

Lessons from Developing San Diego's Pipe Cleaning Scheduler Extension

Dr. J.G. Prins, HDR Engineering, Inc., Folsom CA

Abstract

San Diego's Pipe Cleaning Scheduler was developed as an ArcGIS extension using an ActiveX-DLL. The extension adds three content views to ArcMap, which provide all the functionality to create, print, manage, and track the status of sewer cleaning service requests. Data are managed in an Access database, shape files, a personal geodatabase, or a remote geodatabase using ArcObject controls or ADODB. The printing tool generates a set of maps in three formats on the fly. This paper focuses on the design of the COM objects that support the extension and the lessons learned in the development of this extension.

Application Overview

Design Objectives

The San Diego Pipe Cleaning Scheduler (PSTools) extension for ArcGIS was designed as a tool for the staff of the City of San Diego's Metropolitan Wastewater Department (MWWD) to schedule service requests for cleaning the City's 2,900 miles of sewers. The objectives and scope of this sewer cleaning program as well as how PSTools and other tools were used in achieving the program objectives are discussed in a paper entitled "*Customizing GIS Applications to Optimize Water/Wastewater System Maintenance*" ([Evans, 2003](#)).

The primary design objective for PSTools was to create an ArcGIS extension that utilizes the spatial data processing capabilities of ArcGIS to create, print, manage, and track the status of sewer pipe cleaning service requests. The PSTools service request generation process is built on the following two-step data processing approach to select the sewers that require cleaning and to generate service requests from these selected sewer lines.

Building a Selection Set

The first step in the process (as illustrated in Figure 1) is to build a selection set. In this step, a table in the PSDatabase (i.e., an Access database that stores all cleaning scheduling and service request tracking information), which contains the sewer cleaning scheduling data, is queried to identify all sewers that require cleaning. The query selects all records that comply with the specified cleaning method, crew type, and scheduled time window. The query results are then used to find the sewer lines and the associated manhole in the City GIS sewer coverage. The schedule and facility access information of the selected line and manhole are merged and exported to a selection set dataset that consist of a set of line and node shape files

