IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.11774

Comparative Study on WPF and Win-Forms Frameworks

Yashavanth K L[1], Prof. Poornima Kulkarni[2]

Undergraduate Student, Department of Information Science and Engineering,

RV College of Engineering, Bengaluru, India[1]

Assistant Professor, Department of Information Science and Engineering,

RV College of Engineering, Bengaluru, India[2]

Abstract: In this work, we will focus on two development tools provided by Microsoft for creating Windows applications. Win-forms are quite old and very popular since the days of Visual Basic 6. However, Microsoft has now developed a new technology called Windows Presentation Foundation (WPF). WPF is relatively recent compared to Win-Forms and was first released as a component of the .Net Framework with .Net 3.0. Microsoft has never made it clear that WPF will replace Win-Forms or that the two technologies will coexist. Each tool has advantages and disadvantages over the other. Therefore, we will provide a comparison of the two tools in this work. This enables the developer to choose the best tool to create the aforementioned application.

Keywords: Procedural Approach, Design Options, Controls, WPF,Triggers, Layout, Win-Forms, Memory, Templates, Animation, Control Design, User Interface, Framework & Skinning Structure.

I. INTRODUCTION

Although WPF has been available for a while, it is still a fairly new tool compared to Win-Forms. There are still many developers working today who are not familiar with WPF. Similar to Win-Forms, Windows Presentation Foundation (WPF) is a GUI framework used to build Windows applications. Microsoft has introduced both technologies. A component of Microsoft is WPF, .Net Framework 3. As we've already stated, both tools are used to perform essentially the same task—developing Windows applications—but WPF still has many distinctions from Win-Forms [1]. The main difference between them is that WPF can be built from scratch and has less dependency on the Windows environment than Win-Forms, which is an additional layer on top of basic Windows controls. Here we examine the differences, advantages and disadvantages of the two tools in more detail. As a result, readers of this paper can easily and without doubt decide which one is best and should be used for their application.

II. WINDOWS FORMS (WIN-FORMS)

Windows client applications are built on the Windows Forms platform. This platform offers a collection of managed libraries to give developers of Windows programs a clear, object-oriented, extensible set of classes [3]. A graphical application programming interface (API) is used to manage user interaction with the application and facilitate data display. Windows Forms has a substantial client library that provides an interface to access the Windows operating system's graphical UI and coding that manages visuals. In Win-Forms, each control is an instance of a class [4]. Its structure and behavior are controlled by strategies. It offers an event-driven architecture, allowing applications created using Windows Forms to execute only after certain values given by the user or waiting for certain events to occur.Win-forms offers the option to create new controls and has a selection of controls accessible [5]. Classes exist for designing fonts, brushes, icons, and other graphic elements. The Windows Forms Designer tool from Visual Studio allows users to add controls to forms, organise them in the appropriate layout, use event handlers, and add code to implement user interaction as needed [6]. Win-Forms application settings can also be used to create, store, and preserve run-time state data in XML format. Depending on the user's needs, this type of information can be used to derive application settings, which can then be reused for additional applications [7].

Advantages of Windows Forms:

- If needed, extensive documentation is available on the Internet.
- Tested code and working examples are available in abundance everything is tested thoroughly for years.
- Some support for WPF is also available.

DOI: 10.17148/IJARCCE.2022.11774

- lots of 3rd party controls are available in the market to make your job easier.
- Visual Studio designer is somewhat easier for Winforms as compared to WPF, where lots of work has to be done by the programmer to get exactly what is desired.

Disadvantages of Windows Forms:

- It needs a lot of work to get desired look and feel. So, be prepared to take lots of pain to get it.
- If third party controls are needed definitely it means some extra cost will be added to your budget

III. WINDOWS PRESENTATION FOUNDATION (WPF)

An application running in WPF starts in two threads namely: UI thread and the background thread. thread creates the messaging system and maintains an operations queue using System. Threading. Dispatcher Object. The background thread is used to manage the rendering engine.

