

Web Application Configuration Parameters

ApplinX provides a Web based configuration editor where you can configure framework parameters. These parameters are saved in the *config/gx_appconfig.xml* file. A description of each parameter appears in the Javadoc on the right side of the window (click on a parameter/node to display the relevant Javadoc).

Note:

These parameters can be manually configured in the *config/gx_appconfig.xml* file.

This chapter covers the following topics:

- Session Parameters
- Instant Parameters
- General
- Logoff
- Generated Pages
- Window
- Emulation
- Natural upload/download
- Log
- Performance Monitor
- Macro
- Single Sign On
- FTP
- CSS Classes

Session Parameters

Parameter	Description
Server URL	The ApplinX server URL. By default it is applinx://localhost:2323. For secured SSL socket: applinxs://localhost:23443.
Application name	The name of the ApplinX application.
Session ID	The Session ID in the web server. Can be set according to the connecting IP address (IPv4 and IPv6 address formats are supported) or by an allocated session ID. It is also possible to set the Web Session ID from the code (in the GXBasicContext (JSP)/GXBasicWebForm (.NET) file), overriding the option selected here.
Password	The password required to access a specific session on the ApplinX server. Can be set according to the connecting IP address or by an allocated session ID. It is also possible to set the password from the code (in the GXBasicContext (JSP)/GXBasicWebForm (.NET) file), overriding the option selected here.
Additional	
Description	The description of the session in ApplinX server. This description appears in the list of sessions in ApplinX Administrator. The information entered can be dynamic and may include tokens such as: \$(SESSION_ID): Represents the Web server session ID. \$(IP): Represents the user IP address. \$(BY_APPLINX): Represents the ApplinX session ID.
Connection Pool	The connection pool name used to get a session. When a connection pool name is stated, the framework will connect to the connection pool instead of using a new session.
Host user name	The host user name. Relevant for SSH and Natural-Unix protocols only.
Host password	The host password. Relevant for SSH and Natural-Unix protocols only.
Show Intermediate screens	Determine whether to show intermediate host screens. Relevant for Natural Set Control N command.

Instant Parameters

Parameter	Description
Font size	It is possible to change the default font size to a different size (in pixels). Since the change affects the rendering process performed by the framework on the Web server, this setting is not part of the CSS style sheet file (unlike other font settings), but instead it is included in the server-side instant configuration parameters. Note: Changing the font sizes for generated Web pages is done in the CSS style sheet node. The possible values include the different pixel possibilities as well as "Dynamic by Resolution". The HTML emulation recognizes user resolution and dynamically changes the css name according to the matching resolution. The resolution can be either 640x480, 800x600, 1024x768 or 1280x1024 pixels. The resolution may also depend on the host resolution (80 or 132 characters). The best-fit functionality is configured by default in the emulation template in all frameworks. It is used by the renderer to calculate the top/left according to each tag's position.
Font family	Sets the font family to be used for instant rendering. It is used for proportion calculation to best fit specific fonts for Japanese & Arabic applications. For Japanese the best fitted fixed fonts are: "MS Mincho" or "MS Gothic" For Arabic the best fitted fixed font is: "Courier". For other languages the best fitted font is "Courier New", which is by default declared in the CSS files. Defining the font here will override the CSS definition.
Row spacing	Sets the row height to be used for instant rendering. This height is calculated relatively to the row's font size. The default row height is set to 53% (153% of the font size). For example: When the font size is set to 13px, and using the default row height value of 53%, the row height will be: $13 * 1.53 = 19.89 = \sim 20\text{px}$.
Color mode	Determines whether to display all the colors as they are displayed in the host (background & foreground), display only the background colors or not display any colors at all.
Host keys	The type of the rendered host keys.
Render tags from right	Determines that the HTML tags will be positioned from right to left. The usage is in Hebrew/Arabic applications when using non-fixed font, to ensure alignment to the right. By default the tags are positioned from left to right.

Parameter	Description
Render emulation behavior	<p>Determines whether to render emulation attributes in text fields. Emulation attributes are special ApplinX attributes that start with "gx_" and are added to the rendered text field tag. These attributes provide additional information to the ApplinX JavaScript engine, which changes the behavior of the text fields. The attributes are:</p> <ul style="list-style-type: none"> ● Automatic skipping (gx_autoSkip) - will cause the text field to skip to the neighboring text field, once it is filled with content. ● Data type (gx_dt) - The data type of the field. See GXBaseObjectConstants for possible data types. ● Right adjustment (gx_ra) - AS/400 right adjust. ● Automatic ENTER (gx_au) - AS/400 automatic enter.
Enable displaying tables	Determines whether the relevant transformation (such as displaying tables, or displaying the graphical window frame) is enabled.
Enable displaying graphical window frames	Determines whether the relevant transformation (such as displaying tables, or displaying the graphical window frame) is enabled.
Define render area	<p>Used to define that a specific area of the host screen will be rendered as HTML in Instant HTML pages by specifying the area's boundaries and in this way hide certain portions of the screen. By default, the entire screen is configured to be rendered to the HTML output. When setting the area to be rendered, tags positioned outside the rendered area will not be available for query execution on the screen tag model. The rendered area also affects the rendered top/left attributes of the tag. The top/left tag will be calculated relatively to the starting position of the rendered area.</p>

