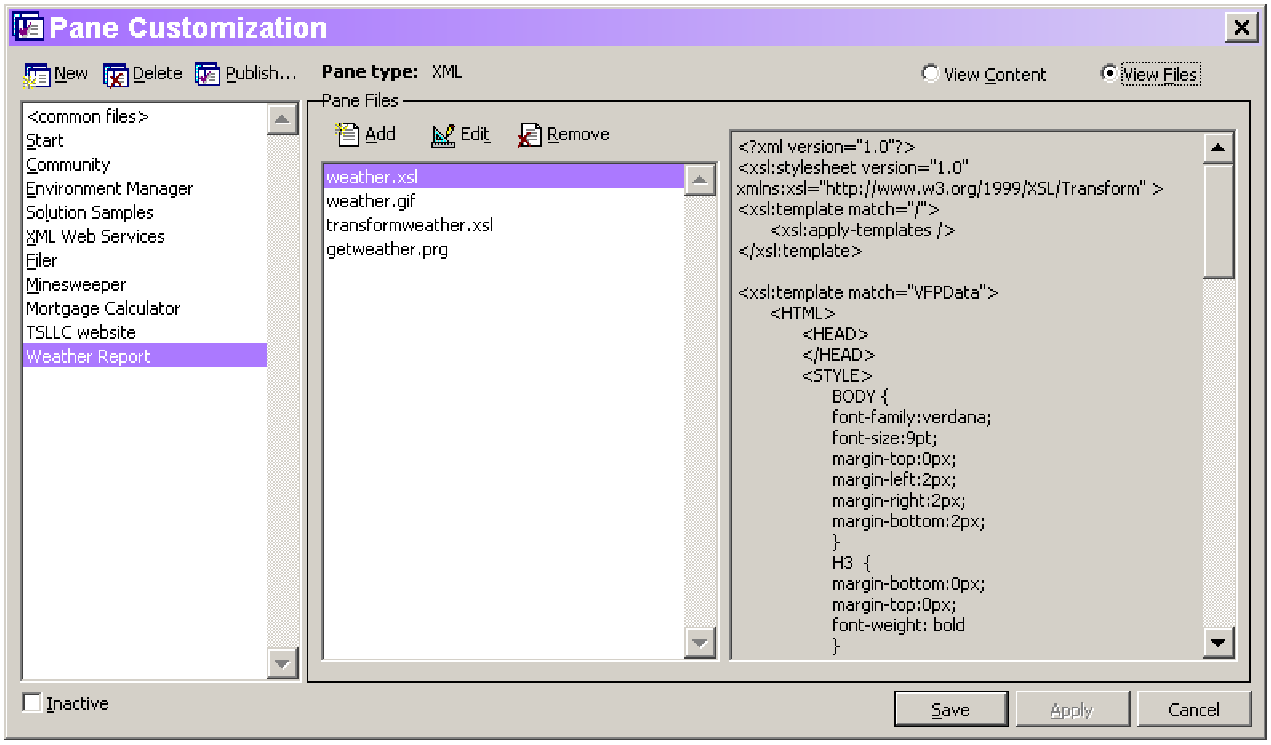
```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
<?xml:stylesheet type="text/xsl" href="weather.xsl"?>
<VFPData>
<!-- CONTENT -->
</VFPData>
```

The key change is the addition of a stylesheet reference. The specified stylesheet (weather.xsl) is used to convert the XML weather data to HTML for display. We'll take a look at it a little later on.

