

# Web Enablement

- Web Enablement - Pure HTML Emulation
  - Web Enablement - Instant Web Application
  - Web Enablement - Composite Web Application
- 

## Web Enablement - Pure HTML Emulation

ApplinX HTML emulation is an out-of-the-box thin client HTML emulation, available in the .NET and J2EE environments. ApplinX is able to turn an existing terminal emulation instantly into a web browser terminal emulation, providing host key and print support and maintaining existing color schemes.

### List of Subjects Relevant to Pure HTML Emulation

**Note:**

If not indicated otherwise, the sections that are referred to below can be found in the *Web Application Development* documentation.

- *Introducing Pure HTML Emulation Development* (see below).
- *Creating an Application* in the *Designing and Developing an Application* documentation: Create a new ApplinX application (including host and repository).
- *Creating an ApplinX Web Application* in the *Getting Started* documentation: Create an ApplinX HTML emulation web application (JSP or .NET) using the Web Application Manager.
- *The HTML Emulation Toolbar* (see below).
- *Configuring your Web Application*: Configure the framework parameters (for detailed descriptions of the parameters, refer to *Web Application Configuration Parameters*):
  - Session parameters: Configure session parameters such as server URL, application name, session ID, password, description, connection pool name, host user name, host password, and more.
  - Instant parameters: CSS classes for different elements such as font size, font family, row spacing, color mode, host keys, render tags from right, render emulation behavior, enable displaying tables, enable displaying graphical window frames, and define render area.

Step-by-step instructions are available in *Controlling Instant Display Properties* in the *Web Application Development* documentation.

- General parameters: Parameters such as using screen locker, alternate row colors, prevent page refresh effect, and perform background check for host screen changes.
- Window parameters: Parameters such as enabling modal windows, the window position, and host key to send when window is closed.

- Emulation parameters: Parameters such as keyboard PF keys, type ahead, and arrow keys.

In addition, step-by-step instructions are available in the following sections:

- *Customizing the Background Check for Host Screen Changes*
- *Enabling the User to Control the Font Size*
- *Opening Multiple Web Sessions*
- *Printing a Capture of the Host Screen*
- *Enabling Sending Dup and FieldMark Characters to the Host*
- *Transferring Natural Data to/from the Host.*
- *Web Application Configuration Parameters > Macros.*
- *Transferring Files (FTP).*
- *Connection Pools* in the *Designing and Developing an Application* documentation.
- *Deploying with ApplinX* in the *Designing and Developing an Application* documentation: Deploy the application.
- *ApplinX Log Files* in the *Designing and Developing an Application* documentation: Refer to the following topics in this section:
  - *Server Log*
  - *Connection Pool Log (Windows only)*
  - *GCT Trace File*
  - *Framework Log*
  - *Monitoring the Performance*
  - *JavaScript Logger Engine*
  - *Printlet Log*
- *Printer Session* in the *Designing and Developing an Application* documentation: Refer also to the following topics:
  - *Printer Session Properties* in the *Reference Guide*.
  - *Printer Parameters* in the *Application Configuration Parameters* section of the *Reference Guide*.
  - *Printer>Offline* in the *Application Configuration Parameters* section of the *Reference Guide*.
  - *Printer Session API*

- ApplinX Printer Applet
- Printlet Servlet Redirector for ApplinX
- For further information, refer to:
  - *Self-Training and Movies* in the *Getting Started* documentation: Information on how to access the self-training material and accompanying movies.

## Introducing Pure HTML Emulation Development

ApplinX HTML emulation is an out of the box thin client HTML emulation, available in the .NET and J2EE environments. The screenshot below displays a host screen in ApplinX HTML emulation in the browser.

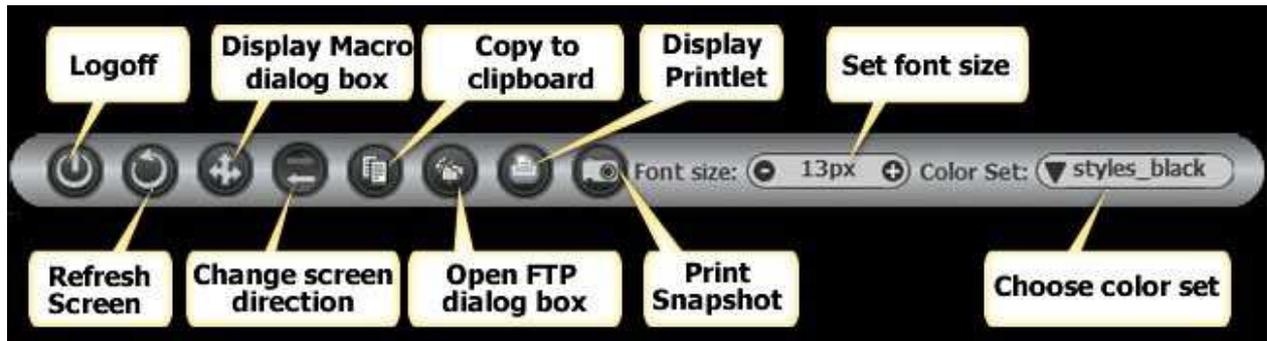
ApplinX is able to turn an existing terminal emulation instantly into a web browser terminal emulation, providing host key and print support and maintaining existing color schemes. This does not require any training for end users and can easily replace locally installed terminal emulation software with the thin-client web browser.

