

ApplinX User Guide

Getting Started

Version 9.9

October 2015

This document applies to ApplinX Version 9.9.

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-GETSTARTED-99-20151027

Table of Contents

1 Solutions and Architecture	1
SOA Enablement	3
Web Enablement	4
2 Hardware Requirements	7
Server	8
ApplinX Development Workstation	8
Network	8
3 Software Requirements	9
Operating Systems and Supported Platforms	10
ApplinX Repository	10
ApplinX for JSP Web Application	10
ApplinX .NET Web Application	11
Supported Browsers	11
4 License Key Concept	13
5 Installation	15
6 Upgrading, Migrating and Installing Fixes	17
Updating a Service Pack (from ApplinX Version 9.7 and above)	18
Migrating	19
Installing a Fix using Scripts	27
7 ApplinX Server	29
8 ApplinX Server Administrator	31
Starting the Server Administrator	32
Logging On	32
Main Screen	32
9 ApplinX Designer (Software AG Designer)	37
Starting the ApplinX Designer	38
Logging on	38
Adding a New Server Connection	39
The Designer Layout	39
10 Creating a New Application	45
11 Creating an ApplinX Web Application	51
Creating an ApplinX for JSP Web Application using the Software AG Designer (recommended)	52
Creating an ApplinX for JSP Web Application using JDK and Tomcat (no IDE)	60
Creating an ApplinX .NET Web Application using Visual Studio 2008	65
Creating an ApplinX .NET Web Application using Visual Studio 2010	65
Creating an ApplinX .NET Web Application using .NET (no IDE)	66
Upgrading an ApplinX .NET Web Application from Visual Studio 2003 to 2005	71
Upgrading an ApplinX .NET Web Application from Visual Studio 2003 to 2008	71
Upgrading an ApplinX .NET Web Application to Visual Studio 2010	72

- 12 Installing ApplinX Demo Applications 73
 - Installing the Demo Web Application using the Software AG Designer 74
 - Installing the Demo Web Application using Tomcat 74
 - Installing the Demo Web Application (Other Java Application Servers) 75
 - Installing the Demo Web Application using Visual Studio 76
- 13 Working with the Demo Applications 77
 - The Instant Demo 78
 - The Composite Demo 86
 - The SOA Demo 92
- 14 Self-Training and Movies 95

1 Solutions and Architecture

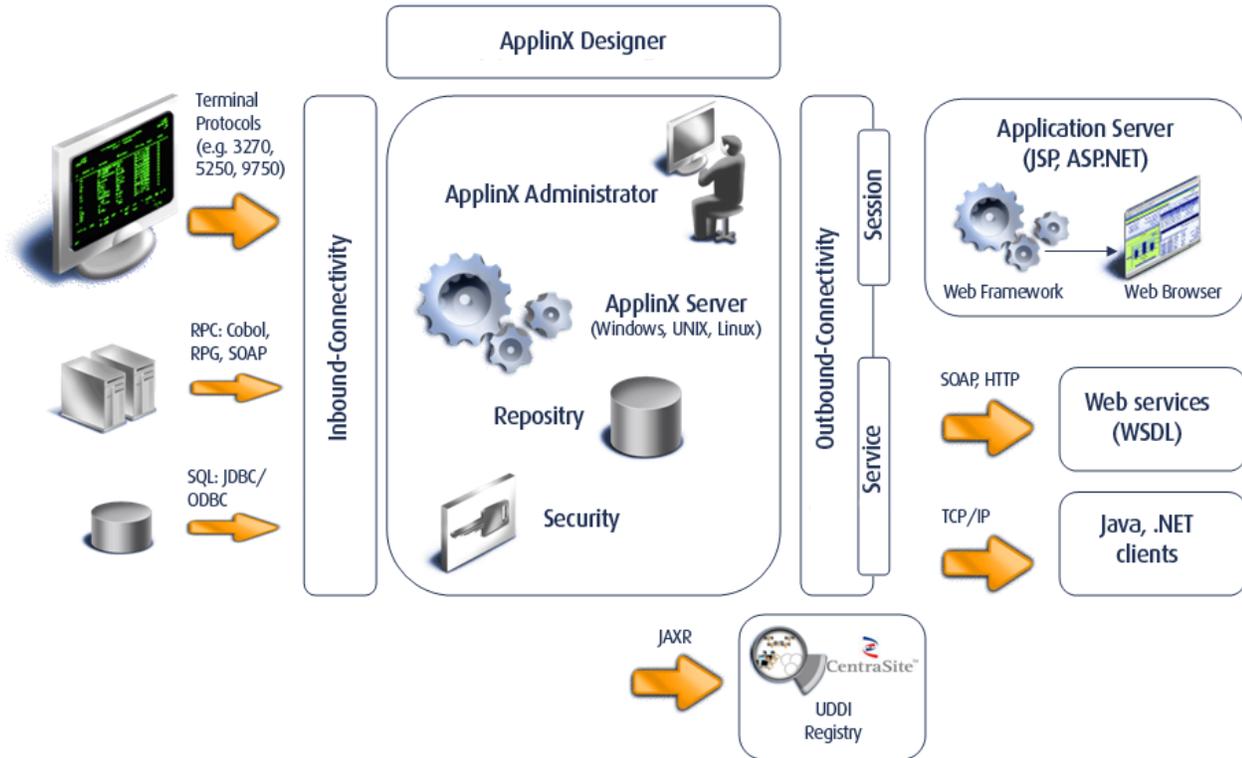
▪ SOA Enablement	3
▪ Web Enablement	4

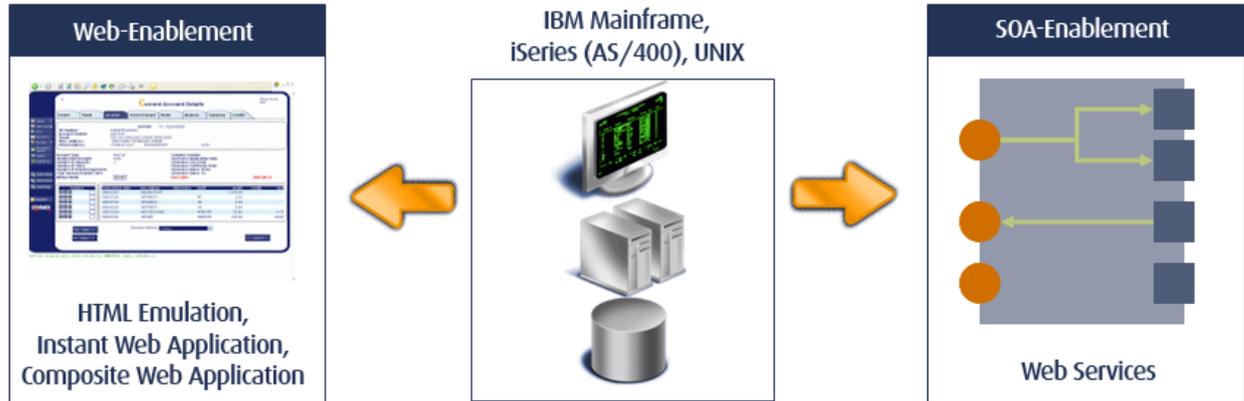
What is ApplinX?