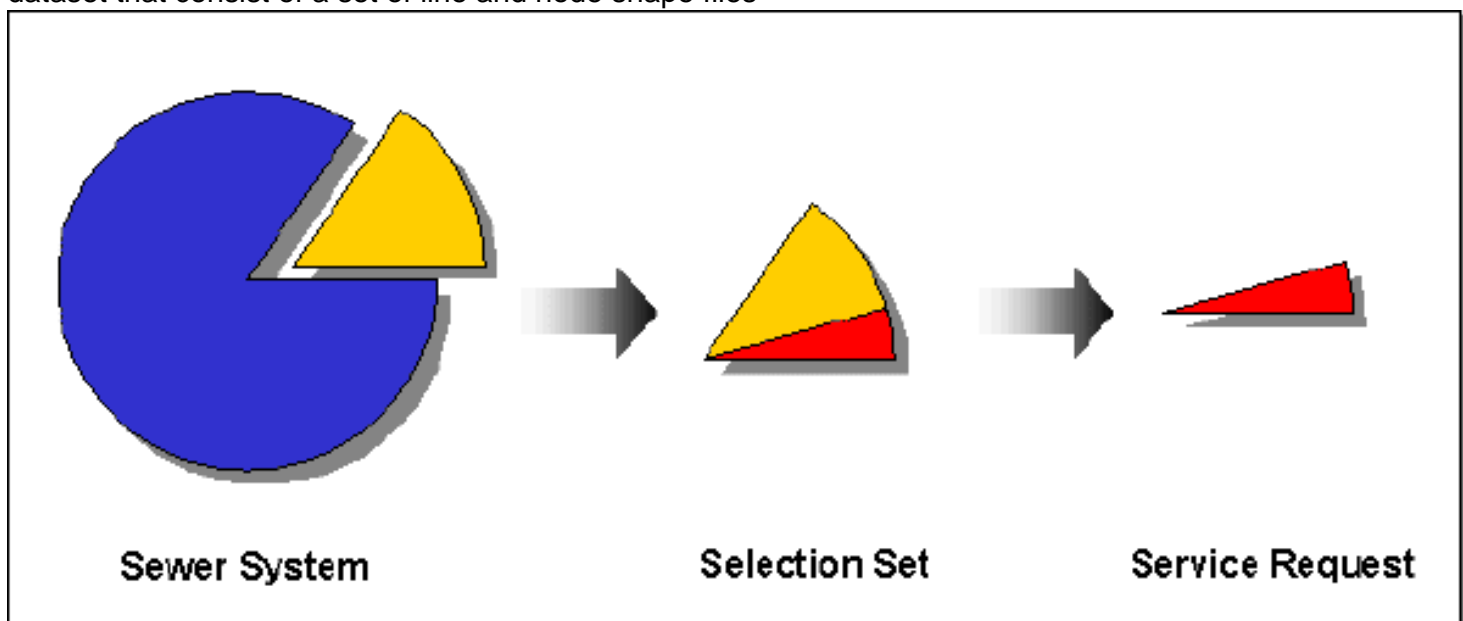


Figure 1- Service Request Generation Approach

Generating a Service Request

The second step is to use the selection set dataset and the PSTools data management tools to graphically select a sub-set of lines and to assign these lines to the service request. When a sufficient length of lines has been added to the service request, the lines and the associated manholes are exported to a set of service request shape files (i.e. a line and node shape file). The attributes of the service request dataset are similar to that of the selection set. Once exported, the service request is printed as a set of maps that includes the maintenance inspection forms to be filled out by the cleaning crews. The printed service requests are bound in map book format and issued to maintenance crews.

This paper focuses on some of PSTools features and the lessons learned in the development of these features. It also includes some tip and tricks for programmers who want to develop similar applications.

Application Components

The PSTools application is not a single software program, rather as illustrated in Figure 2, it is a comprehensive set of data management tools with the associated data management system and data sources. PSTools uses a unique data management approach that was designed to create and manage a service request through its life cycle. PSTools' ArcGIS Extension - the focus of this paper - is one of the data management tools. It includes a number of ActiveX libraries that are developed in Visual Basic or Delphi. Some of the key features of PSTools' ArcGIS Extension are reviewed below.