Refer to *Installing ApplinX Demo Applications* in the *Getting Started* documentation. Also refer to the self-training material included in ApplinX installation.



## The HTML Emulation Toolbar

The following lists the icons which appear on the toolbar, from left to right.



The toolbar provides common emulation capabilities such as:

- Logoff - logs off from the host application and closes the host session.
- Refresh screen - reloads the page.
- Display macro dialog box - records/plays macros.
- Change screen direction (displayed only in right-to-left applications) - changes the screen direction from right-to-left or from left-to-right.
- Copy to clipboard - enables users to copy text from the host screen by highlighting a rectangle on the screen and copying its content.
- Open FTP dialog box - used for transferring files to and from the host.
- Display printlet - the printlet acts as the host printer and allows the host to send print jobs directly to the user's printer.
- Print snapshot - displays the host screen in a pop-up browser window, for printing purposes.
- Set font size - enables users to increase/decrease the on-screen font size to their convenience.
- Choose color set - enables users to toggle between the defined style sheets.

## Web Enablement - Instant Web Application

The instant web application allows modernization of core system applications into modern looking web applications with (little or) no coding. Simple web configurations can improve the general "Look & Feel" of the application: designing a template with top and side images and a "company logo" (Inventory System), modifying the style sheet using different fonts, colors, etc. End users get the information quicker and with improved visibility.

### List of Subjects Relevant to the Instant Web Application

#### Note:

If not indicated otherwise, the sections that are referred to below can be found in the *Web Application Development* documentation.

- *Introducing Instant Web Application Development* (see below).
- *Instant Web Application Development Methodology* (see below).
- *Creating an Application* in the *Designing and Developing an Application* documentation: Create a new ApplinX application (including host and repository).
- *Creating an ApplinX Web Application* in the *Getting Started* documentation: Create an instant web application (JSP or .NET) using the Web Application Manager.
- *Creating a New Template*: Change the default web site template to achieve a different look and feel.
- *Customizing the Default Template*: Create a new template for new ApplinX web applications. This template will appear in the New Application wizard.
- *Configuring your Web Application*: Configure the framework parameters (for detailed descriptions of the parameters, refer to *Web Application Configuration Parameters*):
  - Session parameters: Configure session parameters such as server URL, application name, session ID, password, description, connection pool name, host user name, host password, and more.
  - Instant parameters: Control the display properties of instant pages. Such properties include: font size, font family, row spacing, color mode, host keys, render tags from right, render emulation behavior, enable displaying tables, enable displaying graphical window frames, and define render area.

Step-by-step instructions are available in the following sections:

- *Using a Proportional (Non-Fixed) Font in Web Pages*
- *Controlling Instant Display Properties*
- General parameters: Parameters such as using screen locker, alternate row colors, prevent page refresh effect, and perform background check for host screen changes.
- Window parameters: Parameters such as enabling modal windows, the window position and host key to send when window is closed.
- Emulation parameters: Parameters such as such as keyboard PF keys, type ahead, and arrow keys.

In addition, step-by-step instructions are available in the following sections:

- *Customizing the Background Check for Host Screen Changes*
- *Enabling the User to Control the Font Size*
- *Opening Multiple Web Sessions*
- *Printing a Capture of the Host Screen*

- *Enabling Sending Dup and FieldMark Characters to the Host*
- *Transformations* in the *Designing and Developing an Application* documentation: Use transformations to improve and manipulate your user interface.  
  
See also the following topics:
  - *Creating a New Code Transformation*
  - *Applying a New Transformation to all Screens*
  - *Applying a New Transformation to a Screen Group*
  - *Manipulating Individual Host Fields*
  - *Manipulating Host Characters*
  - *Manipulating Host Keys*
- *Transferring Natural Data to/from the Host.*
- *Web Application Configuration Parameters > Macros.*
- *Transferring Files (FTP)*
- *Connection Pools* in the *Designing and Developing an Application* documentation.
- *Deploying with ApplinX* in the *Designing and Developing an Application* documentation: Deploy the application.
- *ApplinX Log Files* in the *Designing and Developing an Application* documentation: Refer to the following topics in this section:
  - *Server Log*
  - *Connection Pool Log (Windows only)*
  - *GCT Trace File*
  - *Framework Log*
  - *Monitoring the Performance*
  - *JavaScript Logger Engine*
  - *Printlet Log*
- *Printer Session* in the *Designing and Developing an Application* documentation: Refer also to the following topics:
  - *Printer Session Properties* in the *Reference Guide*.
  - *Printer Parameters* in the *Application Configuration Parameters* section of the *Reference Guide*.