ApplinX is a server-based technology that provides web-based applications an efficient, robust and easy way to access and integrate core system applications' data and transactions, without changing the core systems.

ApplinX exposes core systems and encapsulates them into standard programmatic components of the new selected environment (web services, .NET, J2EE, XML). These components can then be used as advanced building blocks in any modern platform - for new applications or existing ones (e.g. CRM applications). This way, core back-office systems can be integrated into new strategic IT platforms, and the workflow can be re-engineered to suit a more streamlined and efficient task-oriented and role-based approach.

webMethods ApplinX - Architecture



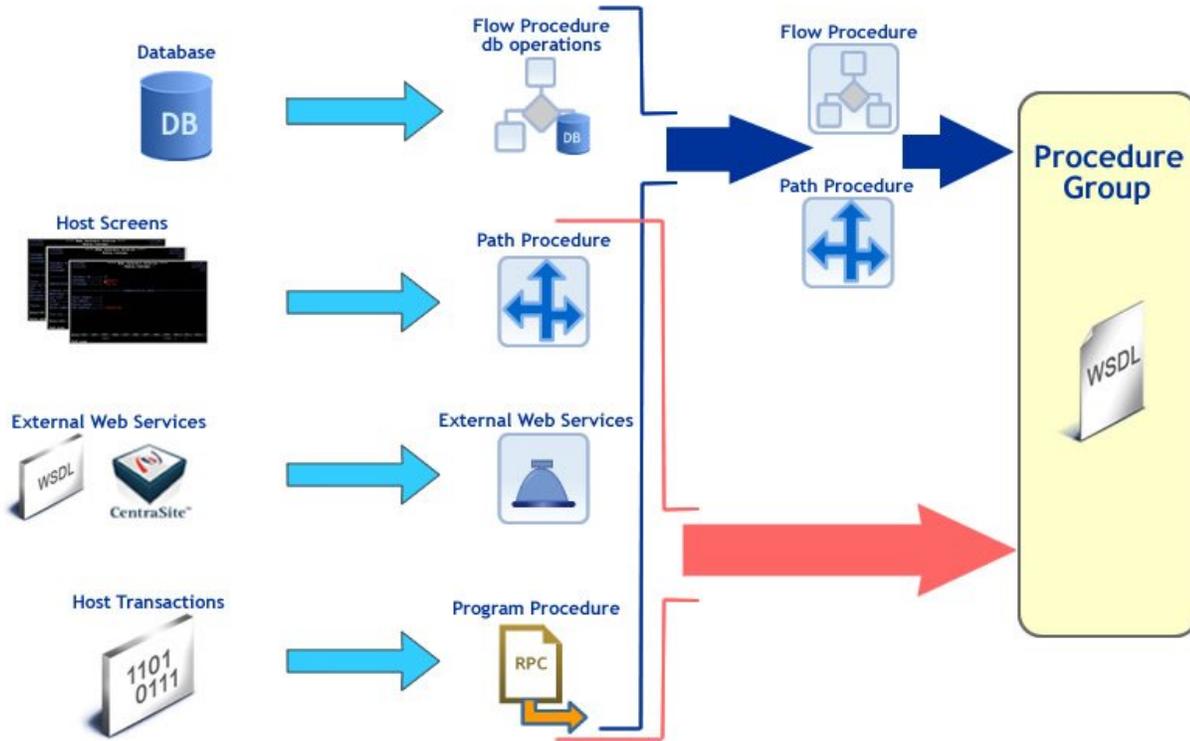


SOA Enablement

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 or transaction 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*. Also refer to the self-training material available on the Software AG Empower website.

Refer to *SOA Enablement* in the *ApplinX Solutions* documentation for further details as how to use this solution.



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. 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](#). Also refer to the self-training material available on the Software AG Empower website.

Refer to *Web Enablement - Pure HTML Emulation* in the *ApplinX Solutions* documentation for further details as how to use this solution.