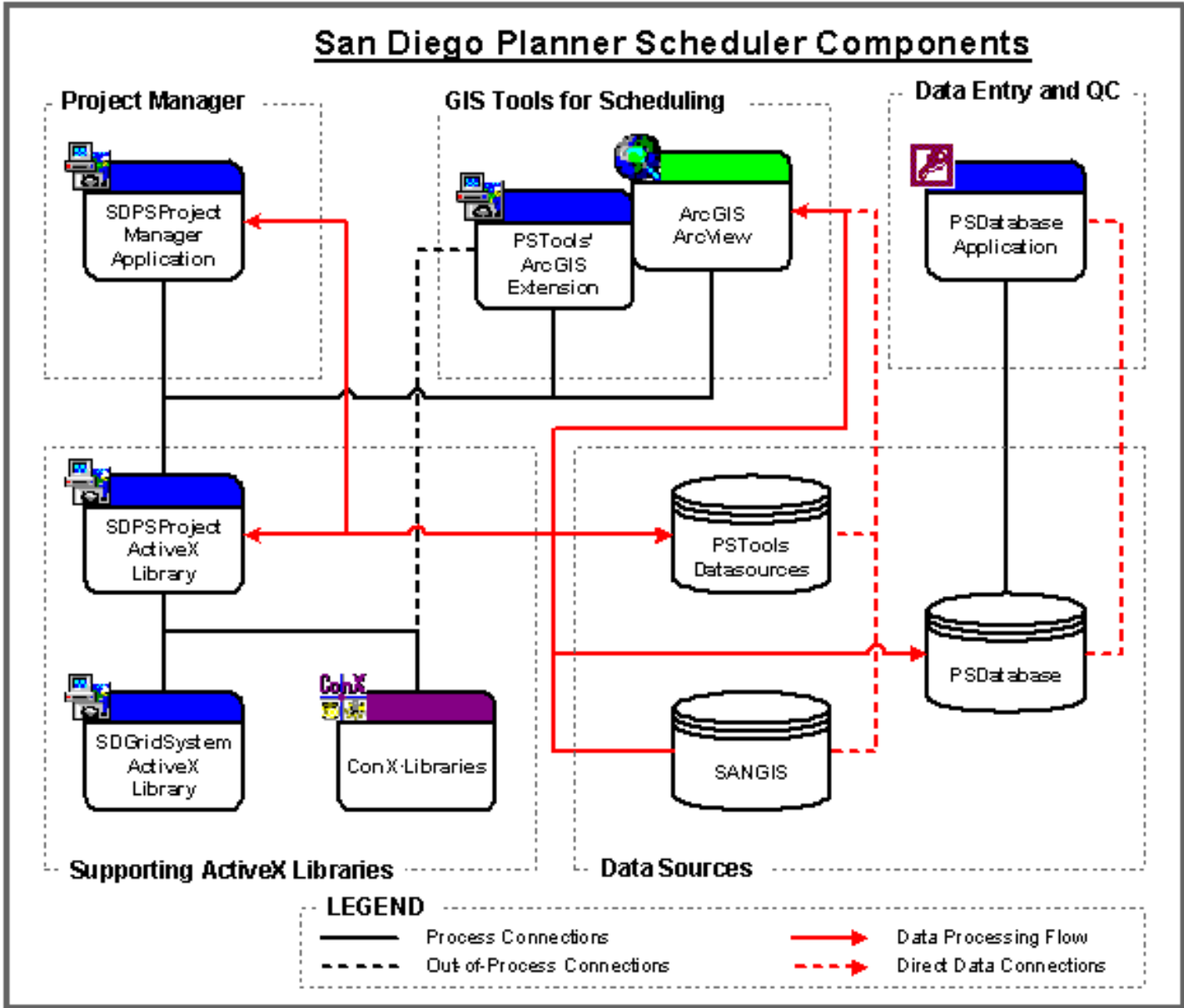


Figure 2 - San Diego's PSTools Application Components Key Application Features

The majority of the functions and options added by the PSTools' ArcGIS Extension is embedded in three user-defined Contents Views: the Selection Set Manager, the Service Request Manager, and the Service Request Event Manager. These Contents Views are added to ArcMap's Table of Contents (TOC) window when the extension is activated.

As shown in the Figure 3, each Contents View has its own embedded toolbar. All commands and settings associated with the Contents View are added to this toolbar instead of providing a separate command bar or menu. This makes it possible to use the Contents Views in a standalone application where each Contents View is displayed as a dialog. See the discussion on [Using Contents Views to Customize ArcGIS](#) for the lessons learned in developing these Contents Views and for highlights on some of the features included in these data management tools.

Each Contents View was designed to provide a specific set of functions that can be used to fulfill a specific task in the service request generation process. Some of the key features of these Contents Views are reviewed below:

Selection Set Manager: The Selection Set Manager allows the user to define and extract a selection set, build a service request, and export the service request. The Selection Set Manager has several unique features. One of these features, the selection tools used in adding or removing lines from the service request, takes control of the ArcGIS feature selection settings and communication between PSTools and ArcGIS through layer selection events. See discussion below for more details.

Service Request Manager: Once the service request is exported from the Selection Set Manager, it can be printed or deleted in the Service Request Manager. Each printed service request consists of several tabloid-size maps in three different formats. The discussion on [Automated Maps](#) below reviews how this printing process streamlines the service request printing and the data management process.

Service Request Event Manager: This Contents View does not interface with the map display, but with the PSDatabase—an Access database that contains the scheduling data, service request tracking information, inspection data, etc. It allows the user to track the status of each service request through a 14-step life cycle. This tool can be used to register an event (e.g., when the crews return a service request after the cleaning has been completed). Some events (e.g., initializing, exporting, and printing of service requests) are automatically registered.

The service request generation process requires access to several dataset types, including geodatabases, shape files, and an Access database. Depending on the data processing step, these datasets can be accessed using either the ArcObjects or the ADODB database connections and table (or record set) objects. PSTools includes a set of ActiveX classes that define the dataset workspaces, connections, table structures, field mapping, etc. to allow any dataset to be accessed as an ArcObjects dataset or an ADODB dataset. See discussion on [Flexibility to Access Multiple Dataset](#) for more details on this feature.

The rest of this paper provides a brief overview of some of the lessons learned in developing PSTools. It also highlights some of the techniques used to establish communication between ArcGIS and PSTools.

