

ApplinX User Guide

ApplinX Solutions

Version 9.8

April 2015

This document applies to ApplinX Version 9.8.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2001-2015 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: APX-UG-SOLUTIONS-98-20150910

Table of Contents

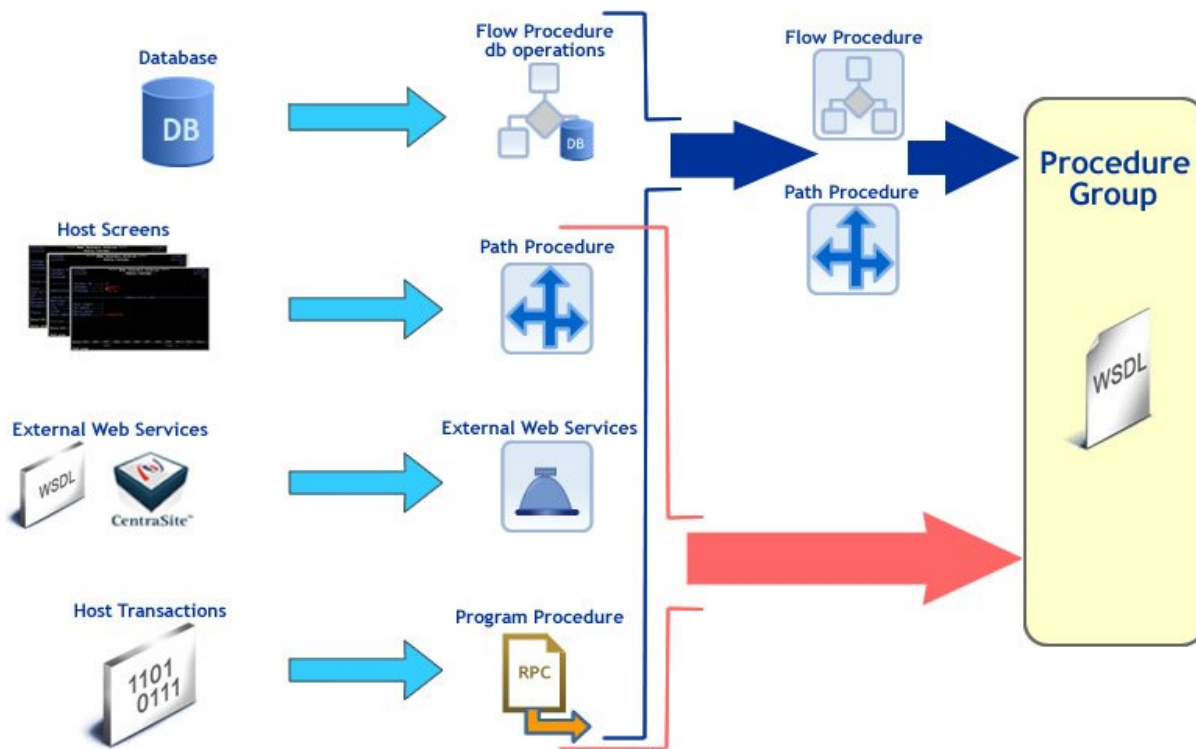
1 SOA Enablement	1
Implementing SOA Enablement	2
Working with Host Transactions (RPC)	4
Working with Databases	5
2 Web Enablement	7
Web Enablement - Pure HTML Emulation	8
Web Enablement - Instant Web Application	11
Web Enablement - Composite Web Application	16
3 Web Page Integration	19
4 Screen Process Extractor	23
Extracting Business Process Activities from Trace Files	24
Defining Configuration Parameters for Extracting Activities from Trace Files	26
PPM	27

1 SOA Enablement

■ Implementing SOA Enablement	2
■ Working with Host Transactions (RPC)	4
■ Working with Databases	5

SOA enablement allows reuse of functionality of the core systems, leveraging the operational qualities of the systems' transaction platform to new IT projects. This is accomplished by exposing the core systems' functionality and data as web services - either at the screen, transaction and database level - and integrating with any other environments that invoke web services. It also enables the core systems to become active equal peer participants in an SOA world.

