COM Automation Extension

Introduction

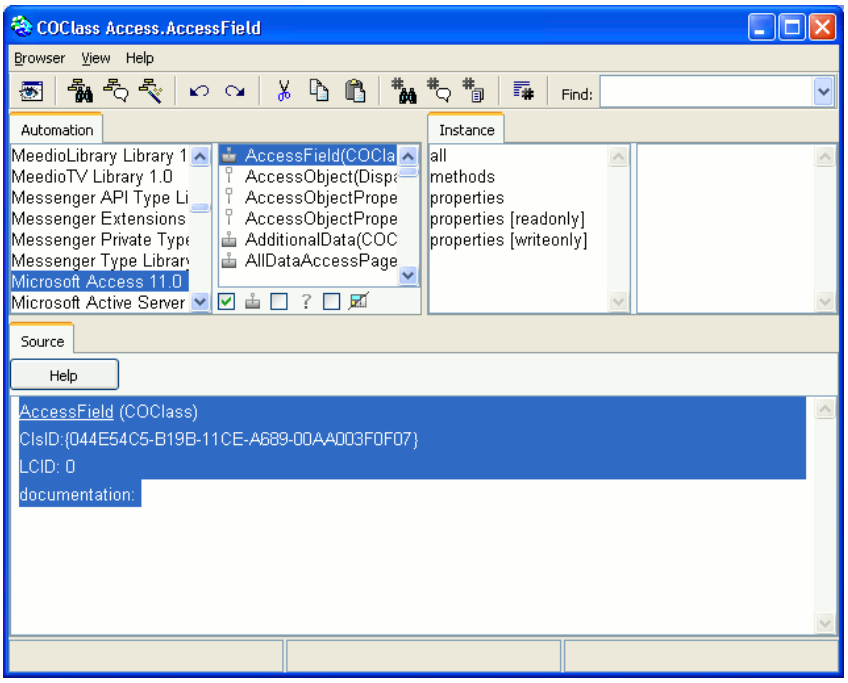
The COM Automation extension consists of two parts:

- An Automation browser which can be used to view type-libraries and the contained type descriptions.
- An extension to the ***Trippy*** Inspector system which enhances VisualWorks' abilities to view COM Automation Objects
- An extended COMDispatchDriver subclass (***AdvancedDispatchDriver***) which provides additional features like automatic release and automatic type information retrieval.

How to use

The Automation Browser

To open the Automation Class Browser open select "Tools -> COM -> Browse Automation Classes…" from the Launcher menu.



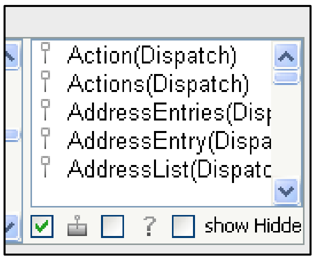
AccessField (COClass)
ClsID:{044E54C5-B19B-11CE-A689-00AA003F0F07}
LCID: 0
documentation:

The AutomationBrowser is a modified version of the RefactoringBrowser. Therefore Smalltalk Developers will already be familiar with most of the things that can be found in the Browser.

The browser consists of four upper list panes and a special source description pane in the lower half of the window:

Type Library pane – The first, leftmost list pane contains all Typelibraries registered in the system. Typelibraries contain detailed information about objects and types provided by a specific COM server.

Types pane – The second list pane contains the types described for the type library selected in the Type Library pane. Types may be



CoClasses

Interfaces

Enumerations

Aliases

Structures

Unions

Each type is easily identified visually by an icon that appears before its variable name in the list. Below the list in this pane there are check boxes that select the types listed. The first check box will show or hide Coclasses and Interfaces (including Dispatched), the second will show all other types when selected, and the last will hide or show all types that are systems types or are marked as hidden.

Protocols pane – This next list pane contains the protocols of the selection in the Types pane. These are not protocols in the Smalltalk sense but static categories for each kind of type. While Coclasses have the protocols “methods” and “properties”, Enumerations will have the protocol “constants”.

Members pane – The rightmost list pane will display the members of the selected Type depending on the selection in the Protocol pane.

Source Description pane – Selecting a member in the Members pane will display detailed information this pane in the lower half of the window. Unlike the Refactoring Browser, it will not allow editing of its contents. The Source Description pane has an option to explain selected text.