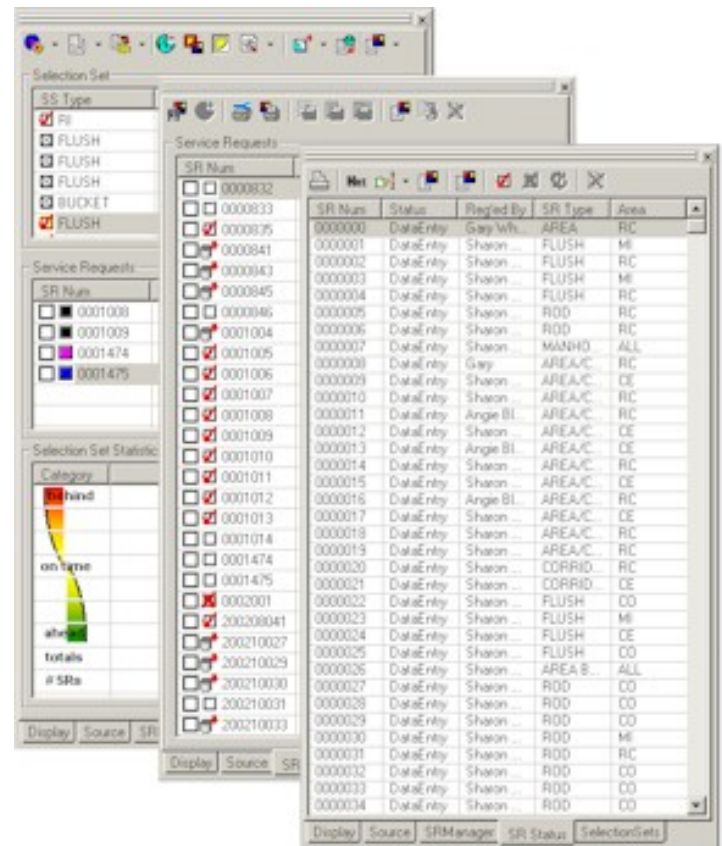


Figure 3 - Selection Set, Service Request, and Service Request Event Manager

Using Contents Views to Customize ArcGIS

Adding a user-defined Contents View to ArcGIS's Table of Contents (TOC) window is a customization option that should be explored when planning an application that will add, remove, and manipulate several datasets in the ArcGIS environment. The following are some reasons why this option should be considered:

- When a user selects the Contents View in the TOC, the view becomes the active Contents View, similar to a dialog that receives the focus when a user opens it or selects it. It will remain the active Contents View until the user selects another Contents View. When the Contents View is activated and deactivated the communication between the extension and ArcGIS should be established or discontinued. This communication is essential to control some of the ArcGIS processes while the Contents View is active.
- Unlike non-modal dialogs (the alternative to having a control that interacts with the ArcGIS desktop), the Contents View is active even when the user focuses on the map or any other control on the desktop. A non-modal dialog loses focus the moment the user selects the desktop or any other control, which makes it difficult to maintain or enforce communication between the dialog and ArcGIS.
- The Contents View shares screen space with other views in the TOC. Adding too many floating dialogs, especially one that consumes a lot of screen space, makes it difficult to access other controls on the desktop.

ArcObjects provides an easy way to add a user-defined Contents View. All that is required is to add a new class module (say `MyExtDll.IMyContentsView`) to a Visual Basic ActiveX library that implements the `EsriCore.IContentsView` interface. Through this user-defined class module your Contents View frame can be linked to the TOC and communication between the extension and ArcGIS can be established. This technique is discussed in the ArcGIS help file and examples are provided in the *ArcObjects Developer Kit*.

The following are some lessons learned in the development of the Contents Views.

Creating the Contents View Frame

In Visual Basic a Contents View can be designed by adding a new form to the project. As illustrated in Figure 4, a picture box can be added to the form to act as the frame for the Contents View (say `picContentsView`). All controls that must be visible in the Contents View should be added to the picture box (i.e., as child controls).

Since the picture box is part of a form, the form could also be designed as a dual purpose form (i.e., a form that can be display as a dialog when called from a standalone application or that can be used as a Contents View in the ArcGIS environment). All three the contents view included in PSTools are designed as dual-purpose forms. Not only does this approach provides the flexibility to use the same application outside the ArcGIS environment, but it also provides a means to test and debug a Contents View before it is deployed in the ArcGIS environment.

The following three code segments are required to configure the form as a dual-purpose form:

Standalone Property: Add a *Standalone* property to the form to control the align mode of the picture box and the form's display mode. The code to set the property value is as follows:

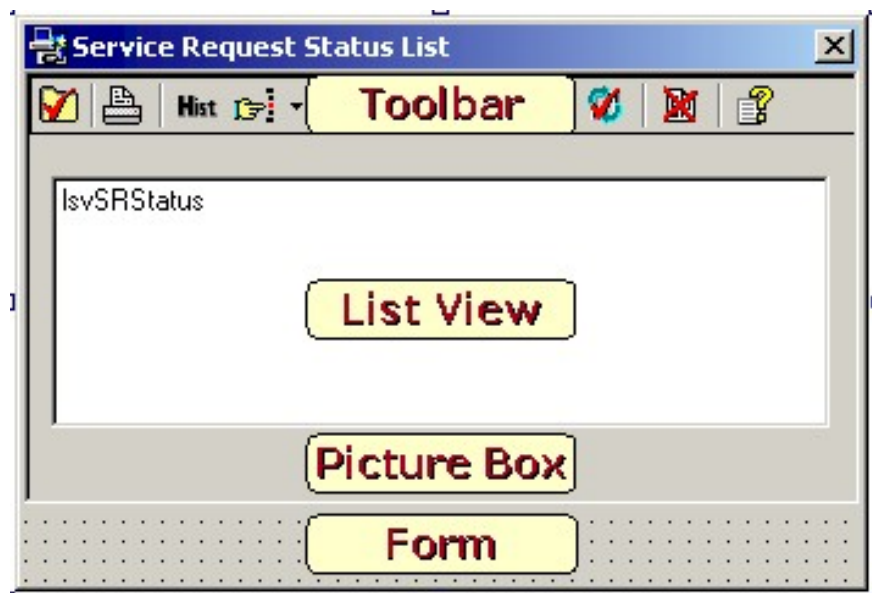


Figure 4 - A Typical Visual Basic Form Design Layout for a Contents View

```
Public Property Let StandAloneForm(Value As Boolean)
    mfStandAloneForm = Value '** Assign the Value to a private variable **
    If (Value) Then
        Me.picContentsView.Align = 1 '** Align Top **
    Else
        Me.picContentsView.Align = 0 '** Align None **
    End If
    SetControls '** Call Sub Routine to Update the Toolbar and other controls **
End Property
```