Refer to *Installing ApplinX Demo Applications* in the *Getting Started* documentation. Also refer to the self-training material (see the Software AG Empower website at <https://empower.software-ag.com>).



Implementing SOA Enablement

SOA enablement, encapsulate complex processes that combines host screens to Path Procedures and exposes them for external use, using Web Services and Procedure Clients. Furthermore, it is possible to retrieve data from sources other than the host screens such as: host transactions (RPC), database queries and external web services. Flow procedures have the ability to perform database operations (select, execute, rollback and commit) to retrieve additional data from a database.

In addition, Path and Flow Procedures are capable of executing procedures of all types (other Path Procedures, Flow Procedures, Program Procedures and External Web Services). This enables the creation of a single web service which can retrieve data from several sources, simultaneously.

List of Subjects Relevant to the SOA Application



Note: If not indicated otherwise, the sections that are referred to below can be found in the *Designing and Developing an Application* documentation.

- *Creating an Application:* Define the host connectivity and the repository database/file location.
- *Creating Screens:* Define screens and map fields so that ApplinX server can recognize each screen and retrieve/send data from/to the host screen.
- *Path Procedures:* Create path procedures to navigate within the host application and to retrieve information from host screens.
- *Flow Procedures:* Encapsulate a complex process that can combine host screens and other data sources: databases, host transactions (RPC) and other web services.
- *Connection Pools:* Manage web service/procedure client consumption by creating connection pools and assigning them to procedures.
- *Connection Information Sets:* Provide necessary information to the connection pools to allow the proper creation of connections.
- *Assigning a Procedure to a Procedure Group:* Expose the procedures assigned to the procedure group as either web services or as procedure clients.
- *ApplinX as a Web Service Provider:* Monitor ApplinX server, sessions and connection pools.
- *Integration between ApplinX and WS-Stack* (also refer to the description of the WS-Stack server properties in the *Administration* documentation).
- *Registering a Web Service to CentraSite.*
- *Registering a Web Service to webMethods Integration Server.*
- *External Web Services:* Consume external web services to retrieve data from sources other than the host screens.
- *Deploying with ApplinX:* Deploy the application and/or ApplinX server.
- *ApplinX Log Files:* Refer to the following topics in this section:
 - *Server Log:* A log of the server's activities and problems.
 - *Connection Pool Log (Windows only):* A log used for fine-tuning connection pool parameters or identifying problems.
 - *Process Tracing:* A log used used to provide a log of performance times of processes such as procedures, paths and programs.
 - *Logging a Message from within a Procedure:* Uses the Log Message node (within Path or Flow Procedures).
 - *Path Procedure Failure Log:* Enables generating a log that includes debug data regarding procedures that fail in runtime.
 - *GCT Trace File:* Enables recording a file which traces the connection communication between the ApplinX server and the host.

- *Appendix J: Character Mode Hosts (VT Protocol) in the Reference Guide.*
- For further information, refer to:
 - *The SOA Demo* in the *Getting Started* documentation: This is a demo web application that implements web services created in ApplinX.
 - *Self-Training and Movies* in the *Getting Started* documentation: Information on how to access the self-training material and accompanying movies.

Working with Host Transactions (RPC)

When working with Program Procedures (RPC host transactions), ApplinX server implements RPC calls to execute host programs and routines, thus adding another type of interaction with hosts in addition to screens and screen data. Currently, COBOL and RPG programs are supported for iSeries hosts only.

List of Subjects Relevant to Working with Host Transactions



Note: If not indicated otherwise, the sections that are referred to below can be found in the *Designing and Developing an Application* documentation.