- *Printer>Offline* in the *Application Configuration Parameters* section of the *Reference Guide*.
- *Printer Session API*.
- *ApplinX Printer Applet*.
- *Printlet Servlet Redirector for ApplinX*.
- For further information, refer to:
  - *Installing ApplinX Demo Applications* in the *Getting Started* documentation.
  - *Self-Training and Movies* in the *Getting Started* documentation: Information on how to access the self-training material and accompanying movies.

## Introducing Instant Web Application Development

The instant web application allows modernization of core system applications into modern looking web applications with (little or) no coding. Simple web configurations can improve the general "Look & Feel" of the application: designing a template with top and side images and a "company logo" (Inventory System), modifying the style sheet using different fonts, colors, etc. End users get the information quicker and with improved visibility. Also new users can be trained more easily (web-like usage).

Using Screen Groups, the same design can be globally applied to multiple screens. Using Transformations, host patterns can be converted to web components. Such transformations include: formatting the screen's header area, formatting the message line, formatting titles and borders, converting input fields into GUI elements (combo boxes, radio buttons, check boxes), adding a calendar component to date input fields, formatting the host function keys into hyperlinks, buttons or images and removing unnecessary characters.

Refer to *Installing ApplinX Demo Applications* in the *Getting Started* documentation. Also refer to the self-training material included in ApplinX installation.



## Instant Driven Development

In this development mode, most host screens remain as Instant HTML pages, meaning they are generated on the fly by ApplinX Framework, without generating specific web pages (JSP or ASPX) for individual screens. Thus, the development process is quicker and simpler, and maintenance efforts are significantly reduced.

Customization of the instant pages is done using Instant Transformations and using Screen Groups. Often, there is not even a need to identify individual screens. Server transformations defined using the Transformation wizard in ApplinX Designer will be displayed in the instant pages. The page used in the kind of development is *GXInstantLogicContext* (JSP) / *GXInstantLogicWebForm* (.NET), which contains Instant Transformations registration.

## Instant Web Application Development Methodology

Following are some theoretical guidelines regarding development of ApplinX instant web applications, using Instant Transformations and Screen Groups.

- A Transformation (or Transform) is a code class (written in Java for the ApplinX Framework for JSP, and in C# or VB.NET for the ApplinX Framework for .NET), that is executed on all host screens or on a specific group of screens (i.e. an ApplinX Screen Group) and modifies the rendered instant HTML page for the current host screen it is executed on.

Transformations can be used to:

- Manipulate individual host fields (for example, display a title field in an underlined, blue font or display an image instead of a description field).
- Manipulate entire screens (for example, convert menu screens to a list of hyperlinks).
- Manipulate host characters (for example, remove all unnecessary dots (.) or dashes (-) from the screen).
- Using transformations: Transformations should be used to manipulate all screens in the host application or groups of screens (using Screen Groups). To manipulate the contents of specific, individual screens, the recommended methodology is to identify the screen(s), generate a web page for it (JSP or ASPX) and modify the generated page contents as necessary. Using a transformation for an individual screen is more complicated and unnatural, and therefore not recommended.
- Identifying screens: As a result, in order to use transformations it is not necessary to identify individual screens, but only screen groups (if needed), or not identify anything. This way, development effort is reduced and future maintenance becomes simpler.
- Mapping Fields to Screen Groups: The recommended methodology is to map all host fields that are common to several screens to a screen group that will be associated (implicitly or explicitly) with these screens. For example, an error message and title can be mapped to a screen group that will apply to all screens (or to specific screens), and should not be mapped repeatedly to individual screens. (Note that it is possible to override the mapping definition of the Screen Group in individual screens in which the field appears in a different position). If such mapping is not possible, refer to the fields according to their position or a unique identifier (such as preceding text).
- Manipulating fields: Once fields are mapped to a screen group, it is possible to write Transformations manipulating them.
- Manipulating entire screens: Once screen groups have been defined for the relevant screens, it is possible to write a transformation that is executed for these screen groups. For example, in order to convert a menu screen into a list of hyperlinks, create a Screen Group that will implicitly apply to all menu screens in the host application, and then create a transformation that will be executed for the Menu screen group and will do the required conversion. Note that it is not recommended to create transformations for individual screens, but rather generate web pages for them.
- Manipulate host characters: Manipulation of host characters usually applies to all screens in the host application, so the recommended methodology is to write a transformation that will manipulate the characters as desired and will be registered for all host screens. When not using the methodology above, it is also possible to create a screen group for the relevant screens and register the transformation only for the screen group.
- Absolute HTML rendering: The instant methodology renders the HTML tags in absolute terms. Rendering the HTML tags in absolute terms prevents indentation problems when trying to add or change elements or style sheets such as GUI elements, host windows and different fonts. In this version, each tag is rendered with corresponding top/left attributes which places the tag in the browser, in an exact position. The main advantage of this approach is that changing/adding tags/style sheets does not affect the neighboring tags of the modified tag. In addition, rendering in absolute terms saves the need to render empty spaces and only the relevant fields with content are rendered and not the entire screen. This significantly improves bandwidth performance since the HTML is much smaller.