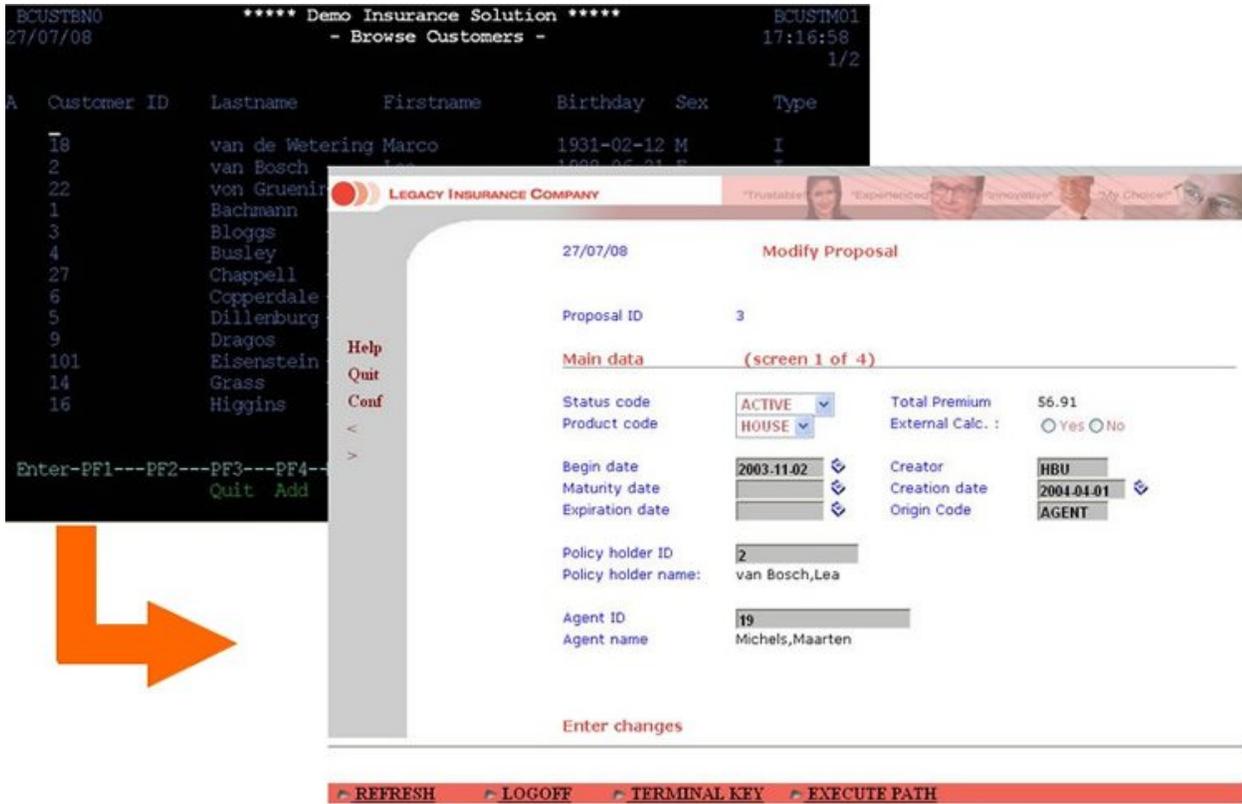
Instant Web Application

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. 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](#). Also refer to the self-training material available on the Software AG Empower website.

Refer to *Web Enablement - Instant Web Application* in the *ApplinX Solutions* documentation for further details as how to use this solution.

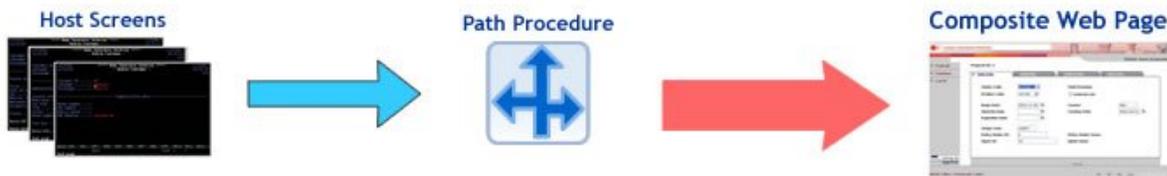


Composite Web Application

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.

Refer to *Installing ApplinX Demo Applications*. Also refer to the self-training material available on the Software AG Empower website.

Refer to *Web Enablement - Composite Web Application* in the *ApplinX Solutions* documentation for further details as how to use this solution.



2 Hardware Requirements

- Server 8
- ApplinX Development Workstation 8
- Network 8

Server

- 2 x 3 GHz Intel(R) Xenon(TM) processors (for 500 concurrent users).
- 2GB free disk space.
- RAM: ApplinX Server requires 200 MB. Each application requires additional minimum 40 MB (depending on the application repository size). Host sessions require 1MB per session.
- Network card.

ApplinX Development Workstation

- 2GB free disk space (server and development tools).
- RAM: minimum 4GB.
- Network card.

Network

ApplinX supports working with IPv6 as well as IPv4 when working with ApplinX components that use IP addresses.

3 Software Requirements

- Operating Systems and Supported Platforms 10
- ApplinX Repository 10
- ApplinX for JSP Web Application 10
- ApplinX .NET Web Application 11
- Supported Browsers 11

Operating Systems and Supported Platforms

For details regarding supported platforms, refer to the *Release Notes*. For the most updated information refer to the Software AG Empower website at <https://empower.softwareag.com>.

ApplinX server is Java-based and can run on a Java Virtual Machine 1.7. ApplinX can run independently as an independent Java process (this is the default Windows/UNIX installation), or as a J2EE application under one of the following application servers:

- Apache Tomcat 5.5/6.0/7.0
- IBM WebSphere 8.5
- WebLogic 10.3/11g
- JBoss 5.0 (Note that it is not possible to deploy an ApplinX server and a JSP web application in the same war file on JBoss 5.0.)

When running ApplinX as a J2EE application, the ApplinX server needs to be deployed with the Deployment Manager for J2EE (refer to *Deploying with ApplinX* in the *Designing and Developing an Application* documentation. It is possible to run ApplinX as a standalone server.

ApplinX Repository

ApplinX uses an internal database as a repository, based on the H2 database. External databases used as repository are no longer supported and will be migrated to the internal DB configuration. If you are using the JDBC/ODBC bridge, you will need to migrate manually. See [Migration of ApplinX Applications with External Database](#).

ApplinX for JSP Web Application

Developer's station:

- JDK 1.7
- Apache Tomcat 5.5/6.0/7.0
- JSP IDE - it is recommended to use Software AG Designer 9 (based on Eclipse 4.4)

Server:

- JDK 1.7
- One of the following application servers:

- Apache Tomcat 5.5/6.0/7.0
- IBM WebSphere 8.5
- WebLogic 10.3/11g
- JBoss 5.0

ApplinX .NET Web Application

Developer's station:

- Microsoft IIS 5.1/6.0/7.5/8.0
- Visual Studio 2008/2010
- Microsoft Visual J# .NET Version 2.0 Redistributable Package - Second edition (when using VS 2008/2010)

Server:

- Microsoft IIS 5.1/6.0/7.5/8.0
- Microsoft .NET Framework Version 2.0/3.0/3.5/4.0 Redistributable Package
- Microsoft Visual J# .NET Version 2.0 Redistributable Package - Second edition



Note: When creating an ApplinX .NET web application, view state is turned off by default (*web.config* file). If you want to use Net web controls in generated pages, ensure that you activate this option. Refer to the Microsoft ASP.NET documentation for further details.

Supported Browsers

- Microsoft Internet Explorer 7, 8, 9, 10, 11



Note: The web integration solution requires recording the browser activity in Internet Explorer 7, 8 or 9.