- *Creating an Application:* Define the host connectivity and the repository database/file location.
- Program Procedures (RPC):
 1. Enable using RPC Programs:
 - *Defining ApplinX RPC Application Parameters.*
 - *Application Configuration Parameters > Host Parameters > RPC* in the *Reference Guide*.
 - *Host Configuration Parameters > RPC* in the *Reference Guide*.
 2. *Program Procedures:* Import a Program Procedure.
 3. *Program Procedures* in the *Reference Guide:* Define program inputs and outputs.
 4. *Assigning a Procedure to a Procedure Group.*
 5. *Procedure Groups (used as Service Providers):* You can copy the WSDL (right-click on a Procedure Group and select **Copy WSDL**) or you can call the Program using a Procedure Client.
 6. *Procedures > Program Procedures* in the *Reference Guide:* Manage RPC connections (Administrative Web Services).
 7. For advanced configuration refer to *Procedure Groups (used as Service Providers)*.
- *Integration between ApplinX and WS-Stack* (also refer to the description of the WS-Stack server properties in the *Administration* documentation).
- *Registering a Web Service to CentraSite.*
- *Deploying with ApplinX:* Deploy the application.

- *ApplinX Log Files*: Refer to the following topics in this section:
 - *Server Log*: A log of the server's activities and problems.
 - *Connection Pool Log (Windows only)*: A log used for fine-tuning connection pool parameters or identifying problems.
 - *Process Tracing*: A log used used to provide a log of performance times of processes such as procedures, paths and programs.
 - *Logging a Message from within a Procedure*: Uses the Log Message node (within Path or Flow Procedures).

Working with Databases

Working with databases in SOA enablement means that ApplinX enables direct database access. Once the database connection is defined, it is possible to connect to it in any flow procedure and perform execute, select, commit and rollback statements.

List of Subjects Relevant to Working with Databases



Note: If not indicated otherwise, the sections that are referred to below can be found in the *Designing and Developing an Application* documentation.

- *Creating an Application*: Define the host connectivity and the repository database/file location.
- *Database Connection*.
- *Flow Procedures* (specifically database-related Flow Procedure nodes, refer to *Procedures > Flow Procedure Nodes* in the *Reference Guide*).
- *Assigning a Procedure to a Procedure Group*.
- *Integration between ApplinX and WS-Stack* (also refer to the description of the WS-Stack server properties in the *Administration* documentation).
- *Registering a Web Service to CentraSite*.
- *Deploying with ApplinX*: Deploy the application.
- *ApplinX Log Files*: Refer to the following topics in this section:
 - *Server Log*: A log of the server's activities and problems.
 - *Connection Pool Log (Windows only)*: A log used for fine-tuning connection pool parameters or identifying problems.
 - *Process Tracing*: A log used used to provide a log of performance times of processes such as procedures, paths and programs.
 - *Logging a Message from within a Procedure*, using the Log Message node (within Path or Flow Procedures.)

- For further information, refer to:
 - *The SOA Demo* in the *Getting Started* documentation: This is a demo web application that implements web services created in ApplinX.
 - *Self-Training and Movies* in the *Getting Started* documentation: Information on how to access the self-training material and accompanying movies.

2 Web Enablement

■ Web Enablement - Pure HTML Emulation	8
■ Web Enablement - Instant Web Application	11
■ Web Enablement - Composite Web Application	16

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.

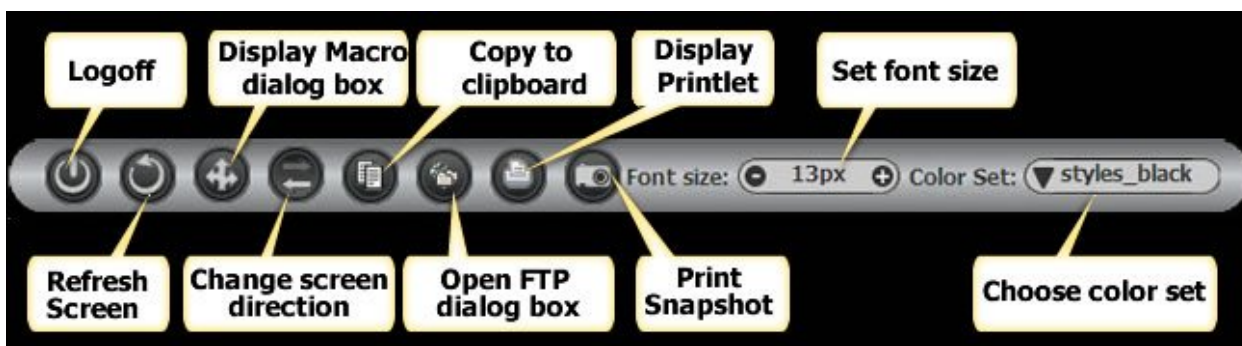
ApplinX is able to turn an existing terminal emulation instantly into a web browser terminal emulation, providing host key and print support and maintaining existing color schemes. This does not require any training for end users and can easily replace locally installed terminal emulation software with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

