

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Page Life-Cycle of ASP.Net

Abhilasha Raval

M.C.A. Department

IGNOU

Gujarat – India

Abstract: The life cycle starts when a user requests a web page through his/her browser. The Web server then process the page through a sequence of steps before response is sent back to the user's browser.

When an ASP.NET page runs, the page goes through a life cycle in which it performs a series of processing steps. These include initialization, instantiating controls, restoring and maintaining state, running event handler code, and rendering. It is important for you to understand the page life cycle so that you can write code at the appropriate life-cycle stage for the effect you intend.

Keywords: Page Request, Start, Page Initialization, Load, Validation, post back Event Handling, Rendering, Unload.

I. INTRODUCTION

The page request occurs before the page life cycle begins. When the page is requested by a user, ASP.NET determines whether the page needs to be parsed and compiled or whether a cached version of the page can be sent in response without running the page. In the start step, page properties such as Request and Response are set. At this stage, the page also determines whether the request is a post back or a new request and sets the IsPostBack property.

II. GENERAL PAGE LIFE CYCLE STAGE IN ASP.NET



Start

In the start step, page properties such as Request and Response are set. At this stage, the page also determines whether the request is a postback or a new request and sets the IsPostBack property. Additionally, during the start step, the page's UICulture property is set.

Page Initialization

During page initialization, controls on the page are available and each control's UniqueID are generated but not their properties.

If the current request is a post back, the post back data has not yet been loaded and control property values have not been restored to the values from view state.

Load

During load, if the current request is a postback, control properties are loaded with information recovered from view state and control state.

The OnLoad event method is fired during this stage.

This is where you will want to set properties for all of the server controls on your page, request query strings, and establish database connections.

Validation

During validation, the Validate method of all validator controls is called, which sets the IsValid property of individual validator controls and of the page.

PostBack Event Handling

If the request is a postback, any event handlers are called. The event handling for server controls occurs during this stage.