- Mozilla Firefox: latest version provided by vendor when this version of ApplinX was released
- Google Chrome: latest version provided by vendor when this version of ApplinX was released
- Safari: latest version provided by vendor when this version of ApplinX was released

4 License Key Concept



Note: During the installation process, the a check is performed that the license is of the same version as the installed ApplinX server. The license key format from version 9.6 and earlier is no longer supported. If you are logged on to Software AG's Empower site, you can request a new license key from Software AG Logistic Service Center (LSC).

About ApplinX Licenses

ApplinX can be purchased with one of the following types of licenses (it is possible to purchase more than one type of license):

- SOA license: Used for SOA enablement, including host transactions (RPC connections) and direct database access, and for web page integration solutions.
- Web enablement license: Used for HTML emulation, instant web application and for composite web application. Printer connections are also counted as web enablement connections.
- BPD license: Used for the Screen Process Extractor.

The ApplinX license is purchased for a number of users for each type of license. For example, you can purchase a license for 100 SOA users, and another license file for 50 web enablement users. When connecting to a session, the number of opened connections is compared with the number of users defined in the license file. The number of possible users must be equal to or greater than the number of opened connections. When trying to connect with an additional user, you will receive the following error message: `Cannot connect session. The server has reached the full capacity of concurrent users.`

License keys from version 9.6 and earlier are no longer supported. The following user messages appears when the server starts if in the license directory there is an old license: `License key format from version 9.6 and earlier is no longer supported. You can request a new license key from Software AG Logistic Service Center (LSC). STOPPING SERVER...`



Notes:

1. When working with a Connection Pool entity, connections will be related to a specific license type only once they are in use within a specific solution (SOA or web enablement).
2. A session used by a developer within the Designer is not counted as a connection.
3. Once the expiration date defined in a license file is reached, the connections count defined in the file is dropped from the accumulated connection count (per type).

Support of SOA and Web Features

During the installation process, only one of the licenses is provided. Copy the second to the ApplinX license directory manually. It will be loaded when the server starts.

Updating Your License

If you wish to change the license key after the installation, you need to locate the license file (*<Software AG Installation>\common\conf*) and replace the old license with the new one. Restart ApplinX server and the Designer.

5 Installation

ApplinX is installed with the webMethods Installer. We recommend you change the default password for the product as soon as possible after installation. Please refer to the webMethods Installer documentation, which you can find at <http://documentation.softwareag.com/>.

We recommend you always install the latest fixes available for your products. These are available from the Empower Product Support web site.

6 Upgrading, Migrating and Installing Fixes

- Updating a Service Pack (from ApplinX Version 9.7 and above) 18
- Migrating 19
- Installing a Fix using Scripts 27

Updating a Service Pack (from ApplinX Version 9.7 and above)

The upgrade process is similar to the installation process. Please refer to the webMethods Installer documentation, which you can find at <http://documentation.softwareag.com/>. When selecting the installation directory, ensure not to select the directory where ApplinX is currently installed (the default path is *C:\SoftwareAG*).

In the panel including the ApplinX specific parameters, browse and select the ApplinX license file.

Post-update Results

- When opening the Software AG Designer, the default workspace will point to a new location. It is possible to select and use the workspace location from the previous version, and in this way preserve your preferences (such as ApplinX server connection list and ApplinX (Eclipse) preferences (**Windows > Preferences > Software AG > ApplinX**)) from the previous version. If you use the new workspace location, check that the preferences suit your needs.
- From version 8.2, the batch and shell commands used to start and stop the server, and to start the Administrator are located in the *<ApplinX installation>/bin* folder.

Post-update Tasks

- When using ApplinX web applications (JSP or .NET): Upgrade the web application using the ApplinX Web Application Manager in the Designer (refer to *Framework Management* in the *Web Application Development* documentation of ApplinX).
- You may have made changes in your previous ApplinX version, in one of the following files. Backup copies of these files can be found (unless otherwise stated) in the ApplinX installation directory with the *.bak* suffix. To apply these changes to your current ApplinX version, you must copy the changes from within the backup files to the relevant file.
 - *GXApplinXService.ini*
 - *bin/startup.bat*
 - *bin/startup.sh*
 - *bin/gxadmin.bat*
 - *bin/gxadmin.sh*
 - *config/gxadmin.prp*
 - *config/gxstartup.prp*

Migrating

ApplinX applications from previous ApplinX versions are able to run on the current ApplinX version once you have performed a number of migration activities. The migration process has been tested for supported versions. Migration can be performed when installing the new ApplinX server version or via batch files after the new server has been installed.

The following topics are covered below:

- [Pre-Migration Steps and Prerequisites](#)
- [Migration](#)
- [Migration Results](#)
- [Post-migration Tasks](#)
- [Post-migration Tasks for Entities Created in Versions before 8.0](#)
- [Post-migration Web Application Tasks](#)

Pre-Migration Steps and Prerequisites

- Back up the repository database.
- When using ApplinX .NET 1.1 web applications based on Visual Studio 2003, you will require installing Visual Studio 2008/2010.NET 2.0-4.0 as .NET framework 1.1 is no longer supported.