The diagram illustrates the transformation of a legacy mainframe screen into a modern web application interface. An orange arrow points from the legacy screen on the left to the modern web application on the right.

Legacy Screen (Left):

```

BCUSTEN0          ***** Demo Insurance Solution *****          BCUSTM01
27/07/08          - Browse Customers -          17:16:58
                                           1/2

A  Customer ID  Lastname      Firstname    Birthday   Sex    Type
-----
18  van de Wetering Marco    1931-02-12  M          I
2   van Bosch
22  von Gruenin
1   Bachmann
3   Bloggs
4   Busley
27  Chappell
6   Copperdale
5   Dillerburg
9   Dragos
101 Eisenstein
14  Grass
16  Higgins

Enter-PF1---PF2---PF3---PF4---
           Quit  Add
  
```

Modern Web Application (Right):

LEGACY INSURANCE COMPANY

27/07/08 Modify Proposal

Proposal ID 3

Main data (screen 1 of 4)

Status code	ACTIVE	Total Premium	\$6.91
Product code	HOUSE	External Calc. :	<input type="radio"/> Yes <input checked="" type="radio"/> No
Begin date	2003-11-02	Creator	HBU
Maturity date		Creation date	2004-04-01
Expiration date		Origin Code	AGENT
Policy holder ID	2		
Policy holder name:	van Bosch,Lea		
Agent ID	19		
Agent name	Michels,Maarten		

Enter changes

REFRESH LOGOFF TERMINAL KEY EXECUTE PATH

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.

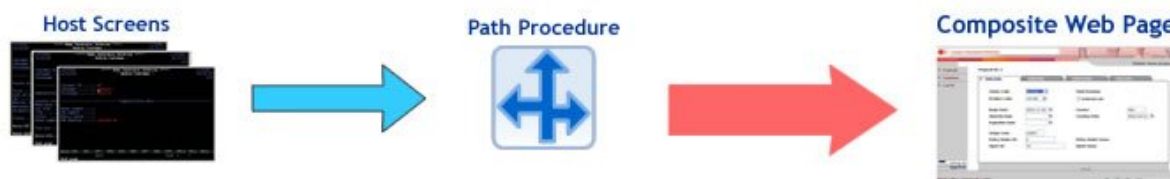
Introducing Composite Web Application Development

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.



3 Web Page Integration

The Web Page Integration feature enables simulating web browser activity and exposing it as a standard web service or integrating it with other ApplinX procedures. Web browser activity is simulated within ApplinX by recording and entering/capturing relevant web content using web browser based tools. The web content is used by the new type of procedure - the Web Procedure. This procedure is specifically designed to enable integrating the user interface (UI) of web pages within ApplinX, taking advantage of the flexible and dynamic capabilities incorporated within ApplinX procedure infrastructure. The Web Procedure can be exposed as a service in the same way the existing ApplinX procedures such as Path and Flow Procedures are exposed.



List of Subjects Relevant to Web Page Integration



Note: If not indicated otherwise, the sections that are referred to below can be found in the *Designing and Developing an Application* documentation.

- **Web Procedures:** When your network requires defining a proxy, set the proxy host name, port, user name and password in the `WebProcedureConfig` section in the `<ApplinX installation>/config/gxconfig.xml` file.