General

Parameter	Description
Use Folders - virtual directory	<p>Determines whether the Web application folder structure should be the same folder structure as configured in the ApplinX repository. Relevant for screens only. This is recommended when the Web application contains a large number of designed Web pages.</p> <p>Virtual directory: Determines the virtual directory used for the project. Should be without any slashes. Needs to be set only if UseFolders is true. For a site (with domain) the value should be empty string.</p> <p>For example:</p> <ul style="list-style-type: none"> ● If the web application URL is <http://localhost:8080/myApp/index.jsp>, the virtual directory is "myApp". ● Also for a web site http://myWebsite/myApp/index.jsp <http://mywebsite/myApp/index.jsp> , the virtual directory is "myApp". ● If for example I have two folders <http://localhost:8080/myApp/mySubApp/index.jsp>, the virtual directory is "myApp/mySubApp".
Use screen locker	<p>The purpose of a screen locker is to indicate that the application is processing your request, and that you are blocked from interfering with the current process by repressing a button/link or keyboard PF/ENTER. To activate the screen locker first select the Use screen locker check box and then access the template/screenLocker.htm file and use the width/height percentages to control the location of the message. Replace the text "Please wait" with an alternative text/image as required. Refer to the documentation for further details.</p>
Alternate row colors	<p>Determines whether to use alternate colors for every other row in a table. Use AlternateCssName to control the alternate css name.</p>

Parameter	Description
Prevent page refresh effect	This feature prevents the displayed page from flickering every time the page is submitted. This affect is prevented by using two separate frames, while only one is visible at each stage. These frames continuously switch between being active and passive each time a page is submitted. The active frame is displayed, and every time a page is submitted, the frame becomes the passive frame, in this way preventing the flickering affect. When the modal window feature is enabled in the Framework Configuration Editor or the 'Enable Natural-Data-transfer support' is enabled in the host, this feature is automatically enabled even if this parameter is not selected.
Perform background check for host screen changes	Determines whether to continuously check in the background for changes in the host screen. If the host screen has changed, the browser is refreshed and a JavaScript event is triggered. As the server is checked continuously, more server resources are required for this. By default, the server is checked every 3, 6, 12 and then every 24 seconds. These intervals can be customized using user exits (refer to the ApplinX documentation for further details).

Logoff

Parameter	Description
Disconnect host session when browser is closed	Determines whether to log off from the host session when the browser is closed. It is recommended to define a session timeout in the ApplinX Designer (Application Properties dialog box>Host tab, Non-activity timeout) to ensure logging off from a session.
Display the following confirmation message	The confirmation message which is displayed by default is displayed in this field. You can change the message as you see fit.
Prompt user before session time-out	Set to true to display a message indicating to the user that the session is about to be disconnected. This message will be displayed towards the end of the Non-activity timeout period (defined in the Application Properties Host tab). The user will be able to select to resume or quit the session. When not selecting either of these options, the user will be logged off automatically. It is possible to customize the relevant prompt page (template/logoffPrompt.htm).
Termination path	Defines a termination path to use when the user logs off by either closing the browser, or clicking the logoff link (activates logoff.jsp/asp).

Generated Pages

Parameter	Description
Reflect host protected - Dynamically disable CSS	Determines whether the protected fields in the host will be read only in the Web application (when <code>gx_fillForm</code> is called). Dynamically disable CSS: The <code>css</code> class that is used to give a different look to a read-only input field, which becomes read-only according to the value of the <code>ReflectHostProtected</code> parameter.
Reflect foreground colors	Determines whether the field foreground color in the Web application will be the same as the field foreground color in the host. The relevant <code>css</code> classes are in <code>css/styles_generated.css</code> file.
Reflect background colors	Determines whether the field background color in the Web application will be the same as the field background color in the host. The relevant <code>css</code> classes are in <code>css/styles_generated.css</code> file. <code>/css_colors</code> folder in the emulation template.
Reflect maximum field length	Determines whether the fields' maximum length attribute in the host will be the maximum length attribute in the input fields in the Web application. This saves you the work of manually adding this attribute to each input field when designing the page.
Reflect emulation behavior	<p>Determines whether to add emulation attributes in text fields. Emulation attributes are special ApplinX attributes that start with "gx_" and are added to the rendered text field tag. These attributes provide additional information to the ApplinX JavaScript engine, which changes the behavior of the text fields. The attributes are:</p> <ul style="list-style-type: none"> ● <code>gx_autoSkip</code> - will cause the text field to skip to the neighboring text field, once it is filled with content. ● <code>gx_dt</code> - The data type of the field. See <code>GXBaseObjectConstants</code> for possible data types. ● <code>gx_ra</code> - AS/400 right adjust ● <code>gx_au</code> - AS/400 automatic enter