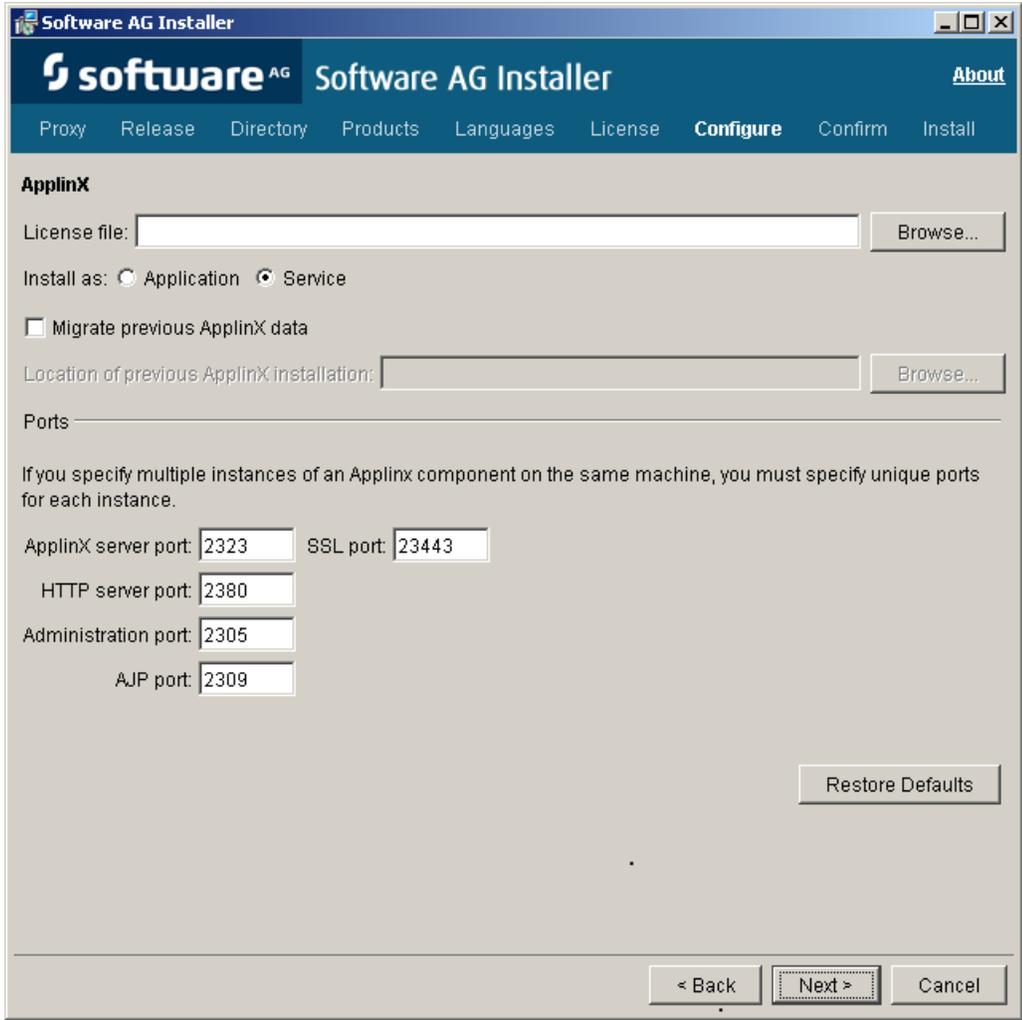
Migration

Migration can be carried out as follows:

- [Migration of all ApplinX applications when installing ApplinX](#)
- [Migration of all ApplinX applications after installing ApplinX](#)
- [Migration of a specific ApplinX application](#)
- [Migration of ApplinX applications with external database](#)

Migration of all ApplinX applications when installing ApplinX

To migrate ApplinX applications during the installation process: In the screen where you select the ApplinX parameters, select the **Migrate previous ApplinX data** check box and specify the existing installation location.



Migration of all ApplinX applications after installing ApplinX

To migrate all ApplinX applications after the installation has been completed: Run the `<ApplinX root>\utilities\migrate_ApplinX_server.bat/sh` file, enter the installation path of your previous installation, and then enter the new path of the current installation.

For example: `migrate_ApplinX_server.bat "c:\ApplinX52" "c:\SoftwareAG\ApplinX"`

Migration of a specific ApplinX application

To migrate a single ApplinX application from versions previous to 8.0:

1. Export the relevant GXZ file of the application (the GXZ file in older versions (5 and below) includes both the application and the entities).
2. Convert the GXZ file to a GXAR file (using `<ApplinX root>/utilities/convert_gxz_to_gxar.bat`). Refer to *Convert Utility Batch File* in the *Administration* documentation.
3. Import the GXAR file into ApplinX 8.x. Refer to *Importing an Application's Configuration or Entities* in the *Designing and Developing an Application* documentation.

To migrate a single ApplinX application from version 8.x and above:

1. Export the relevant GXAR file of the application (the GXAR file includes both the application and entities).
2. Import the GXAR to the new version. Refer to *Importing an Application's Configuration or Entities* in the *Designing and Developing an Application* documentation.

Migration of ApplinX applications with external database

For ApplinX applications that use an external database (except JDBC-ODBC), the database type will be changed automatically to internal database. A message warns that there are applications containing a JDBC-ODBC database type that will not be migrated. You have two options:

1. Choose **OK** to continue with the installation. ApplinX will migrate all applications except for the JDBC-ODBC types. The names of these applications will be written to a log.
2. Choose **Cancel** to stop the installation, then modify those applications.

➤ To modify JDBC-ODBC applications

- Change the repository type to internal DB /gxz in the original ApplinX installation installed on your computer, and then restart the installation and migration.

Migration Results

The following is the outcome of the migration process and relates to migrating all ApplinX applications:

- The entire old *config* folder (except for the *http.xml* file) is copied to the new installation location. The *gxstartup.prp* file from this folder is renamed to *.bak*, and the new *gxstartup.prp* file (from the new installation) is used.
- The new *config* folder (from the new installation) is renamed to *config.bak*.
- The entire *host-applications* folder is copied except for the demo applications. The demo applications from the new installation are used.

- All database related jar/zip (*classes*.zip, classes*.jar, *sql*, *db2**) are copied to the *lib* folder of the new installation. Note: any other DB driver should be copied manually to `\ApplinX\applinx-web\WEB-INF\lib`.
- The batch and shell commands used to start and stop the server, and to start the Administrator are no longer located in the ApplinX installation folder. These files are now located in `<ApplinX installation>/bin` folder and are named *startup.bat* and *shutdown.bat* accordingly for the server, and *gxadmin* for the Administrator.

The *start-gxserver.bat/sh* and *gxadmin.bat/sh* files from the previous ApplinX installation are copied to `<ApplinX Installation>\bin` and the *gxstartup.prp* and *GXApplinXService.ini* files are copied to `<ApplinX Installation>`. These files are renamed to **.bak*.



Notes:

1. Any manual changes made in the previous ApplinX versions (such as language, file encoding, database jar etc.) must be backed up and copied to the new version.
2. When migrating from APX 8.0, *conf* folder (internal Tomcat) is not copied from the old installation. Therefore any manual change made in this folder must be re-done again in the new *conf* folder.

Relevant for migrating from ApplinX versions before 8.0:

The following actions occur when the server is started for the first time after the migration process has been completed:

- Each application (configured in *config/gxconfig.xml*), will be created as a separate file and copied to *host-application/<application name>/gxconfig.xml*. The original file *config/gxconfig.xml* will hold from now on only ApplinX server configuration and host definitions which are not in use by any application.
- Replay files are copied to *host-applications/<application name>/records* and new recorded files are saved in this location.
- If a file based repository (GXZ or MDB) is configured for the application, the repository is copied to *host-applications/<application name>/db*.