- *Creating an Application:* Create a new ApplinX application (including host and repository). Note that when creating a new application, you are required to enter a host. If you are only using the Web Page Integration solution, and are not integrating with other ApplinX components, this may seem to you unnecessary. Due to technical limitations, it is currently mandatory to define a host as part of the process of creating an application. As such, we recommend selecting a host from the list of predefined hosts.
- *Creating a Web Procedure.*
- *Running the Web Procedure.*
- *Editing the Web Procedure.*
- *Assigning a Procedure to a Procedure Group.*
- *Defining Procedure Inputs and Outputs.*
- *Working with Procedure Nodes.*
- *Using the Mapper to Map Source Elements to Target Elements.*

You may also want to read the following topics in the *Reference Guide*:

- *What is an XPath?*
- *Web Procedure Nodes*
- *General Nodes (Relevant for Flow, Path and Web Procedures)*
- *Web Procedure Specific Expressions*
- *General Expressions (relevant for Flow, Path and Web Procedures)*
- *Troubleshooting*

Limitations in Version 9.7

- Internet Explorer 7, 8 or 9 must be installed on the Designer machine in order to record Web Procedures.
- Recording dynamic pages (that use AJAX or heavily rely on scripting) might not always capture all events (such as hover or key strokes) or page parts loaded as results of such events.
- Keystrokes that are not used to fill contents in the page (such as ENTER or arrow keys), are not recorded. Therefore, it is recommended to explicitly click and select elements while recording a Web Procedure and avoid using such keys. For example, on a page where you can press the ENTER key to submit a form, you should use the mouse to explicitly click and select the Submit form button and not use the ENTER key.
- Blank frames that appear in the Selection mode may indicate the use of one of the following limitations:
 - Cross domain iframes: If the iframe displays a page from a different domain, its access is denied and its content cannot be used.
 - Non-HTML elements (such as Flash or Java applets).

- Pages that contain HTML 5 tags may not work properly.
- When selecting a list of elements, it is not possible to select a list including two elements from different iframes.

4

Screen Process Extractor

■ Extracting Business Process Activities from Trace Files	24
■ Defining Configuration Parameters for Extracting Activities from Trace Files	26
■ PPM	27

The Screen Process Extractor is used to analyze activities performed by the user and make business decisions based on this analysis.

Business Activity entities are created in ApplinX Designer and are used by the Screen Process Extractor. The Business Activity entity enables defining rules and conditions that are later searched for when analyzing trace files during the process. Once these rules and conditions are met, the relevant information regarding the activity is detailed in an XML file. The XML file is then used by ARIS PPM to analyze business processes.

A relevant license is required to implement this feature.

To complete the process extraction you need to:

1. Create a new ApplinX application (including host and repository). Refer to *Creating an Application* in the *Designing and Developing an Application* documentation.
2. Activate the Terminal Emulation Proxy (refer to *Application Configuration Parameters > Host Parameters > Terminal Emulation Proxy* in the *Reference Guide*) and configure trace file recordings (refer to *Recording Trace Files* in the *Designing and Developing an Application* documentation).
3. Change all the end users' terminal emulation host addresses to the ApplinX server address using the port which is configured in the Terminal Emulation Proxy Configuration.
4. Identify screens: define the screen and fields that are relevant for the business activities that you intend to track. You can import screen maps if you are using the supported map format. Refer to *Screens* in the *Designing and Developing an Application* documentation.
5. Define business activities: An activity is a set of rules and conditions which together create a business activity. Refer to *Business Activity Entity* in the *Designing and Developing an Application* documentation.
6. Define the configuration parameters that are to be used when analyzing trace files. Refer to [Defining Configuration Parameters for Extracting Activities from Trace Files](#) below.
7. Run an analysis on a trace file (either via a wizard in the Designer, typically used when developing, or via a batch file, typically used for the production environment). Refer to [Extracting Business Process Activities from Trace Files](#) below.
8. Analyze the outcome of the activity using ARIS PPM. Refer to [PPM](#) below.