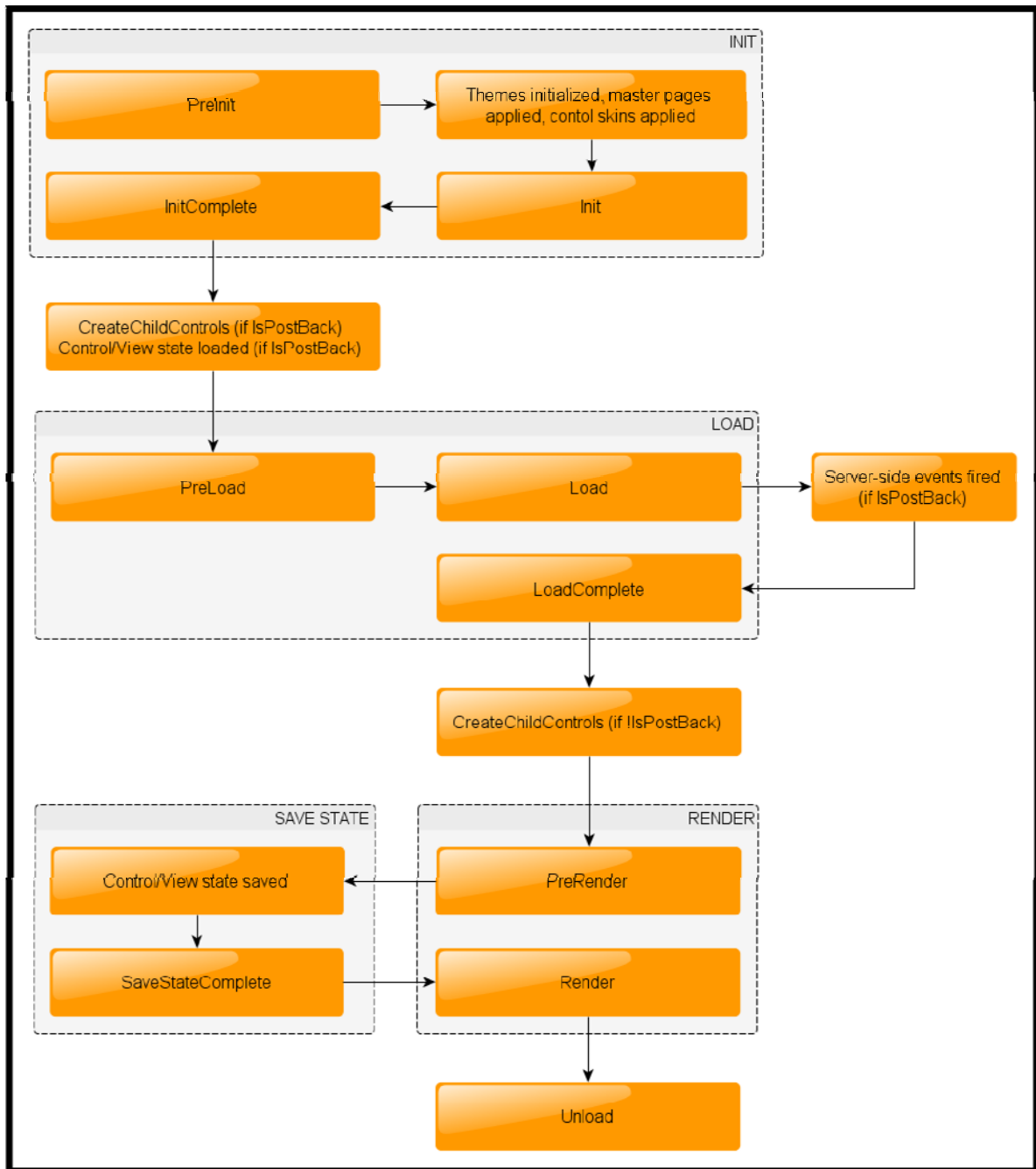
Render

During rendering, view state is saved to the page and all controls. During the phase, the page call the render method for each control, providing a text writer that writes its output to the output stream of the page's response property.

Unload

Unload is called when the page has been fully rendered, sent to the client, and is ready to be discarded. At this point, page properties such as Response and Request are unloaded and any cleanup is performed. The cleanup includes routines such as closing database connections and file streams, or, event logging and other tasks.

III. PAGE LIFE CYCLE EVENT

**PreInit:**

Purpose of use event:

- Check the IsPostBack property to determine whether this is the first time the page is begin processed.
- Create or re-create dynamic controls.
- Set a Master page dynamically.
- Set the them property dynamically.
- Read or set profile property values.

Init:

Raised after all controls have been initialized and any skin settings have been applied. The Init event of individual controls occurs before Init event of page.

Init complete

This event is raised at the end of the page's initialization stage ,Use this event to make changes to view state that you want to make sure are persisted after the next post back

PreLoad:

Raised after the page loads view state of itself and all controls

Load:

The page calls the OnLoad event method on the page ,then recursively does the same for each child control, which does the same for each of it's child controls until the page and all control are loaded.

Use the OnLoad event method to set properties in controls and establish database connections.

Control events:

Use these events to handle specific control events, such as Button's Onclick event. Controls have their events invoked at this stage such as responding to a button click, the selected index changing on a list box and so on.

If your page has validation controls, check the IsValid status of the controls and the page before continuing with your event handler

LoadComplete

Use this event for task that require that all other controls on the page be loaded.

PreRender

Raised after the Page object has created all controls that are required in order to render the page, including child controls of composite controls.

SavestateComplete

Before this event occurs,viewstate has been saved for the page and for all controls.any changes to the page or controls at this point will be ignored.

Use this event perform tasks that require view state to be SAVED, BUT that do not make any changes to controls.

Render

This is not an event; instead, at this stage of processing, the Page object calls this method on each control. All ASP.NET Web server controls have a Render method that writes out the control's markup to send to the browser.

Unload

This event occurs for each control and then for the page.in controls, use this event to do fonal cleanup for specific controls, such as closing control specific database connections.

For the page itself, this event to do final clean-up work, such as closing open files and database connections, or finishing up logging or other request- specific task.

IV. CONCLUSION

When an ASP.Net page runs, the page goes through a life cycle. These include initialization, instantiating controls, restoring and maintaining state, running event handler code, and rendering.

In order to memorize whole page life cycle process of Asp.net, keep store this word "**SILVER**" in your memory, which is defined as **S** = Start, **I** = Initialization, **L** = Load, **V** = Validate, **E** = Event Handlers, **R** = Render

Some parts of the life cycle occur only when a page is processed as a postback. For postbacks, the page life cycle is the same during a partial-page postback as when you use an **UpdatePanel** control as it is during a full-page postback.

References

1. Beginning ASP.NET 4 in C# ,Auther Name: Imar Spaanjaars,
2. Pro ASP.NET 4 in C# 2010 ,
3. Professional ASP.NET 4 in C# and VB