- Instant rendering API: The instant component opens the instant solution as an API. It provides you the abilities to register as many transformations (which are standard code classes) as you require. These transformations can manipulate the HTML output by adding, removing, replacing and changing the screen tag model using a rich library of tags and an easy-to-use API.

## Web Enablement - Composite Web Application

As the composite web application is based on the instant web application, before reading this section, read the previous chapter: *Web Enablement - Instant Web Application*.

### List of Subjects Relevant to the Composite Web Application

**Note:**

If not indicated otherwise, the sections that are referred to below can be found in the *Web Application Development* documentation.

- *Creating an Application* in the *Designing and Developing an Application* documentation: Create a new ApplinX application (including host and repository).
- *Application Map* in the *Designing and Developing an Application* documentation: Enable application map recording.
- *Transferring Natural Data to/from the Host*.
- *Creating an ApplinX Web Application* in the *Getting Started* documentation: Create a new ApplinX web application (JSP or .NET) using Eclipse or Visual Studio's IDE.
- *Page Customization*.
- *Activating an Application Map from a Menu*.
- *Path Procedures* in the *Designing and Developing an Application* documentation: Use path procedures for host navigation.
- *Creating a Button / Hyperlink for Executing a Path Procedure*.
- *Collecting Data from Multiple Host Screens*.
- *Binding Procedure Outputs to an ApplinX Framework Based Web Page*.
- *Transferring Files (FTP)*.
- *Connection Pools* in the *Designing and Developing an Application* documentation.
- *Deploying with ApplinX* in the *Designing and Developing an Application* documentation: Deploy the application.
- *ApplinX Log Files* in the *Designing and Developing an Application* documentation: Refer to the following topics in this section:
  - *Server Log*

- *Connection Pool Log (Windows only)*
- *Process Tracing*
- *Logging a Message from within a Procedure*
- *Path Procedure Failure Log*
- *GCT Trace File*
- *Framework Log*
- *Monitoring the Performance*
- *JavaScript Logger Engine*
- *Printlet Log*
- *Printer Session* in the *Designing and Developing an Application* documentation: Refer also to the following topics:
  - *Printer Session Properties* in the *Reference Guide*.
  - *Printer Parameters* in the *Application Configuration Parameters* section of the *Reference Guide*.
  - *Printer>Offline* in the *Application Configuration Parameters* section of the *Reference Guide*.
  - *Printer Session API*
  - *ApplinX Printer Applet*
  - *Printlet Servlet Redirector for ApplinX*
- For further information, refer to:
  - *Installing ApplinX Demo Applications* in the *Getting Started* documentation.
  - *Self-Training and Movies* in the *Getting Started* documentation: Information on how to access the self-training material and accompanying movies.

## Introducing Composite Web Application Development

Using the composite web application it is possible to fully customize and extend your web applications using your Java or .NET development environment. The composite web application enables aggregating information from many core systems into one web page. It is possible to integrate legacy assets on various levels (screen, transaction, data), support various web environments making use of the ApplinX Base Objects (API) and customizing web frameworks.

The composite web application consists of generated web pages that are fully customizable. Upon sending an ApplinX server request such as a PF key, navigation path execution or application map navigation request, the web application attempts to load the page that has the same file name as the ApplinX session's current host screen. If none is found, the web application checks whether the current host screen belongs to a specific screen group and attempts to load a page that matches that screen group name.

Finally, if no match is found, an instant page will be loaded with a representation of the current host screen, allowing users to continue working.

It is recommended to carefully read the previous section (*Web Enablement - Instant Web Application*) as the composite web application is based on the instant web application.

Refer to *Installing ApplinX Demo Applications* in the *Getting Started* documentation. Also refer to the self-training material included in ApplinX installation.