A post-migration report is written to the `<ApplinX root>/log/gxlog.txt`.



Note: If the application name includes space characters, these are replaced by underscores, as space characters are not supported in versions from 8.0 and above. However, previous SOAP clients will be able to connect to old addresses which include spaces.

Post-migration Tasks

- When opening ApplinX Designer, enter the name of a workspace directory. When upgrading from ApplinX version 8.0 and above, define a new workspace directory in order to be able to work with both versions.

Post-migration Tasks for Entities Created in Versions before 8.0



Note: The following entities are still supported in runtime, but cannot be edited or changed, it is therefore strongly recommended to follow the guidelines below

- [Old Paths and Path Wrappers](#)
- [Path-based Tables](#)
- [Map Paths](#)
- [Namespaces](#)
- [Procedure Groups](#)
- [Tables](#)
- [Procedure Clients](#)

Old Paths and Path Wrappers

Paths and Path Wrappers require a conversion process.



Note: When converting a Path Wrapper, the path which is invoked by the path wrapper is also converted. When converting a Path which is invoked by a Path Wrapper, the Path Wrapper is also converted.

> To convert

- 1 Right-click on the Path or Path Wrapper and select **Convert**. The Conversion wizard is displayed. The first screen displays the name of the Path and Path Wrapper that are to be converted.
- 2 Click **Next**.
- 3 **Warning Messages:** The conversion process may detect that the Path/Path Wrapper which you are attempting to convert includes steps which need to be manually tweaked, after the conversion has been completed, or may include attributes or settings which cannot be converted. Read the warnings carefully to understand whether you are required to make changes after completing the conversion.
 - **Path warning messages**
Warning messages appear when a path detects one of the following issues. Manual changes may be required for items that are not supported:
 - Path timeout (the conversion process does not support this).

- Exception paths are not handled in the conversion process. Now, there can be only one Exception path per application, defined in the application properties.
- "*"lastup" field is not supported.
- <any unexpected screen> is not supported.
- Input values in "any screen" are not supported. It is recommended to define a Screen Group for these fields instead of mapping to ANY screen.
- Inputs are not mapped when the input field is not found in either the source or target screen.

- **Path Wrapper warning messages**

The conversion process does not support collecting attribute data from within a specific instance of a screen. To preserve this setting, you need to manually define to collect data from this screen: when you know the specific instance, preserve the links to outputs in the relevant step and remove links to outputs within irrelevant steps.
- Retrieving field colors using the Path Wrapper returned colors such as LightBlue and LightYellow in a single word. After the conversion these colors are returned as two words, separated with a space (i.e. Light Blue, Light Yellow).
- It is only possible to convert a Path Wrapper that invokes an old Path. Path Wrappers which invoke Path Procedures cannot be automatically converted.
- The **Always collect last screen** option, if selected, is ignored in the conversion process.

Click **Next**.

- 4 When ApplinX detects that an entity with the name of the entity that you are trying to convert, already exists, you are required to enter a different name. Click **Next**.
- 5 The **Summary** screen is displayed. Here you can see the entities that are to be converted. Select whether to open these entities in the Editor once the conversion process is completed.

Path-based Tables

Old Path based tables are still supported in runtime, but cannot be edited or changed. It is recommended to convert the path used by the table.

1. Right-click on the path and select **Convert**.
2. The table must be represented as a Data Structure array in the path procedure output. In order to do this, one must first define a Data Structure type (entity) that will represent a single row in the table. Follow these steps to do this:
 - a. In the relevant ApplinX screen, add a Table definition of the tabular data. This allows you to easily create a Data Structure entity and map data to the procedure output.
 - b. In the relevant step, right-click the screen table and select **Create Data Structure**.
 - c. Select the Path Procedure root node and then define an output as an array of the newly created structure type.

- d. In the mapper, drag the screen table onto the data structure array output. Ensure that all of the table columns are properly mapped to the arrays attributes.
- e. Create a Procedure Group and assign the path procedure to it.
- f. Generate a Procedure Client into the web project (refer to *Generating a Procedure Client* in the *Designing and Developing an Application* documentation).
- g. In the relevant web page, replace the `gx_fillTable` code with code that executes and binds the procedure result to the HTML table (refer to *Binding Procedure Outputs to an ApplinX Framework Based Web Page* in the *Web Application Development* documentation).

Examples for collecting tables can be found in *CompositeDemo* (*BrowseCustomers1.jsp/asp*) or *SOADemo* (*CustomerList.jsp/asp*).

Map Paths

All map paths are automatically merged into a single Application Map during the repository synchronization (refer to *Application Map* in the *Designing and Developing an Application* documentation). The Map paths are removed and each step in them is assigned to its relevant screen. These steps appear in the screen editor, in the **Map Steps** tab.



Note: Advanced usage in the old Map path flow tab or dynamic variables is currently not supported in the new Application Map. Refer to *Application Map* in the *Designing and Developing an Application* documentation for further details.

Namespaces

The Namespace entity is deprecated. It is not possible to create new Namespaces, however, in procedure groups you can still specify a Namespace. A migrated application that had Data Structures (Business entity in 5.2 and lower) that were associated with namespaces still refers to them, but this reference is not displayed and is not editable. Procedure groups display the namespace name and allow you to edit the name.

Procedure Groups

If you have Procedure Groups in the repository, their SOAP clients are redirected to WS-Stack. Set each procedure group to automatically be deployed to WS-Stack (in the Procedure Group editor).



Note: For performance improvements, it is recommended to regenerate all your SOAP clients. For automatic deployment of the web services to WS-Stack, the application initialization mode should be set to "Automatic".

Tables

Sorting functionality in tables is no longer provided.

Procedure Clients

When working with an application which included space characters in the application name, you are required to manually change the application name in each of the Procedures Clients, replacing the space character with an underscore.