The sections each need to call the web service's GetWeather method, passing an appropriate zip code. Although many web service methods return an XML string as the result, the GetWeather method has a more complex result, which is returned as an object. This requires additional processing, so using a data source of Web Service isn't possible. Instead, VFP code calls the web service and processes the result. Since each section needs the same code, the code is parameterized. It's worth noting that most of this code was generated by dragging the registered web service from the Toolbox into a code window:

```
LPARAMETERS oContent, nZipCode
LOCAL loService AS "XML Web Service"
* LOCAL loService AS "MSSOAP.SoapClient30"
* Do not remove or alter following line. It is used to support IntelliSense
* for your XML Web service.
*__VFPWSDef__: loService = http://www.ejse.com/WeatherService/Service.ASMX?WSDL , ;
 Service , ServiceSoap
LOCAL loException, lcErrorMsg, loWSHandler, cXML

TRY
   loWSHandler = NEWOBJECT("WSHandler", ;
      IIF(VERSION(2)=0,"",HOME()+"FFC\")+"_ws3client.vcx")
   loService = loWSHandler.SetupClient( ;
     "http://www.ejse.com/WeatherService/Service.ASMX?WSDL", "Service", "ServiceSoap")
   * Call your XML Web service here.  ex: leResult = loService.SomeMethod()
   oXML = loService.GetWeatherInfo(nZipCode)
   IF oXML.Length>0 AND NOT ISNULL(oXML.Item(0).ParentNode)
      cXML = oXML.item(0).ParentNode.xml
   ENDIF

   * Remove XMLNS info
   cRemove = STREXTRACT(cXML,"<GetWeatherInfoResult",'>')
   cXML = STRTRAN(cXML,cRemove,"")

   * Remove degree symbol
   cXML = STRTRAN(cXML,CHR(176)," ")
CATCH TO loException
   lcErrorMsg="Error: "+TRANSFORM(loException.Errorno)+" - "+loException.Message
   DO CASE
   CASE VARTYPE(loService)#"O"
      * Handle SOAP error connecting to web service
   CASE !EMPTY(loService.FaultCode)
      * Handle SOAP error calling method
      lcErrorMsg=lcErrorMsg+CHR(13)+loService.Detail
   OTHERWISE
      * Handle other error
   ENDCASE
   * Use for debugging purposes
   MESSAGEBOX(lcErrorMsg)
FINALLY
ENDTRY

RETURN cXML
```

If only one section needed this code, we could specify a Data Source type of Script and paste the code into the edit box, substituting the appropriate zip code. Since every section of the pane uses it, it makes more sense to store the code as a PRG and call it from each pane. To add the PRG to the pane, choose the View Files option button in the Pane Customization window, which switches to the view shown in Figure 38. Use the Add button to add the PRG file. In Figure 38, the program has already been added; it's called GetWeather.PRG.



```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match="/">
    <xsl:apply-templates />
</xsl:template>

<xsl:template match="VFPData">
    <HTML>
        <HEAD>
        </HEAD>
        <STYLE>
            BODY {
            font-family:verdana;
            font-size:9pt;
            margin-top:0px;
            margin-left:2px;
            margin-right:2px;
            margin-bottom:2px;
            }
            H3  {
            margin-bottom:0px;
            margin-top:0px;
            font-weight: bold
            }
```

Figure 38: Pane Files—Choosing the View Files option button lets you see the files associated with (and stored with) your pane.

Each section of the Weather Report pane needs to make a call to the GetWeather program. To do this, specify the Data source as Script, and then use code like this (for New York) as the script:

```
LPARAMETERS oContent

LOCAL cOldPath, cXML

cOldPath = SET("PATH")
SET PATH TO (oContent.CacheDir)

cXML = GetWeather(oContent, 10001)

SET PATH TO &cOldPath

RETURN cXML
```

Note the oContent parameter. Scripts used on the Data page must be prepared to receive a single parameter, an object reference to the pane's content. In this case, we pass that parameter along to GetWeather, which uses it in case of a problem with the web service. The oContent parameter is also used to set the path, so the GetWeather program can be found.

In addition to any manipulation you perform in the code that creates the XML data, you have another opportunity to process the XML, using the Transform Data page. You can specify no transformation, an XSL transformation, or a VFP script. You have the same choices of source for either XSL or a VFP script as for the data itself, those listed in Table 1.