Form OnResize Event: Add the following code to the forms OnResize event to set the picture box size when the form is displayed or resized in Standalone mode. The assignment of the picture box's size will trigger its OnResize event. Note that since the picture box's *Align* property is set to *AlignToTop* (i.e. aligned to the top and width of the form), only the height property can be set.

```
Private Sub Form_Resize()
    If mfStandAloneForm Then '** Check if FORM if in Standalone Mode **
        '** Check the minimum and maximum dimensions of the form **
        If Me.Width < 5000 Then
            Me.Width = 5000 '** Minimum Width of 5000 twips **
        ElseIf Me.Width > 7000 Then
            Me.Width = 7000 '** Maximum Width of 5000 twips **
        End If

        '** Set the Height of PictureBox - this will trigger the PictureBox.OnSize Event **
        Me.picContentsView.Height = Me.ScaleHeight
    End If
End Sub
```

Picture Box OnResize Event: Add the following code to the picture box's OnResize event to control the size of the contents view when not on Standalone mode

```
Private Sub picContentsView_Resize()
    If Not mfStandAloneForm Then '** Check Size of Picture Box if NOT Standalone **
        If picContentsView.Width < 4500 Then
            picContentsView.Width = 4500 '** Minimum Width of 4500 twips **
            Exit Sub '** Exit sub to prevent multiple to lsvSRStatus_Resize **
        ElseIf picContentsView.Width > 6500 Then
            picContentsView.Width = 6500 '** Maximum Width of 6500 twips **
            Exit Sub '** Exit sub to prevent multiple to lsvSRStatus_Resize **
        End If
    End If

    lsvSRStatus_Resize '** Call Sub to Resize the List View and other Controls **
End Sub
```

To display the form as a dialog the Standalone property must be set to true. The form will then control the resizing of the picture box. When it is used as a Contents View the Standalone property is set to false and ArcGIS controls the size of picture box by calling its OnResize event directly.

Adding Command Buttons to the Contents View

There are two options to add user-defined commands as command buttons or menu options to the ArcGIS desktop. The one option is to add a new command bar that contains the user-defined commands and tools. Another option, as illustrated in the Figure 4, is to add the command buttons to a toolbar on the Contents View.

This approach was used in developing the three PSTools Contents Views. Some of the benefits of this approach are:

- The same toolbar and set of commands are used when the form is used in the Standalone mode or as a Contents View.
- The commands interact directly with the form's controls and the extension, which makes it easier to control each command's action or the button's state.
- Each Contents View becomes a self-contained control with its own set of command buttons and options and does not require any other user-defined menus or command bars on the desktop.

- A help button on the toolbar makes it possible to access a specific topic in a user-defined help file.

Controlling Map Selection from the Contents View

Set Active and Visible Layers

InPStools the extension and Contents Views are design to take full control of the display of all layers. As illustrated in the Figure 5, the service request and selection sets layers are organized under group layers. Each service request or selection set is also represented with a group layer, which contains a line and node layer. The legend symbols for the service request and selection set layers are programmatically assigned.

