ASP.NET Page Life Cycle

A page in an ASP.NET application consists of several server controls. These are the fundamental building blocks of an ASP.NET application. The Life cycle of an ASP.NET page, depends on whether the page is requested for the *first time or it is a postback*. Postback is a process by which a page can request for itself.

➤ *When the Page is requested for the first time*

The Life Cycle of a page when requested for the first time:

Initializing: During this phase, the server creates an instance of the server control

Loading: During this phase, the instance of the control is loaded onto the page object in which it is defined.

PreRendering: During this phase, the control is updated with the changes made to it. This prepares the control for rendering.

Saving: During this phase, the state information of the control is saved. For example, if a value is set for the control during the Load event, it is embedded in the HTML tag that will be returned to the browser.

Rendering: During this phase, the server creates the corresponding HTML tag for the control.

Disposing: During this phase, all cleanup tasks, such as closing files and database connections opened by the control are performed.

Unloading: During this phase, all cleanup tasks, such as destroying the instances of server control are performed. This is the final event in the life cycle of a server control

➤ *Life cycle when the page processed during a postback event*

The processing sequence in which a page is processed during a postback event is:

Initializing: During this phase, the server creates an instance of the server control

Loading view state: During this phase, the view state of the control posted by the client is reloaded into the new instance of the control.

Loading: During this phase, the instance of the control is loaded onto the page object in which it is defined.

Loading the postback data: During this phase, the server searches any data corresponding to the control that is loaded in the data posted by the client.

PreRendering: During this phase, the control is updated with the changes made to it. This prepares the control for rendering.

Saving state: During this phase, the change in the state of control between the current request and the previous request of the page is saved. For each change, the corresponding event is raised. For example, if the text of a textbox is changed, the new text is saved and a text_change event is raised.

Rendering: During this phase, the server creates the corresponding HTML tag for the control.

Disposing: During this phase, all cleanup tasks, such as closing files and database connections opened by the control are performed.

Unloading: During this phase, all cleanup tasks, such as destroying the instances of server control are performed. This is the final event in the life cycle of a server control

The events associated with the relevant page cycle phases are:

- Page Initialization: Page_Init
- View State Loading:LoadViewState
- Postback data processing: LoadPostData
- Page Loading: Page_Load
- PostBack Change Notification: RaisePostDataChangedEvent
- PostBack Event Handling: RaisePostBackEvent
- Page Pre Rendering Phase: Page_PreRender
- View State Saving: SaveViewState
- Page Rendering: Page_Render
- Page Unloading: Page_UnLoad

The following figure illustrates how the server controls on an ASP.NET page is processed by the server:

New Web Server

Control instance

User browses
ASP.NET
Page

1. Control
initialized
2. Load event
called
3. PreRender
Event called
4. Save State
Event called
5. Render Event
called
6. Dispose Event
called

HTML for control
returned to User's
browser with state
data

Web server

Control Instance

User interacts with HTML
causing a postback

Postback cycle repeats itself

1. Control initialized
2. Load State Event
called
3. Load Event called
4. PostBack Event
called
5. PreRender Event
called
6. Save State Event
called
7. Render Event called
8. Dispose Event
called

HTML
control
returned to
User's
browser with
state data