MODULE 12

COMMANDS

MODULE TOPICS

WPF Command Model

Bindings, Sources, and Targets

ICommand Interface

The Command Library

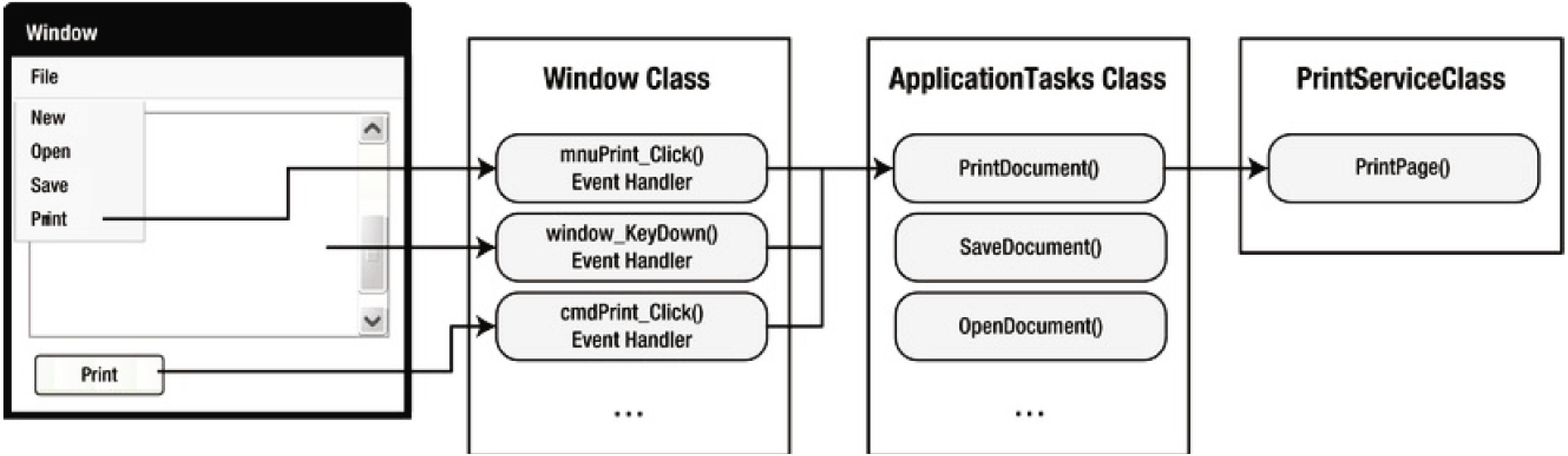
Executing Commands

Disabling Commands

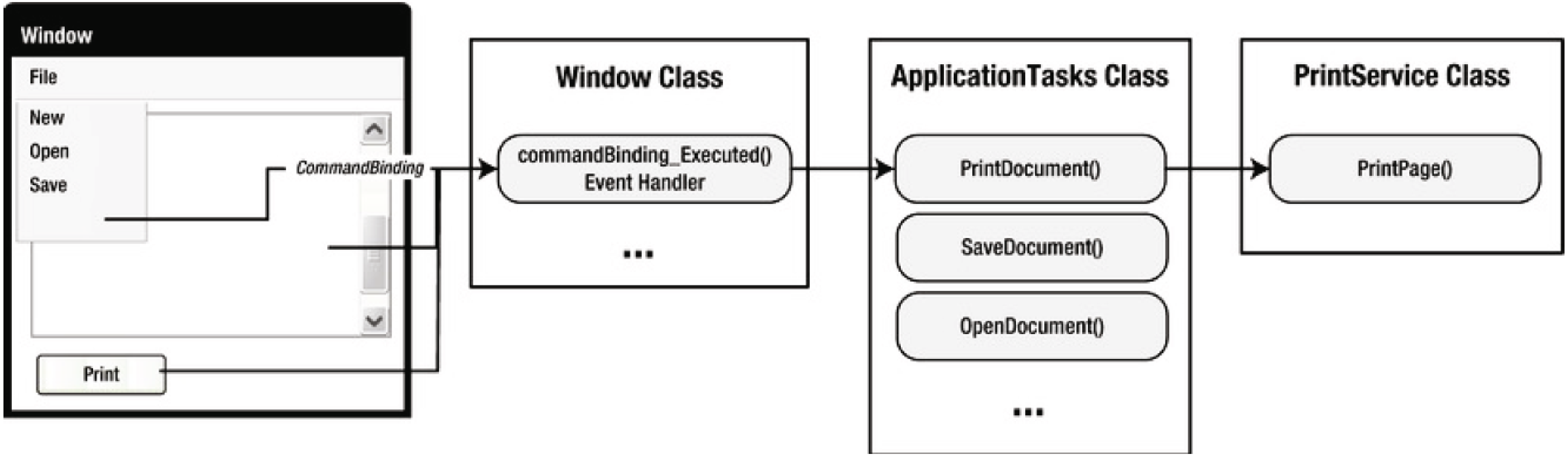
WPF COMMAND MODEL

- Separate the elements that invokes a command from the logic that executes the command
- Commands can be disabled and the elements will also be disabled

EVENT HANDLER MODEL



COMMAND MODEL



WPF COMMAND MODEL

- 4 Pieces to a Command
  - Commands - Represent application task and keeps track of whether it can execute
  - Command Bindings - Links a command to the related application logic
  - Command Sources - Triggers a command (MenuItem, Button)
  - Command Targets - Element on which the command is being performed (TextBox for a paste operation)

ICOMMAND INTERFACE

```
public interface ICommand
{
    void Execute(object parameter);
    bool CanExecute(object parameter);
    event EventHandler CanExecuteChanged;
}
```

COMMAND LIBRARY

- WPF includes a basic command library that has more than 100 commands
    - These commands are exposed through the static properties of five dedicated static classes

APPLICATION COMMANDS

This class provides the common commands, including clipboard commands (such as Copy, Cut, and Paste) and document commands (such as New, Open, Save, SaveAs, Print, and so on)

NAVIGATION COMMANDS

This class provides commands used for navigation, including some that are designed for page-based applications (such as BrowseBack, BrowseForward, and NextPage) and others that are suitable for document-based applications (such as IncreaseZoom and Refresh)