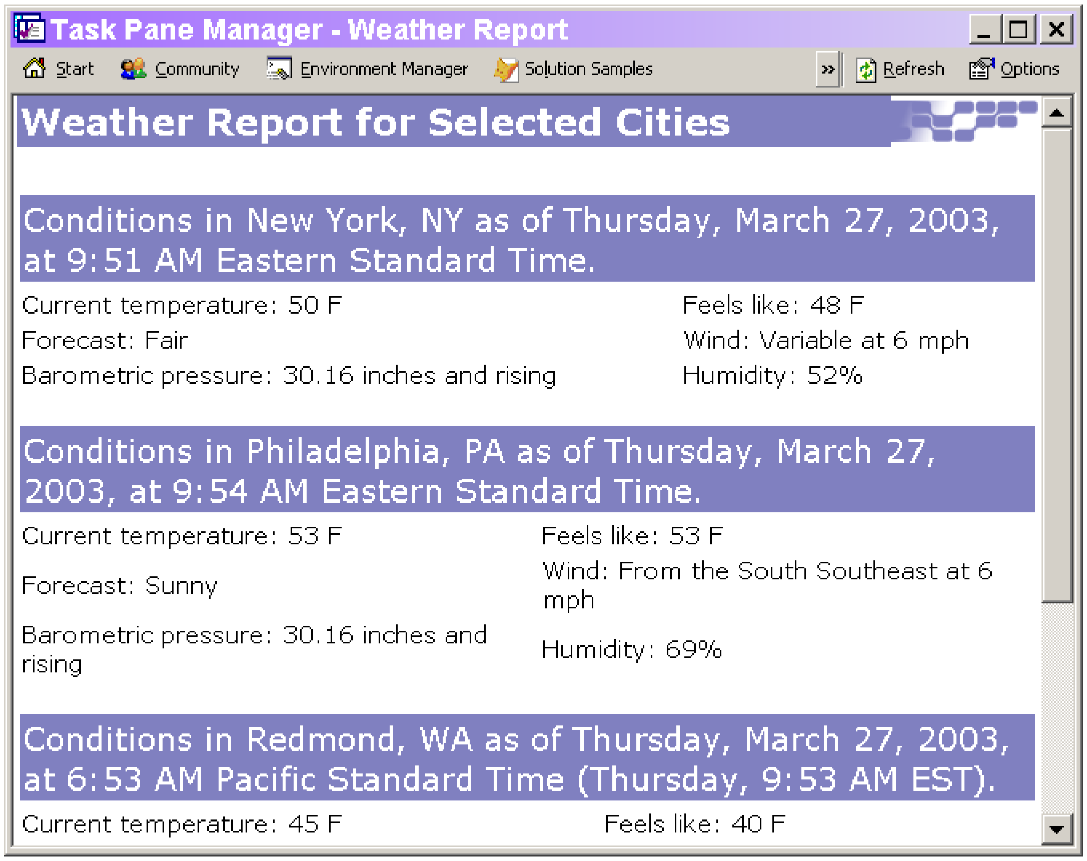
For the Weather Report pane, the XML returned from the GetWeatherInfo method is fine as it is, so the Transform Data type is set to None.

The pane itself (Weather Report) also has no transformation specified. However, we do need to convert the complete XML result to HTML. That's the role of the Weather.XSL stylesheet specified in the Data section of the pane:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:output method="html" />
<xsl:template match="/">
   <xsl:apply-templates />
</xsl:template>

<xsl:template match="VFPData">
   <HTML>
   <HEAD>
   </HEAD>
      <STYLE>
         BODY {
         font-family:verdana;
         font-size:9pt;
         margin-top:0px;
         margin-left:2px;
         margin-right:2px;
         margin-bottom:2px;
         }
         H3  {
         margin-bottom:0px;
         margin-top:0px;
         font-weight: bold
         }
         TD   {font-size:9pt}
         a:link {  color: #0033CC;text-decoration: none}
         a:visited {   text-decoration: none}
         a:hover {  color: #CC0000;text-decoration: underline}
         A {  text-decoration: underline; font-family: Verdana, Arial, Helvetica, sans-
serif; color: #0066FF
         }
         TD.TableTitle
         {
         padding:2px;
         background-color:#8080C0;
         color:#FFFFFF;
         }
         A.toggle:link {  color: #000000;text-decoration: none}
         A.toggle:visited {   text-decoration: none}
         A.toggle:hover {  color: #000000;text-decoration: none}
         A.toggle {  text-decoration: none; color: #000000}


      </STYLE>

      <BODY leftmargin="0" topmargin="0">
         <TABLE width="100%" cellspacing="0" cellpadding="0">
            <TR>
               <TD class="TableTitle" width="100%" nowrap="nowrap"><h3>Weather Report for
Selected Cities</h3></TD>
               <TD  align="left" valign="center"><img src="weather.gif" alt="" /></TD>
            </TR>
            <TR>
               <TD colspan="2" height="10"></TD>
            </TR>
         </TABLE>
         <br/>
         <xsl:apply-templates select="GetWeatherInfoResult"/>
      </BODY>