Post-migration Web Application Tasks

JSP:

1. Ensure that you have:
 - JDK 1.7
 - A supported web server (refer to [ApplinX for JSP Web Application](#) in the section *Software Requirements*).
2. It is recommended to import/copy the web applications from older Eclipse versions to Software AG Designer 9.9 (based on Eclipse 4.4 technology), as this is the common Software AG development tool, used with other Software AG products such as ApplinX, EntireX, CentraSite and Integration Server.
3. Upgrade the web application using the ApplinX Web Application Manager in the Designer (refer to *Framework Management* in the *Web Application Development* documentation).
4. Compile the JSP project:
 - With IDE (from Eclipse or any other IDE)
 - Without IDE (*make.bat*)



Note: When working with an application which included space characters in the application name, you are required to manually change the application name in the web application (e.g. in *GXBasicWebForm.jsp* file, or in the Framework Configuration Editor, in the **Application name** parameter).

.NET:

1. Visual Studio 2003 and .NET 1.1 Users: Visual Studio 2003/.NET 1.1 applications should be migrated to Visual Studio 2008/2010.NET 2.0-4.0 using Microsoft's Visual Studio automatic converter.
2. It is recommended to convert from .NET project type "Web Project" to "Web Site" as from the current version ApplinX generates .NET web pages based on the "Web Site" format and file structure.

3. Upgrade the web application using the ApplinX Web Application Manager in the Designer (refer to *Framework Management* in the *Web Application Development* documentation).
 4. Compile (relevant for .NET "Web Projects" type projects):
 - With IDE (from Visual studio)
 - Without IDE (*make.bat*)
-  **Note:** When working with an application which included space characters in the application name, you are required to manually change the application name in the web application (e.g. in *GXBasicWebForm.cs/GXBasicWebForm.vb* file, or in the Framework Configuration Editor, in the **Application name** parameter).

Installing a Fix using Scripts

Fixes are available for UNIX and Window on the Software AG Empower website. Follow the instructions below when downloading the fix from Empower.

Note that when applying a fix, you must make sure that the installed ApplinX version is of the same service pack as the fix version, i.e. that the first three digits are the same (e.g. 9.5.1).

» To install a fix

- 1 The fix will install and add all the necessary new files to your existing ApplinX installation.
- 2 Before starting the fix installation, close ApplinX and Eclipse and stop the ApplinX service.
- 3 To start the fix installation process, extract the fix zip file to the existing ApplinX directory: For example: *C:\SoftwareAG\ApplinX*. A new directory is created: *\hotfix-<hot fix version>*.
- 4 To install the fix, just run *install-hotfix.bat* on Windows platforms, or *install-hotfix.sh* on UNIX platforms. On Windows platforms: Check that the `ECLIPSE_PATH` parameter is pointing to the desired Eclipse installation location and that you have removed the `rem` remark indication.
- 5 During the fix installation, a backup folder will be created containing the original files from before the update (*backup-for-hotfix-<hot fix version>*).
- 6 After the fix installation, a log is created with a list of all the copied files (*install.hotfix.log*).
- 7 Upgrade the web application using the ApplinX Web Application Manager in the Designer (refer to *Framework Management* in the *Web Application Development* documentation).

 **Note:** It is possible to uninstall the fix. This will revert all updated files to their original state. To uninstall the fix, just run *uninstall-hotfix.bat* on Windows platforms, or *uninstall-hotfix.sh* on UNIX platforms. After the fix uninstallation, a log is created with a list of all copied files (*uninstall.hotfix.log*).

7 ApplinX Server

The ApplinX server can be started either as a Windows service or using a batch file. ApplinX can also run under a Java Application Server (for further details, refer to the relevant topics in *Deploying with ApplinX* which is included in the *Designing and Developing an Application* documentation).

Windows Service:

In the installation process, it is possible to define that ApplinX server runs as a Windows system service, started and stopped via **Start > Settings > Control Panel > Administrative Tools > Services**. By default, the service is automatically run when starting the computer.

Batch Files:

ApplinX Server can be run using batch command files, located in the *<ApplinX installation>/bin* folder:

<i>startup.bat</i>	Starts the ApplinX server.
<i>shutdown.bat</i>	Stops the ApplinX server.

UNIX Environments:

In the ApplinX installation directory, you can find the following shell scripts:

<i>startup.sh</i>	Starts the ApplinX server.
<i>shutdown.sh</i>	Stops the ApplinX server.

8 ApplinX Server Administrator

- Starting the Server Administrator 32
- Logging On 32
- Main Screen 32

Starting the Server Administrator

From the **Start** menu, select **Software AG > Administration > ApplinX Administration <version number>**.

Alternatively, you can run the Administrator from a remote PC (for example: on a Microsoft Windows operating system) and connect to the server using TCP/IP.

Logging On

Once you start the ApplinX Administrator, the **Logon** screen is displayed. When this is the first time you are logging on, by default, "Administrator" will be displayed in the **User name** field. Type the server address (IPv4 and IPv6 address formats are supported) and port. When it is not the first time you are logging on, the Administrator will automatically try to connect to the last server it connected to.