Currently supported features

Explain:
The browser supports an explain feature that will attempt to provide a short explanation for whatever is selected in the Source Description pane and will allow you to open a new Browser on the selected element, if the element is a custom type provided by a type library. This feature operates across type libraries and for hidden types. Currently the functionality is only accessible by selecting a text in code tool and selecting “Explain” from a pop-up menu or pressing <Control>-E.

Search:
A user may search for a type by using the search input field in the top-right of the browser window. As in the Refactoring Browser, if the user enters the name of the type to search for

and presses <Enter> the Browser will list Types it finds in a dialog allowing the user to select the Type it should navigate to. The search string may also contain the name of a type library, (e.g. a possible query may be “Word.Application”). Searching for members of types (e.g. methods) is currently not supported.

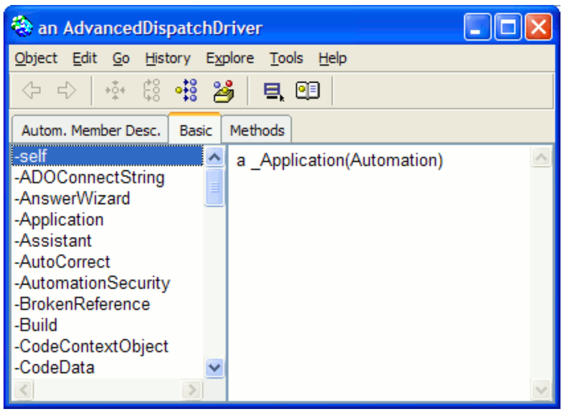
References:
Two menu items allow searching references to a specific type, which are occurrences of a type inside another type or as parameter/return value type. Type aliases will be resolved and references to the alias will also be listed. “Local References” will list all references inside the type library in which the type is defined. “Global References” will search all type libraries for references that are registered on the computer. Whereas searching for local references is quite fast, searching all global references may take some time. Therefore, during a global search a progress dialog will be displayed.

Instance creation:
The browser allows creating instances of a coclass and directly interacting with it in an inspector. To do this, select a coclass Type in the Type list pane and select “Create Instance and inspect” from its pop-up context menu. An inspector will open on the created coclass. For more information about inspectors and related extensions please refer to the **The extended inspector** section in this document.

Implementors:
It is now possible to browse implementors of a method or property. There are two versions, **local implementors** and **global implementors**. A search for local implementors conducts a search within a single type library whereas a global implementors search searches all type libraries registered on the computer.

The extended inspector

The Inspector (***Trippy***) has been enhanced to show additional information about instances of ***AdvancedDispatchDriver***, an enhanced subclass of ***COMDispatchDriver*** which is described in a separate chapter of this document. This driver may represent any Automation coclass and may be created by the Automation Browser.


an AdvancedDispatchDriver
Object Edit Go History Explore Tools Help
Autom. Member Desc. Basic Methods

```
-self
-ADOConnectString
-AnswerWizard
-Application
-Assistant
-AutoCorrect
-AutomationSecurity
-BrokenReference
-Build
-CodeContextObject
-CodeData
```

a _Application(Automation)

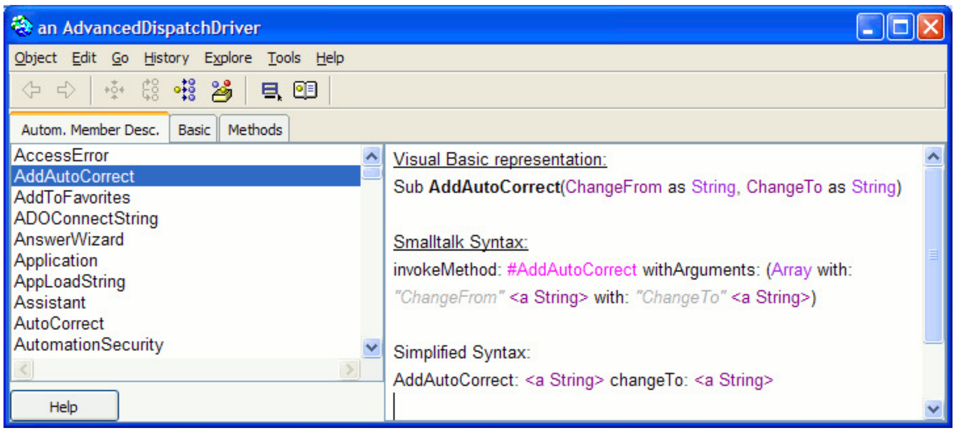
The ***Basic*** tab has been extended to support the display and update of Automation properties of the object inspected. One may set properties of the remote Server object in an inspector just by typing a new value in the inspector and accepting it.