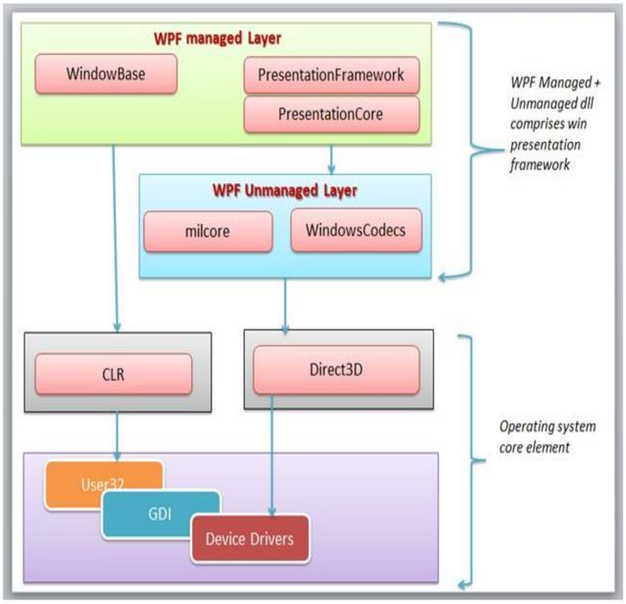

Figure 1. Overall Architecture of WPF

As depicted in the above diagram, WPF is divided into three main sections namely: Presentation framework, User interface framework, and Data framework., Media Integration Layer(Milecore) and the Presentation core.Mil is an abbreviation for media integration library. This section belongs to unmanaged code because of the fact that it behaves like an overpass between DirectX / User32 unmanaged API and WPF managed. DirectX is used internally by the WPF. DirectX communicates with drivers and provides the content.

Managed Layer- Three main elements of the managed layer are Presentation Framework, Window Base, and Presentation Core. . WPF's managed code is its main component.

- Presentation Framework- A high level feature such as application windows, panels, controls, layouts, and content that aids our construction of the applications that are a part of this section. Data binding, time-dependencies, animations, and much more are also included in the end-user presentation.
- Presentation Core- This section is exposed to low-level API.WPF's presentation core implements UI Element and visual services, which include a managed wrapper for MIL.
- Window Base- Basic elements such as Dispatcher objects and Dependency objects are a part of this section.

Unmanaged Layer- This layer is made up of milCore and Windows codes

- milCore- Unlike the WPF managed code, this part of the code is unmanaged. The milCore provides the basic support for 2D and 3D surfaces.

DOI: 10.17148/IJARCCE.2022.11774

Core operating system Layer(Kernel)

- **DirectX-** WPF renders all graphics through DirectX which is a low level API.. DirectX renders the content by communicating with the drivers.
- **User32-** User32 is utilized by every application since it is a primary core. It handles memory separation and process separation.User32 decides which element will appear where on the screen..
- **GDI-** the Common Language Runtime (CLR) is used in WPF to take advantage of the full .NET Framework and to render the graphics more efficiently. GDI: The GDI interface facilitates the addition of a broader set of graphics primitives and various improvements in rendering quality.
- **Device Driver-** There's a dedicated device driver for every operating system... Applications access low-level APIs through device drivers.

PROGRAMMING MODEL OF WPF

- **XAML only:-**Design-time compilation is not required to load a raw XAML "page" into Internet Explorer It facilitates for static content, data binding, and animation. There are no codes or scripts allowed in XAML. All of these tasks require programming, including event handling, data access, service calls, and file access. It facilitates for intelliSense, color coding, and compile-time checking. C# and VB code can be embedded into script blocks inside XAML.
- **Code only:-** All the objects in WPF are defined as .NET classes. Using programming language model ,users can instantiate which is similar to Windows Forms. It is ineffective. This model will use XAML mark-up for all design tools. A more neat and concise representation of element hierarchy is possible in XAML
- **XAML and Code:-**When it comes to initial property setting and static UI layout, XAML is quite expressive. It can be read by humans. It is easier for tools to parse. The dynamic behaviour of applications requires programmatic coding.
- **XAML and code separation:-**Separating the code to be handled from the XAML is one of WPF's key characteristics. Designers can independently work on the application's presentation, while developers can create code logic without worrying about how the presentation looks.

Advantages of WPF:

- Very powerful in styling.
- Provides skinned structure for form design.
- Environment dependency is less so it's easy to create own look and feel.
- Ability to reuse existing code.
- Data binding is simple yet highly advanced.
- Much quicker to develop normal apps.
- Forms are probably simple,
- once you are experienced.
- Futuristic technology for developing advance applications.
- Supports declarative as well as procedural approach.
- Procedural methods can be used as and when required.
- Framework is much smarter and requires quite less code behind to do general things.
- Apps are much more maintainable due to good data binding and effective declarative code.
- Excellent layout is provided.
- Document support is very good and efficient.
- Multimedia, Audio and video support is very good.
- Two dimension and three dimension graphics is supported.
- Also provides support for Win-Forms control.