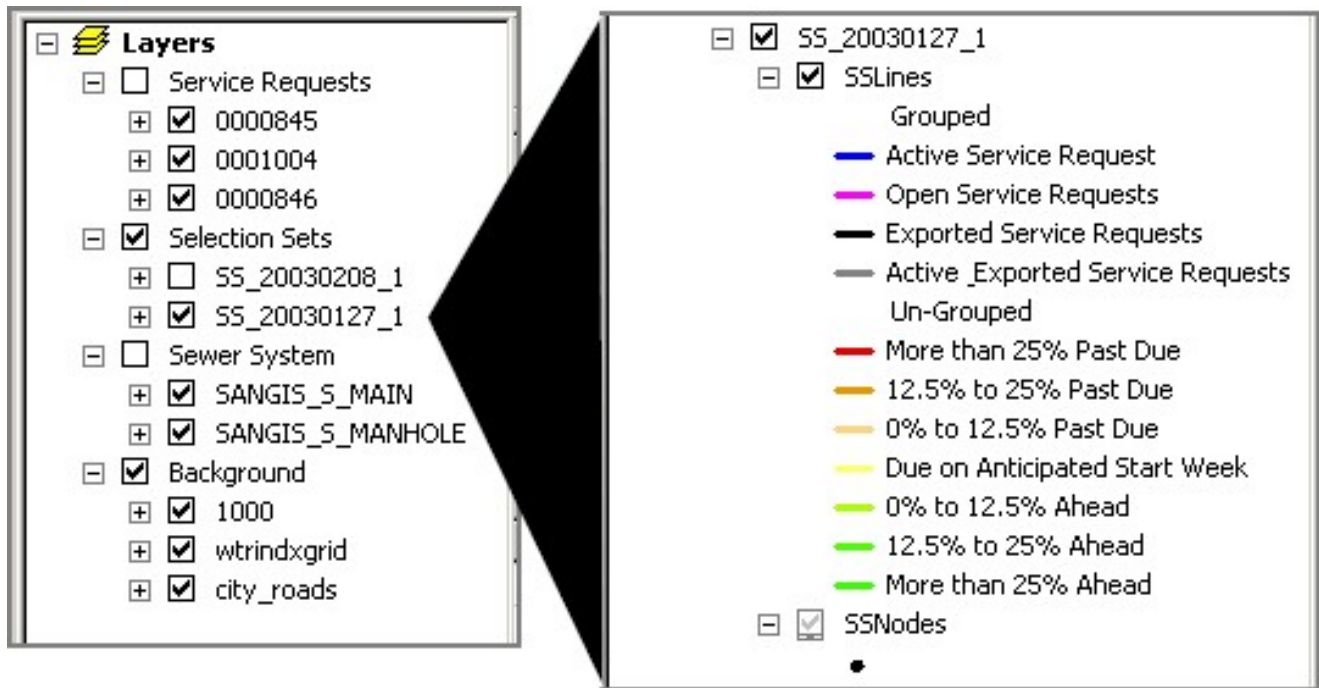


Figure 5 - Typical Layer Organization and Legend Details for the Selection Sets

Service request and selection set layers are loaded or unloaded using the Contents Views' toolbar options. The visibility of the layer is also controlled by Contents View . The user can use the Select Other Visible Layers option (see Figure 6) to select what other project layers should be displayed with the selection set or service request layers.

When a user activates either the Selection Set Manager or Service Request Manager, it not only activates the associated Contents View, it also turn layers on or off to display only the layers associated with the Contents View. All layer control settings are saved to an external project file that is loaded when the extension is activated.



Figure 6 - Other Visible Layers can be Selected in the Selection Set Layer Option Dialog

When a selection set is selected as the *Active Selection Set* in the Selection Set Manager, the display of all other selection sets are turned off and only the *Active Selection Set* is visible. Only the selected other visible layers will be displayed with the active layer. Similarly, when a service request is printed, only the selected service request and the other visible layers will be visible.

Using Selectable Layers and Layer Selection Events

One of the unique features of PSTools is that it allows users to select lines from a selection set and to assign it to a service request by using only the standard ArcGIS feature selection tool. This is accomplished by control the selectable layer and using the selectable layer's IFeatureLayerSelectionEvents interface.

When a selection set is set as the Active Selection Set in the Selection Set Manager, the selection set is automatically set as the only selectable layer. To active the selection tool, the user must select the service request to be updated and one of the selection modes as shown in the Figure 7.

When a user make a selection, the FeatureLayerSelectionChanged event of IFeatureLayerSelectionEvents interface is intercepted and the selected features are processed. First, each selected item is checked to make sure it can be selected (i.e., based on the selected selection mode) and it will be removed from the selection if not. The net result is that the user sees an updated selection with only selectable feature in the selection.

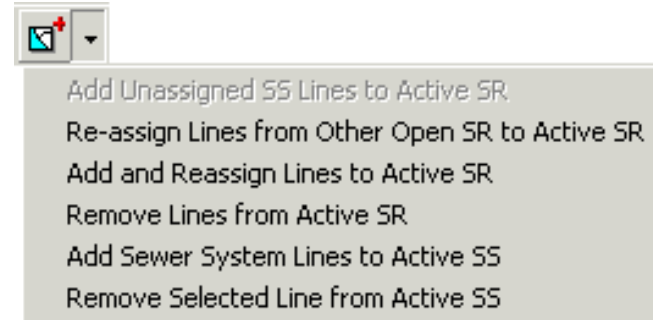


Figure 7 - Selection Set Manager's Feature Selection Modes

Once the selection is updated, the tables of the contents view are updated to display the number of lines and the total feet of line in the selection. When satisfied with the selection, the user can use the Add to Service Request toolbar button to assign the selected lines to the service requests and the selection is cleared. The selection set attributes are updated and the map display is refreshed.