Extracting Business Process Activities from Trace Files

ApplinX enables extracting business activities from trace files. To do this, ApplinX goes over the recorded sessions (in the form of trace files) and looks for instances of the defined business activities. Whenever an instance of an activity is identified, all relevant information required for PPM is written to an XML file. A summary of the information is written to a CSV file and a report of the process is displayed in the ApplinX console. Debugging information can be found in ApplinX server log.

Business activities can be extracted from the trace files either using a wizard or by running a batch file. The wizard can also be used to configure the desired parameters for extracting the business activities.

► **To extract activities from trace files using the wizard**

- 1 Right-click on the node of the relevant application and select **Extract Business Process Activities**. The wizard is displayed.
- 2 Select whether to base the extraction process on previously defined configuration parameters (refer to *Defining Configuration Parameters for the Extracting Business Process Activities analysis* below).
- 3 Select the trace file(s) or folder to be analyzed. Select whether once the file has been analyzed, the file name should indicate this by changing the extension to ".analyzed".
- 4 Select the output folder where the analysis PPM activities XML files are to be saved.
- 5 Select the log level of the analysis log (Debug, Errors, Info and Warnings). Notice that the location of the log file is displayed in the wizard.
- 6 Click **Next**. The **Selection of Business Activity Entities** screen is displayed.
- 7 Select whether to extract all business activities (default) or only selected activities.
- 8 Click **Next**. Determine whether to run the analysis immediately. Determine whether to save these settings in a configuration parameters file, and define its name and location.
- 9 Click **Finish**.

The extraction process will begin. A progress bar is displayed indicating the progress of the process. When clicking **Cancel**, the process will stop at the current point and no additional trace files will be analyzed. Once the process is completed or stopped, an XML file is created for each trace file, in the location you defined. This file contains all the events found in the analyzed trace file.

► **To extract activities from trace files using a batch file (using the command prompt window)**

- 1 Open a command prompt window.
- 2 Change the current directory to the <ApplinX home>/Utilities directory.
- 3 Type "extract_activities.bat" or "extract_activities.sh" followed by your required parameters. (The minus sign and letter should precede the value to distinguish between the parameters. The order of the parameters is not significant.)

Parameter	Description	Default
-s	Server address	127.0.0.1
-p	Server port	2323
-u	ApplinX user name (required parameter)	
-w	ApplinX user password	Empty by default
-a	ApplinX application name (required parameter)	
-x	The name of the XML file to use for the analysis process	Default.xml
-d	A semi-colon separated list of folders and gct files to analyze	The "TraceFilesLocations" specified in the XML file

Defining Configuration Parameters for Extracting Activities from Trace Files

Using the Extract Business Activities wizard, you can define configuration parameters that can be used for analyzing trace files. These parameters are saved to an XML file and can later be used when analyzing trace files either using the wizard or from batch files.



Note: It is possible to configure the parameters and immediately after, run the analysis by selecting the relevant check boxes in the last screen of the wizard.

▶ To define the configuration parameters for the activity extraction process

- 1 Right-click on the node of the relevant application and select **Extract Activities**. The wizard is displayed.
- 2 Select an existing configuration file which will provide the basic parameters for your configuration. Once you select a file the parameters defined in the selected configuration file are entered in the fields.
- 3 Select the trace file(s) or folder to be analyzed. Select whether once the file has been analyzed, the file name should indicate this by changing the extension to ".analyzed".
- 4 Select the output folder where the analysis PPM activities XML files are to be saved.
- 5 Select the log level of the analysis log (Debug, Errors, Info and Warnings). Notice that the location of the log file is displayed in the wizard.
- 6 Click **Next**. The **Selection of Business Activity Entities** screen is displayed.
- 7 Select whether to extract all business activities or only selected activities.
- 8 Click **Next**. Determine whether to run the analysis immediately. Determine whether to save these settings in a configuration parameters file, and define its name and location

Click **Finish**.

PPM

Analyze activities and processes in ARIS PPM: Ensure that the XML files that were created are located in the ARIS PPM installation directory. These files are loaded into PPM, where they can play a part in analyzing a broader business process. Refer to the ARIS PPM documentation for further details.