Disadvantages of WPF:

- Requires .Net 3.0 or higher.
- Compared to Win-Forms still some of the controls and definitions are missing.
- Available support is less as compared to Win-Forms.
- Requires DX9 compatible graphics card to support advanced graphics.
- Technology is quite new so learning curve is quite high.

IV. Comparative Study

Both WPF and Windows Forms, as we saw with their introduction, are used in the creation of client-side Windows programs, and both have their advantages and disadvantages. Hence, it is impossible to draw a firm distinction between

DOI: 10.17148/IJARCCE.2022.11774

the two when it comes to creating Windows applications. Instead of making a decision based on a single option, one must consider the needs of the application. In order to better support his or her choice, we will now create a table-by-table comparison of the two tools. This comparison will help us determine which tool is better able to handle the functionality that an application requires.

Table 1: Comparative analysis of Windows Presentation Foundation and win-Forms

Technology	**Win-Forms**	**WPF**
Multimedia Support	No	Yes
Skinned Form Design	No	Yes
Form Styling	Weak	Powerful
Graphical Interfacelike from, user controls etc	Yes	Yes
DesigningOwn Look	Hard	Easy
Reusabilityof code	Less	High
DataBinding	Simple	Advanced
Codebehind work	High	Less
LearningCurve	Small	High
Procedural Programming Support	Yes	Yes
Documentation availability	Extensive	Medium
Availability of 3rd party tools	High	Low
On Screen Document support	Yes	Yes
Use of Declarative Programming	Less	High
FixedFormat Documents	No	Yes
Images	No	Yes
Futuristic Technology	No	Yes
XAML support	No	Yes
Effort in BindingData	High	Less
2-D and 3-D Graphics	No	Yes

.Net 3.0 or Higher	Not Required	Required
DX9 Compatible GraphicCard	Not Required	Required
Windows Environment Dependency	More	Less
Tools and Controls available	More	Less
Technology	Old	New
Memory Consumption	Less	High
CPU Load	Less	High
Load Time	High	Less
Availability of Help Forums & Blogs	More	Less
RoutedEvents	No	Yes
Video & Audio	No	Yes

When choosing between WPF and Win-Forms to build an application, Table 1 tabular comparisons between the two can be helpful. One thing is certain, though: switching to a single tool for all of your solutions is not a wise idea. However, it is advisable to consider the requirements of your application before deciding which tool to use. For example, if your application requires extensive multimedia capabilities, very sophisticated data bindings, or if speed is more important to you than memory utilisation, WPF is unquestionably the best option. However, if the situation is reversed and preserving memory for your programme is considerably more critical than speed, you must use Windows Forms.

V. CONCLUSION

After weighing the pros and cons of the two tools, it becomes clear that choosing one tool for all purposes is not the best option. Instead of using a single tool in isolation, always consider the application requirements and look for the factors listed in Table 2.After weighing the pros and cons of the two tools, it becomes clear that choosing one tool for all purposes is not the best option. Instead of using a single tool in isolation, always consider the application requirements and look for the factors listed in Table 2. Win-Forms is a good choice if you generally support the aspects you're looking for because it's simple to find resources and get support for it if you need it. However, if your application includes some features that are only supported by WPF and not by Win-Forms, you should choose WPF. Speed and memory usage can also affect the choice. Win-Forms is unquestionably your top choice if memory usage is your main concern, but otherwise WPF provides faster performance. Therefore, the best tool depends primarily on the needs of the application.

ACKNOWLEDGMENT

The authors of this paper express their gratitude to **Mr. Sharath Mahadev** and his colleagues of Talcuz Solutions for their time and guidance.

REFERENCES

[1]. Chaudhuri, Surajit and Umeshwar Dayal, "An overview of data warehousing and OLAP technology." ACM Sigmod Record, Vol. 26, Issue 1, 1997, pp: 65-74.

[2]. Jörg Thomas, and Stefan Dessloch. "Formalizing ETL jobs for incremental loading of data warehouses." Datenbanksysteme in Business, Technologie und Web (BTW)–13. Fachtagung des GI-Fachbereichs" Datenbanken und Informationssysteme"(DBIS), 2009.