Window

Parameter	Description
Enable modal windows - opened window top, opened window left, opened window attributes, Host key to send when the window is closed	<p>Determines whether to activate ApplinX pop-up manager and transform recognized host windows to Web pop-up windows.</p> <p>Note: This feature is disabled by default for mobile devices such as iPad, iPhone, Android etc. You can override this behavior with a function provided in userExits.js. See Enabling Modal Windows for Mobile Devices</p>
Set window position to center	Determines that the pop-up window is opened in the center of the screen.
Set window specific position	Determines that the pop-up window is opened in a specific position.
Opened window attributes	Determines the opened window's attributes when using the method <code>gx_window.open</code>
Host key to send when the window is closed	Determines the key to send to the host when the modal window is closed

Emulation

Parameter	Description
Support type ahead	<p>Determines whether to enable the user to type continuously, without waiting for the browser to display the next page. This feature uses an ActiveX component (works in IE only). This feature does not work when a modal window is open, or during the process of opening/closing a pop-up window.</p>
Support Dup and FieldMark host keys	Dup and FieldMark are special Mainframe characters. In Instant and generated pages, a user can send to the host a non printable character in specific input fields.
Use keyboard PF keys	Set when the PF keys in the browser are enabled. Default value: true.
Tab on input fields only	Determines whether the TAB key navigates between the input (unprotected) fields of the screen only, or also between additional buttons and hyperlinks in the Web page.

Parameter	Description
Automatic skip for all input fields	<p>Determines whether or not the cursor will automatically move to the next input field once the current input field is filled with text (applies to all input fields).</p> <p>When the <code>autoSkipAllFields</code> property is set to "true" and IME (Input Method Editor) input is on, the cursor moved to the next field when the end of the field is reached. After pressing ENTER, the IME characters are selected and the number of characters is compared with the maximum length of the field. If the number of characters is more then the maximum length, the extra characters are deleted and the cursor moves to the next field.</p>
Block illegal characters in host numeric fields	Determines whether it will be possible to type non-numeric characters in numeric input fields. The type of the field is a property of the host field which the input field is based on.
Select content when focus on input field	Determines whether when moving the cursor to the following field (using the arrows or TAB key), the field content will be selected and then overwritten or the text will not be selected and when typing, the text will be inserted in the field.
Show blinking caret	Displays a blinking caret in input fields. Internet Explorer only.
Show host blinking fields	Some hosts contain definitions determining that certain fields should blink. This feature enables these fields to blink in the browser. This is not supported in generated pages.
Place caret at end of field	Determines whether the caret's position should be at the end of the input fields.
Use paste behavior as in terminal emulators	Determines whether the paste behavior of text will be as in terminal emulators, that once a field is filled, the remaining text is pasted in the following field or whether the text will be pasted in a single field. Only supported in Internet Explorer.
Arrow keys navigation	Determines whether or not it will be possible to navigate throughout the screen using the arrow keys, and whether this navigation will be only between input fields or between protected and input fields.
Label focus style	This option allows you to determine whether the font style of a protected field will become underline, bold or will not change when a user either clicks on the protected field or moves between the protected fields using the keyboard arrow buttons.
Double click label action key	Sends the specified action key to the host when double-clicking on a label.

Parameter	Description
Color set	This functionality enables the end user select one of the pre-defined style sheets for instant screens. In the emulation template this functionality is available by default ("change color set" link in the page footer). These relative css file names must be separated by a comma. For example: css/styles_white.css,css/styles_black.css

Natural upload/download

Parameters	Description
Use ActiveX for Natural data transfer (IE only)	Defining to use the ActiveX component, enables the work flow's behavior to be identical to that of Entire Connection. This is relevant for Internet Explorer only. Notes: <ol style="list-style-type: none"> ActiveX will work only if you have Microsoft .NET Framework version 3.5/4.0 installed on your machine. The site using the ActiveX needs to be defined as a "trusted site". Under Tools > Internet Options > Security, click Custom level and under ActiveX controls and plug-ins > Allow Scriptlets check Enable.
Automatically start download	Determines whether to automatically start downloading a file, when the first download screen is displayed. When not selected, a message appears informing you that the host requires you to download a Natural file. If there is a record count from the download, it will be displayed on the page.
Download file in one go/Download file in blocks	According to the selected radio button, determines whether to download the file as a whole, or whether to download the file in blocks, appending each block to the existing file. This parameter is typically set when expecting to download large files and is only relevant when working with ActiveX and with a non Natural Unix protocol which supports Natural data transfer.
Default file extension	Enables defining the default file extension: TXT or CSV.