> Most application classes have a ***Visible*** property. When such an object is created, this property usually is set to ***false***, which means that the application window is invisible. Try changing the value to ***true*** and accept it. The application window should become visible now.

Inspector: Automation Member Description tab

In addition to Automation property display support, the inspector will show different additional tabs, depending on the kind of Automation object which is being inspected.

The ***Automation Member Description*** tab will be displayed for all kind of coclasses represented by an ***AdvancedDispatchDriver***.


an AdvancedDispatchDriver
Object Edit Go History Explore Tools Help
Autom. Member Desc. Basic Methods
Help

```
AccessError
AddAutoCorrect
AddToFavorites
ADOConnectString
AnswerWizard
Application
AppLoadString
Assistant
AutoCorrect
AutomationSecurity
```

Visual Basic representation:
Sub **AddAutoCorrect**(ChangeFrom as String, ChangeTo as String)

Smalltalk Syntax:
invokeMethod: #AddAutoCorrect withArguments: (Array with: "ChangeFrom" <a String> with: "ChangeTo" <a String>)

Simplified Syntax:
AddAutoCorrect: <a String> changeTo: <a String>

For the selected member (which is a property or a function) it will display a syntax highlighted Visual Basic representation, the normal Smalltalk syntax for calling an automation member for a ***COMDispatchDriver*** and the simplified Smalltalk syntax.

Both, VB and Smalltalk representations are displayed because specific types of information are easier to acquire from only either of them. For example, the Smalltalk syntax does not say anything about the result type but provides detailed information about classes which may be used to pass arguments.

- Using simplified syntax for calling functions is only possible with ***AdvancedDispatchDrivers*** or (***COMDispatchDrivers*** support this syntax if their specificationTable has been set manually).
- Syntax highlighting uses the settings of ***RBCodeHighlighting*** if it is loaded.

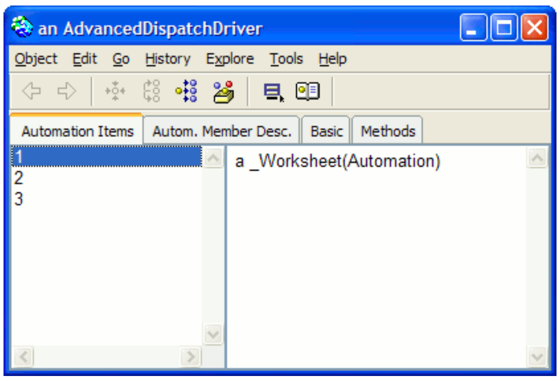
Further the tab displays a help button which opens a help on the selected member if the Automation Server provided a help file for the server.

An additional feature of the "Automation Member Description" tab is the Send/Send-And-Dive Functionality you may already know. This Automation version of the send functionality allows calling parameterized methods/properties by providing a dialog which allows entering parameter values. An Automation send can be performed by selecting "Send/Send and Dive" from a member's context menu.

Inspector: Automation items tab

For collection coclasses, an additional tab will be displayed which provides access to all items of the collection.

Editing items is supported in the inspector (if the coclass supports it) but make sure the correct type of object is used.



The AdvancedDispatchDriver class

Features:

The ***AdvancedDispatchDriver*** class which has already been mentioned is a ***COMDispatchDriver*** subclass which has been enhanced in several ways. First it provides support for finalization. It isn't necessary to release the driver manually for freeing the remote Automation object. The second difference is that such dispatch drivers automatically fetch their specification table as required for supporting the simplified syntax.

- ***Simplified syntax*** means that instead of using ***#invokeMethod:***, ***#getProperty:*** or ***#setProperty:*** methods you simply call the Automation method as if it was implemented by the Smalltalk class. This also has the advantage that you don't have to create parameter arrays when you call functions with more than three parameters.
- You don't have to bother about performance. The specification tables are cached and only created once per session. The cache is cleared at each image startup to ensure the tables are up-to-date.