DOI: 10.17148/IJARCCE.2022.11774

[3]. Mekterović, Igor, and Ljiljana Brkić, "Delta view generation for incremental loading of large dimensions in a data warehouse." In Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2015, pp: 1417-1422.
[4]. Tank, Darshan M., Amit Ganatra, Y. P. Kosta, and C. K. Bhensdadia, "Speeding ETL processing in data warehouses using high-performance joins for Changed Data Capture (CDC)." In Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing, IEEE, 2010, pp: 365-368.
[5]. Wiener, J., and J. Naughton, "Incremental loading of object databases", Stanford InfoLab, 1996.
[6]. Jörg T, Dessloch S, "Towards generating ETL processes for incremental loading", In Proceedings of the 2008 international symposium on database engineering applications (IDEAS'08), ACM, 2008, pp: 101–110.
[7]. Behrend, Andreas, and Thomas Jörg, "Optimized incremental ETL jobs for maintaining data warehouses.", In Proceedings of the Fourteenth International Database Engineering & Applications Symposium, 2010, pp: 216-224.
[8]. Bokade MB, Dhande SS, Vyavahare HR, "Framework of change data capture and real time data warehouse", International journal of engineering research and technology, vol 2. ESRSA Publications, 2013.
[9]. Ankorion, I, "Change data capture efficient ETL for real-time bi", Information Management, Vol. 15, Issue 1, 2005, p.36.
[10]. Lindsay B, Haas L, Mohan C, Pirahesh H, Wilms P, "A snapshot differential refresh algorithm", In Proceedings of the ACM-SIGMOD conference, Vol. 15, Issue 2, 1986, pp: 53-60.
[11]. Labio W, Garcia-Molina H, "Efficient snapshot differential algorithms for data warehousing", In Proceedings of the 22nd inter- national conference on very large data bases (VLDB'96), Morgan Kaufmann Publishers Inc, 1996, pp: 63–74.
[12]. S. Kozielski and R. Wrembel, "New Trends in Data Warehousing and Data Analysis", Annals of Information Systems, Springer, Vol. 3, 2008, pp: 19-49.
[13]. M. Bokun and C. Taglienti, "Incremental Data Warehouse Updates," Information Management Magazine, 1998.
[14]. Liu Jun, Hu ChaoJu, Yuan HeJin, "Application of Web Services on The Real-time Data Warehouse Technology", Advances in Energy Engineering (ICAEE), 2010 International Conference, 2010, pp: 335 – 338.
[15]. Eccles, Mitchell J., David J. Evans, and Anthony J. Beaumont, "True real-time change data capture with web service database encapsulation.", 6th World Congress on Services. IEEE, 2010.
[16]. C. R. Valêncio, M. H. Marioto, G. F. Donega Zafalon, J. M. Machado and J. C. Momente, "Real Time Delta Extraction Based on Triggers to Support Data Warehousing," In Proceedings of The International Conference on Parallel and Distributed Computing, Applications and Technologies, 2013.
[17]. Sukarsa, I. Made, Ni Wayan Wisswani, and IK Gd Darma, "Change data capture on OLTP staging area for nearly real time data warehouse base on database trigger.", International Journal of Computer Applications, Vol. 52, Issue 11, 2012, pp: 32-37.
[18]. Denny, Atmaja, I. Putu Medagia, Ari Saptawijaya, and Siti Aminah, "Implementation of change data capture in ETL process for data warehouse using HDFS and apache spark.", In Proceedings of 2017 International Workshop on Big Data and Information Security (IWBIS), IEEE, 2017, pp. 49-55.
[19]. Biswas, Neepa, Anamitra Sarkar, and Kartick Chandra Mondal, "Efficient incremental loading in ETL processing for real-time data integration.", Innovations in Systems and Software Engineering, Vol. 16, Issue 1, 2020, pp: 53-61.
[20]. Vijayalakshmi, M., and R. I. Minu, "Incremental Load Processing on ETL System through Cloud.", In Proceedings of the International Conference for Advancement in Technology (ICONAT), IEEE, 2022, pp: 1-4.
[21]. Chakraborty, Sonali, "A Novel Approach of Using Materialized Queries for Retrieving Results from Data Warehouse." In Proceedings of the International Conference on Intelligent Vision and Computing, Springer, Cham, 2022, pp: 22-35.
[22]. Alkhayat, Zainab, and Kadhim BS Aljanabi, "Optimal data warehouse design with data marts and data cube aggregation." In Proceedings of 3rd International Scientific Conference of Alkafeel University, Vol. 2386, Issue 1, 2021, pp: 050012.
[23]. R. Kimball and J. Caserta, The Data Warehouse ETL Toolkit : Practical Techniques for Extracting,Cleaning,Conforming,and Delivering Data, ser. The Data Warehouse Toolkit. Wiley Publishing, Inc, 2004.
[24]. M. Bokun and C. Taglienti, "Incremental Data Warehouse Updates," Information Management Magazine, 1998.