Flexibility to Access Multiple Dataset Types

Workspace and Table Definition Classes Adds Flexibility

When using dataset from various dataset types (e.g., geodatabases, Access databases, shape files, etc.) in a large application the following are some of the factors that should be considered during the application design:

- Which database driver should be used?
- Should the database be accessible outside the ESRI environment?
- Could the dataset's workspace change (e.g., from a shape file to geodatabase)?
- How to coordinate the mapping of field from different dataset.
- How to convert and link long and short field names.
- How much effort will it take to update the code if a field name change.

In the development of PSTools we addressed most of the above listed factors by adding a Workspace Definition class object and a Table Definition class object. These classes support two database drivers: the ESRI database access as defined in ArcObjects and ADODB database access.

The Workspace Definition class stores all the workspace connection information required to open the workspace as an ESRI workspace or an ADODB connection. The available Workspace Definitions provide support for Access databases, shape files, local and remote geodatabase, and remote databases (e.g., Microsoft SQL).

The Table Definition class includes the information that makes connection and communication with dataset possible. This includes the table's workspace definition, the table name, and the field mapping. The Table Definition supports two interfaces - one to open the table as an ADODB dataset and the other to open it as an ESRI dataset. The latter would only be available if ArcObjects is supported on the user's machine. Table Definitions for feature datasets not only allows the user to access that dataset as an ESRI feature class, but it can be opened as a table (i.e., the datasets attribute table with no spatial support).

The field map definition includes only the fields that are used in the application and each field map identifies the field by an Alias, a Short Name and a Long Name. In the case of shape files the field map data can be maintained in a field map table that is stored with the shape file. The dataset's fields can be accessed using the alias, short name or long name definition. This enabled seamless links between data sources with short and long name definitions (e.g., geodatabases and shape files).

Only the field aliases are used in the application code to access a field. Therefore, when a field name change only the field map definition have to be updated. This minimizes the effort to maintain communication between datasets and ensures that the right field is accessed at all times.

Automated Map Making

Generating Maps in Memory

The service request is printed as a set of maps that is issued to the crews in a map book. Each set of maps used three formats:

A location map - The map book contains only one location amp, which displays all the lines included in the service request and is zoomed to the extent of service request.

A set of grid maps - The City use a 6000 x 4000 feet main grid system. The set of maps includes a grid map for each main grid with service request lines located in it. This map is zoomed to the extent of the main grid and the grid number and the sub-grid numbers are annotated on the map.

A service request maps - Each main grid is sub-divided into twenty-four 1,000 x 1,000 ft sub-grids. The set of maps includes a service request map for each sub-grid that contains service request lines. This map's map frame is resized to the size of the sub-grid. The rest of the space-to the right of the map frame-is used to add the inspection forms, which contains a record for each service request line and node shown on the map (see Figure 8). The inspection forms are generated on the fly showing only records for the lines and nodes displayed on the map.