EDITING COMMANDS

This class provides a long list of mostly document-editing commands, including commands for moving around (MoveToLineEnd, MoveLeftByWord, MoveUpByPage, and so on), selecting content (SelectToLineEnd, SelectLeftByWord), and changing formatting (ToggleBold and ToggleUnderline)

COMPONENT COMMANDS

This includes commands that are used by userinterface components, including commands for moving around and selecting content that are similar to (and even duplicate) some of the commands in the EditingCommands class

MEDIA COMMANDS

This class includes a set of commands for dealing with multimedia (such as Play, Pause, NextTrack, and IncreaseVolume)

APPLICATION COMMANDS

New	Copy	SelectAll
Open	Cut	Stop
Save	Paste	Help
SaveAs	Delete	CorrectionList
Close	Paste	ContextMenu
Print	Delete	Properties
Undo	Find	PrintPreview
Redo	Replace	CancelPrint

EXECUTING COMMANDS

```
<Button Command="ApplicationCommands.New">New</Button>
```

COMMAND BINDINGS

```
CommandBinding binding = new CommandBinding(ApplicationCommands.New);
binding.Executed += NewCommand_Executed;
this.CommandBindings.Add(binding);
```

```
<Window.CommandBindings>
    <CommandBinding Command="ApplicationCommands.New" Executed="NewCommand_Execu
</Window.CommandBindings>
```

DISABLING COMMANDS

```
CommandBinding binding = new CommandBinding(ApplicationCommands.Save);
binding.Executed += SaveCommand_Executed;
binding.CanExecute += SaveCommand_CanExecute;
this.CommandBindings.Add(binding);
```

```
void SaveCommand_CanExecute(object sender, CanExecuteRoutedEventArgs e)
{
    e.CanExecute = isDirty;
}
```

WALKTHRU - COMMANDS

ANY QUESTIONS?