   </HTML>
```

```
</xsl:template>

<xsl:template match="GetWeatherInfoResult">
   <table width="100%">
   <tr>
   <td class="TableTitle">
   <font size="3">Conditions in <xsl:value-of select="Location"/>
    as of <xsl:value-of select="LastUpdated"/></font>
   </td></tr></table>
   <table width="100%">
   <tr><td>Current temperature: <xsl:value-of select="Temprature"/></td>
   <td>Feels like: <xsl:value-of select="FeelsLike"/></td></tr>
   <tr><td>Forecast: <xsl:value-of select="Forecast"/></td>
   <td>Wind: <xsl:value-of select="Wind"/></td></tr>
   <tr><td>Barometric pressure: <xsl:value-of select="Pressure"/></td>
   <td>Humidity: <xsl:value-of select="Humidity"/></td></tr>
   </table>
   <br/>
</xsl:template>

</xsl:stylesheet>
```

These settings give you everything you need for the Weather Report pane. In the version in the session materials, I've set up four cities (New York, Philadelphia, Redmond, and San Diego), but you can set up whatever list interests you. Figure 39 shows the Weather Report pane. (When I first built this pane and took the screenshot, the web service results includes the date and time of the weather observation. Although the XML returned still includes the appropriate tag, it no longer seems to be populated.)



Weather Report for Selected Cities

Conditions in New York, NY as of Thursday, March 27, 2003, at 9:51 AM Eastern Standard Time.

Current temperature: 50 F	Feels like: 48 F
Forecast: Fair	Wind: Variable at 6 mph
Barometric pressure: 30.16 inches and rising	Humidity: 52%

Conditions in Philadelphia, PA as of Thursday, March 27, 2003, at 9:54 AM Eastern Standard Time.

Current temperature: 53 F	Feels like: 53 F
Forecast: Sunny	Wind: From the South Southeast at 6 mph
Barometric pressure: 30.16 inches and rising	Humidity: 69%

Conditions in Redmond, WA as of Thursday, March 27, 2003, at 6:53 AM Pacific Standard Time (Thursday, 9:53 AM EST).

Current temperature: 45 F	Feels like: 40 F

Figure 39: A custom XML pane—This pane shows the weather for selected cities, using a web service that provides the information.

This example doesn't demonstrate all the features an XML or HTML pane can have. The Default Data page of the Pane Customization window lets you specify initial contents for a web-based pane, in case no web access is available the first time the pane is displayed. Once a pane has been used once, the contents are cached and those cached contents displayed if no web access is available.

Far more interesting is the Handler Code page. Not only can pane content be dynamic, but it can run code, navigate to other panes, open a browser, and pretty much anything else you can write code for. By convention, any URL in a pane that begins with the string "vfps:" (presumably, for "Visual FoxPro script") executes the handler code. The rest of the URL is parsed to indicate what action to take and to provide parameters for that action.

The Task Pane engine provides a number of script actions; they're shown in Table 2. For example, clicking on a hyperlink that has this URL opens a browser to the Tomorrow's Solutions web site:

```
vfps:linkto?url=http://www.tomorrowssolutionsllc.com
```

Table 2. Predefined script actions—You can use these actions in your pane without writing any code.

Action	**Purpose**	**Parameters received**
gotopane	Switch to the specified pane	uniqueid=cPaneUniqueID

help	Display the specified help topic	id=cTopicID
		
		or
		
		topic=cTopicName

linkto	Open a browser displaying the specified web page	url=cUrl
message	Display a message box	msg=cMessage
options	Display the Task Pane Options dialog with a specified pane chosen	uniqueid=cPaneUniqueID
refresh	Reload the current pane	None

In addition, you can write custom code and define your own actions. Handler code is VFP code. It receives four parameters, shown in Table 3. When you click the Modify button on the Handler Code page to begin writing handler code, the window that opens contains the necessary parameter declarations, along with explanatory comments.

Table 3. Parameters to handler code—Code called by a "vfps:" link receives these parameters.

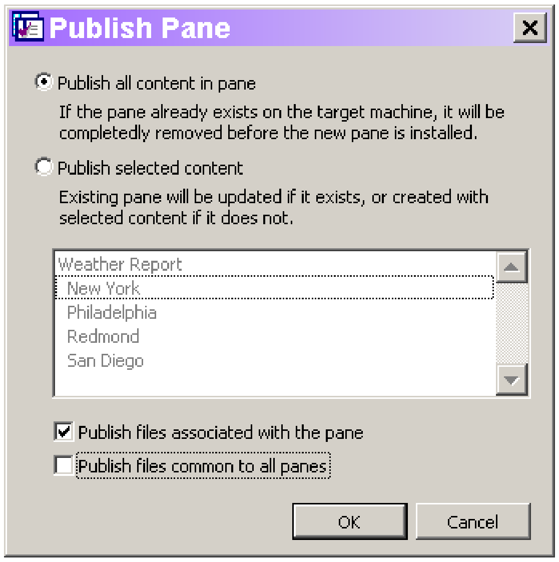
Parameter	**Meaning**

Parameter	Meaning
cAction	The action specified in the link, such as "gotopane."
oParameters	A collection listing the parameters passed in the link. The collection has a GetParam method to return the value of a specified parameter.
oBrowser	An object reference to the Browser ActiveX object in which the pane is displayed.
oContent	An object reference to the content of the pane.

A number of the panes provided with VFP 8 use handler code, using both the built-in actions and custom actions. Exploring those panes in the Pane Customization window is a good way to get a feeling for what's possible.

Publishing Panes

While you can create panes just for your own use, it's easy to distribute your panes to others. The tool allows you to create an XML file that others can install in their own Task Pane Manager. To publish a pane, highlight the appropriate pane in the Pane Customization window and click Publish. The Publish Pane dialog (Figure 40) lets you indicate exactly what portion of the pane and its associated files should be included in the XML file.


Publish Pane

[✓ Publish all content in pane]

If the pane already exists on the target machine, it will be completedly removed before the new pane is installed.

Publish selected content

Existing pane will be updated if it exists, or created with selected content if it does not.

Weather Report
New York
Philadelphia
Redmond
San Diego

[✓] Publish files associated with the pane

[ ] Publish files common to all panes

Figure 40: Publishing task panes—The Publish Pane dialog lets you specify which portions of the pane and which associated files are included in the XML file.

You have two choices regarding pane content. You can publish the entire pane, or only selected sections. For files, you have several options. Check Publish files associated with the pane to ensure that all the files used by the pane (such as TransformWeather.XSL in the Weather Report example) are included. Check Publish Files common to all panes to distribute files used by all panes, such as those involved in error reporting.

When you choose OK, you're prompted to specify a file name for the XML file. The default is the name of the pane with spaces replaced by underscores. For example, for the Weather Forecast pane, the default file name is weather_report.XML.

Under the Hood

The strategy used for task pane data is a little unusual, so it's worth taking a brief look at it. By default, task pane data is stored in a directory tree beginning in the TaskPane subdirectory of the user application data directory (specified by HOME(7)). You can change the data storage location in the Task Pane Options window. The discussion here assumes you're using the default location.

The TaskPane directory contains two tables and a subdirectory called PaneCache. The TaskPane table contains one record for each pane. The PaneContent table contains one record for each section of each pane, using the pane's unique id to link the records. Both tables store a fair amount of the pane's data in memo fields. For example, the TaskPane table has ClassLib and ClassName memos to store the class information for VFP Controls panes. Similarly, the Data memo of PaneContent stores the actual data for a pane or section, as specified on the Data page of the Pane Customization window. The PaneContent table also has records for additional files related to a pane, such as the associated icon. The file contents are stored in the Filedata memo field.

The PaneCache directory is where things get interesting. It has a subdirectory for each pane, named using the pane's unique ID. The files that belong to that pane (such as GetWeather.PRG and Weather.XSL in the Weather Report pane) are additionally stored in that directory. It's also used as a working directory for code executed by the pane. The most important consequence of this structure is that once you've added a file to a pane, modifying the original file doesn't affect the pane and modifying the code in the pane doesn't affect your original file on disk.

The Bottom Line

The Task Pane Manager is an incredibly powerful portal that lets you combine a wide variety of functionality into one container. Look around for task panes that provide handy tools. Consider defining your own task panes for things you need all the time, or to share functionality with coworkers and others. If you come up with a really useful pane, consider posting it for the VFP community at http://taskpane.leafe.com.

Some material in these notes is drawn from two articles that originally appeared in FoxPro Advisor. “Introducing the Task Pane Manager” appeared in the February, 2003 issue. “Customize the Task Pane Manager” was in the April, 2003 issue.