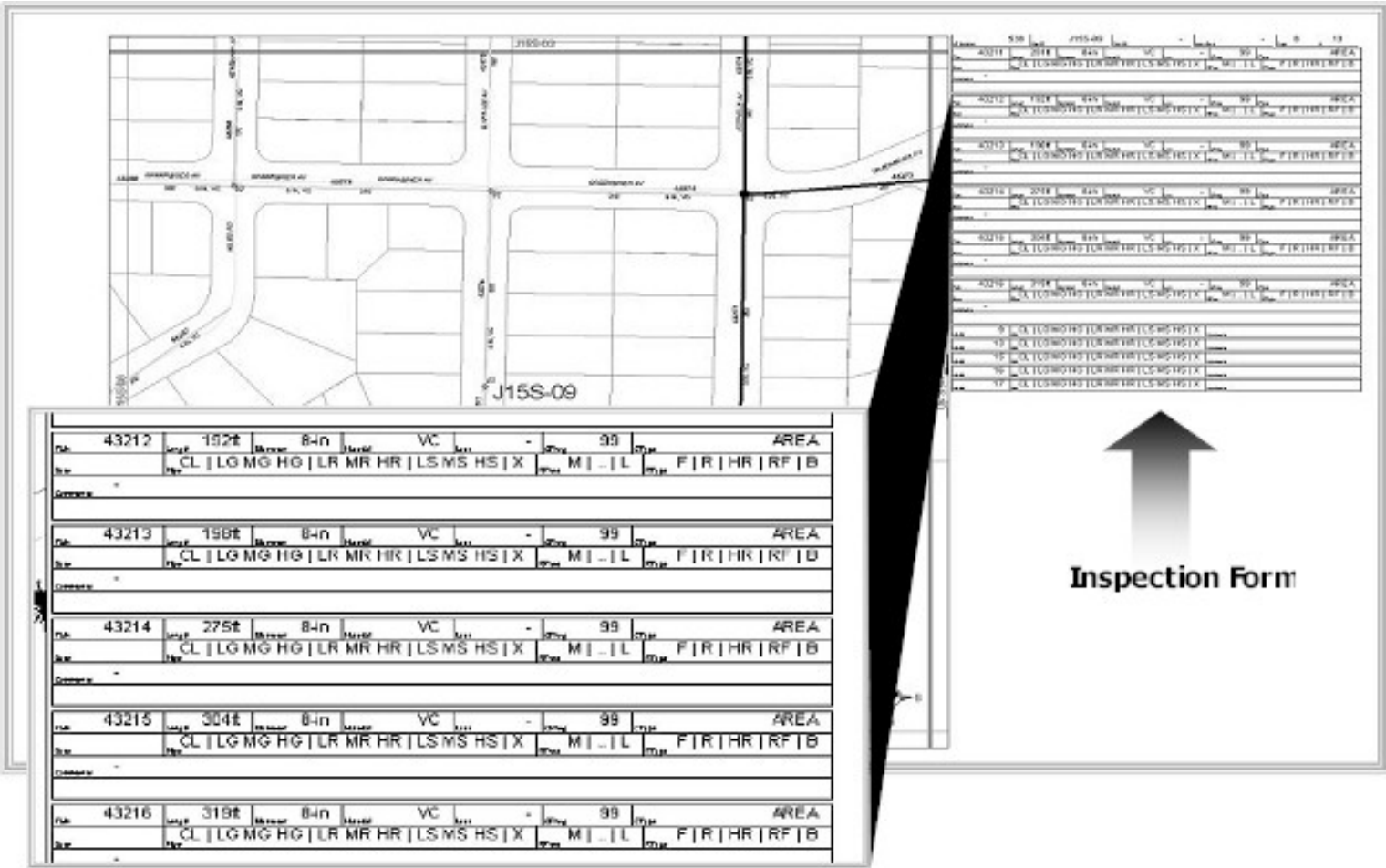


Figure 8 - Typical Service Request Map with Inspection Form

Since ArcMap supports only one map per project, an automated mapping processing utility was developed to print the service request map book. In the map generation process the service request dataset is used to select all the main and sub-grids to be printed. It then generates a collection of maps in memory in the sequence it would appear in the map book. The stored maps will be rendered and printed in the sequence it is stored.

The printing process is activated from the Service Request Manager. Two print buttons are provided: one to print a single service request and the second to print multiple service requests in batch mode. It also includes a print preview option. The preview process will generate the maps in memory and opens a dialog that allows the user to select the map to be rendered or printed.

Add Graphic Elements to Map

Four ActiveX libraries (DLL) support the automated mapping process. The graphic element objects used in the formatting of the maps (e.g., the lines and text to generate the service request form) are created in the DLLs. Since these sub routines that creates these objects runs in a different application workspace (i.e., that of the DLL) than ArcMap these graphical components would not be visible on the map if created with an in-process call to initiate the objects. However, ArcObject provides an *IObjectFactory* interface to the *esricore.Application* object that support a Create method-an out-of-process method to create ArcMap graphical components from other application workspaces. The following code shows the incorrect and correct way to generate these elements:

The incorrect in-process call that will not display the graphic element on the map:

```
Dim pElement = esricore.ITextElement

Set pElement = New esricore.TextElement
pElement.Text = "My Text"
```

The correct out-of-process call that will display the graphic element on the map:

```
Dim pFactory = esricore.IObjectFactory
Dim pElement = esricore.ITextElement

Set pFactory = MyExtensionApplication
Set pElement = pFactory.Create("esricore.TextElement")
pElement.Text = "My Text"
```

The latter example assumes the *MyExtensionApplication* variable points to the *esricore.Application* link that is passed to the extension.

Conclusions

This paper focuses on some of the unique features of the PSTools application and addresses some of the lessons learned in the development of the application. PSTools was designed to automate the service request generation process as much as possible. However, it still gives the user maximum flexibility in producing a customized end result.

PSTools includes several automated data processes that run in the background without the users knowledge or input. This approach makes the application more user friendly, minimize data processing steps, and minimize data management errors. The user input is kept to a minimum throughout the service request generation and management process. However, the main benefit of this approach is that it makes ArcGIS accessible to none GIS users.

References:

Evans, J., "Customizing GIS Applications to Optimize Water/Wastewater System Maintenance", Paper # 941, ESRI International User Conference 2003, San Diego, CA.

Dr. J.G. "Koos" Prins is a Senior Professional Associate at HDR Engineering, Inc. and the Owner of ConX-Software.