Log

The Framework Log lists the errors and exceptions and can then be used to debug the framework. When in production, the log should be used with discretion.

Parameters	Description
File name	The log is written to this file.
Append to existing file	Selecting this check box determines whether when restarting the Web server, the log file will be overwritten or a new file will be created.
Log level	The contents of the log file are as detailed as this property defines, where every level includes the levels above it. For example, the Debug level also logs Normal, Warnings and Errors Only levels. Available values: "Normal", "Warnings", "Errors only" and "Debug" (by default Normal is selected).
Log history	Determines the number of backups saved before overwriting the old log files. For example: 10 means "save the last 10 log files, in addition to the current one, then start to overwrite".
Max. file size	Starts a new log file after the current file has been filled to the maximum file size.

Performance Monitor

Determines whether to create a performance log. The performance monitor is used for analyzing framework performance. It writes log comments for every framework request. A performance log line contains the following data: Session ID, Screen Name, total request time, total server side divided into: attach, sendKeys, instant generation, detach, total page client load time.

Parameters	Description
Enable performance monitoring	Determines whether to create a performance log.
File name	The log is written to this file.
Write performance log per session ID	When selected, creates a performance log for each session.
Trace sessions	Determines whether to trace all sessions or only the session specified here.
Description	Short description means that the information is displayed in a tabular structure. Detailed description displays the information as a paragraph.

Macro

Parameters	Description
User name	The user macros file will be saved according to the user name, therefore it should be a unique ID of the Web user. The User name can be according to the IP Address, it can be cookie based, or set in the code. By default it is configured to IP Address, which is a good idea when the user's IP is constant. When selecting the cookie based option, ApplinX will assign a cookie to the end user, and will read the macro according to the cookie. Macros will be lost when you delete your cookies. To enable this option you should set SaveUserNameLocally to true. Select Other to use a session variable which was saved according to the login page user name, or to use the NT user name (relevant for .NET only) (Request.ServerVariables["AUTH_USER"])
Macros folder	Determines the folder on the server where the macros will be saved.
Encrypt macro file	Determines whether to encrypt the macro. Recommended to use when the macro includes user names and/or passwords. When changing this setting, existing macros will no longer function. The existing macros will be deleted from the macros folder when recording new macros.

Single Sign On

Parameters	Description
Enable Single SignOn	Determines whether to activate the single sign option.
SignOn screen recognition	Determines the method to use to recognize the sign on screen. According to the screen name. Automatically recognize the user name and password fields. Set from the code.
Screen name	The name of the SignOn screen, when recognizing the signOn screen according to the screen's name.
Define application field names	Returns the application field name which determines the user name field.
User/Password retrieving:	
Source	Determines the type of user/password retriever to use: <ul style="list-style-type: none"> ● Based on information retrieved from an HTTP request. ● Based on information retrieved from an HTTP session. ● Set in the code.
User parameter name	
SignOn execution:	
Type	Determines what happens once the SignOn is recognized: <ul style="list-style-type: none"> ● Using an action key (enter an action key). ● Executing a path (enter a path name) . ● Set in the code.
Action key	Enter the action key that is to be activated

FTP

In ApplinX framework, it is possible to transfer files from the client to the host or from the host to the client, using the FTP dialog screens. The HTML emulation contains a link in the footer that opens an FTP Web dialog box. To upload/download files using the FTP option, you need to configure the following parameters.

Parameters	Description
Host type	Sets the type of the remote FTP host. Possible values are: Mainframe AS400 Other - for any other kind By Applinx - when the FTP host is the same as the host used in the current ApplinX session.
Host address	Sets the IP address of the remote FTP host.

CSS Classes

Parameters	Description
Instant only	
Window frame CSS class	The CSS class of window frame.
Host keys CSS class	The CSS class of the host keys
Table CSS class	The CSS class of the table tag.
Table Odd rows CSS	
Render intensified CSS class	Sets if to render intensified css class for intensified host fields.
Render application field CSS class	Sets if to render the application field css class.
Generated & Instant	
Table even rows CSS class	
Render intensified CSS class	Determines whether to render an intensified css class for intensified host fields.
Render application field CSS class	Determines whether to render a css class for application fields
Table even rows CSS class	
External CSS file parameter name	Defines an external css request parameter name. Used for portal integration. A portal application may provide to the framework as a query string a css URL as follows: http://?cssurl= ("cssurl" will be the value of the field in this case)