Main Screen

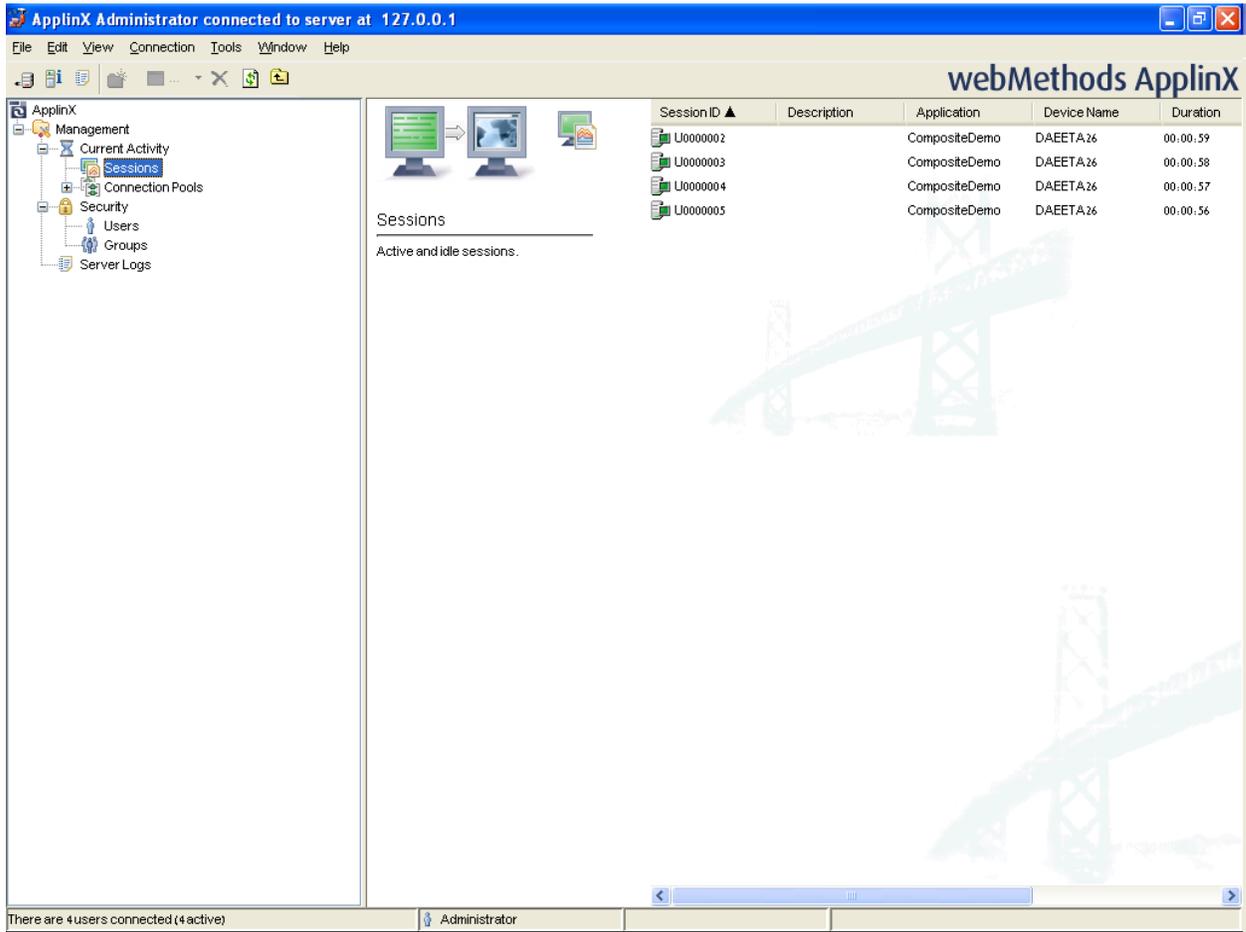
The screen consists of four parts:

- Title bar, menu bar and toolbar.
- ApplinX Explorer pane.
- Main view: The main view corresponds to the selected node/entity/session in the ApplinX Explorer.
- Status bar: The status bar displays information according to the object selected in the main view. The status bar may include information such as: the user name, application name, the number of users connected, indication of file tracing/replaying or the host address.

Operations, features and applications that the logged-in user is not authorized to use will be disabled or are invisible.

- [ApplinX Explorer](#)
- [Toolbar](#)

- Menu Bar



ApplinX Explorer

Enables viewing and controlling users' activity, maintaining the server's configuration, viewing runtime information about connection pools, sessions, server logs, define hosts and design and configure applications.

ApplinX	The root node.
Management	Includes managerial functions.
Current Activity	Controls the sessions connected to the server, the server activity and the connection pools. The Current Activity node is unavailable when the ApplinX Administrator is not connected to the server.
Sessions	Lists the currently connected sessions.
Connection Pools	Provides runtime information regarding the connection pools.
Security	Holds the definitions of the users, their permissions and roles, and the user groups.
Server Logs	Lists the server logs.

Toolbar

Icon	Description
	Disconnect from Server
	Connect to Server
	Server Information...
	Server Log
	New Folder
	New User
	New Group
	Delete
	Refresh
	Up One Level

Menu Bar

File Menu

Menu Item	Description
New	Creates a new user or group.
Open	Opens the selected item.
Delete	Deletes the selected item.
Properties	Opens the properties of the selected item.
Exit	Exits the Administrator.

Edit Menu

Menu Item	Description
Select All	Selects all entities displayed in the main view.
Invert Selection	Selects all entities that were not selected, and deselects entities that were selected.

View Menu

Menu Item	Description
ApplinX Explorer	Hides/shows the ApplinX Explorer tree area of the screen.
Summary	Hides/shows the summary area of the screen.
Server Log	Displays the server log.
Refresh	Refreshes the main view to match the data stored on the server.

Connection Menu

Menu Item	Description
Connect/Disconnect	Connects and disconnects the Administrator from the server.
Cancel Session	Displays the View Session window of the selected session.

Tools Menu

Menu Item	Description
Server Info	Displays the Server Information dialog box.
User Information	Displays the User Information dialog box.
Restart Log	Restarts the log.

9 ApplinX Designer (Software AG Designer)

▪ Starting the ApplinX Designer	38
▪ Logging on	38
▪ Adding a New Server Connection	39
▪ The Designer Layout	39

Starting the ApplinX Designer

From the **Start** menu, select **All Programs > Software AG > Tools > Software AG Designer <version number>**.

Alternatively, you can run the Designer from a remote PC (for example: on a Microsoft Windows operating system) and connect to the server using TCP/IP. When running the Designer from a remote PC, some Designer features may not work, for example opening the file system.

The first time you start the Designer, a Welcome page will be displayed. From this page you can access the ApplinX perspective and the help contents. If you close the Welcome page, and you would like to display the ApplinX perspective, go to **Window > Open Perspective > Other** and select **ApplinX**.

If you are not yet familiar with Eclipse, see the Eclipse online help at <http://www.eclipse.org/documentation/>, or start the Software AG Designer and then choose **Help > Help Contents**. General information on Eclipse can then be found under *Workbench User Guide*. When working with the Software AG Designer, the online help also provides help for the currently installed Software AG products; this can be found under *Software AG Designer Guides*.

Logging on

Once the Designer is displayed, you will see the available list of servers. Double-click on a server or select **Connect** from the right-click context menu. When this is the first time you are logging on, by default, "Administrator" will be displayed in the **User name** field. When it is not the first time you are logging on, ApplinX Designer will automatically try to connect to the last server it connected to.

You can select to remember the user password in the **Login** dialog box. Once this is selected, the **Login** dialog box is not shown again. To require entering a password and displaying the **Login** dialog box, change the **Remember passwords** check box in **Window > Preferences > Software AG > ApplinX**.



Note: The first time the **Remember password** check box is selected, an Eclipse Secure Storage message is displayed. This message offers remembering an internal Eclipse password using a hint, but this is not applicable to the ApplinX server password, therefore click **No**.

Adding a New Server Connection

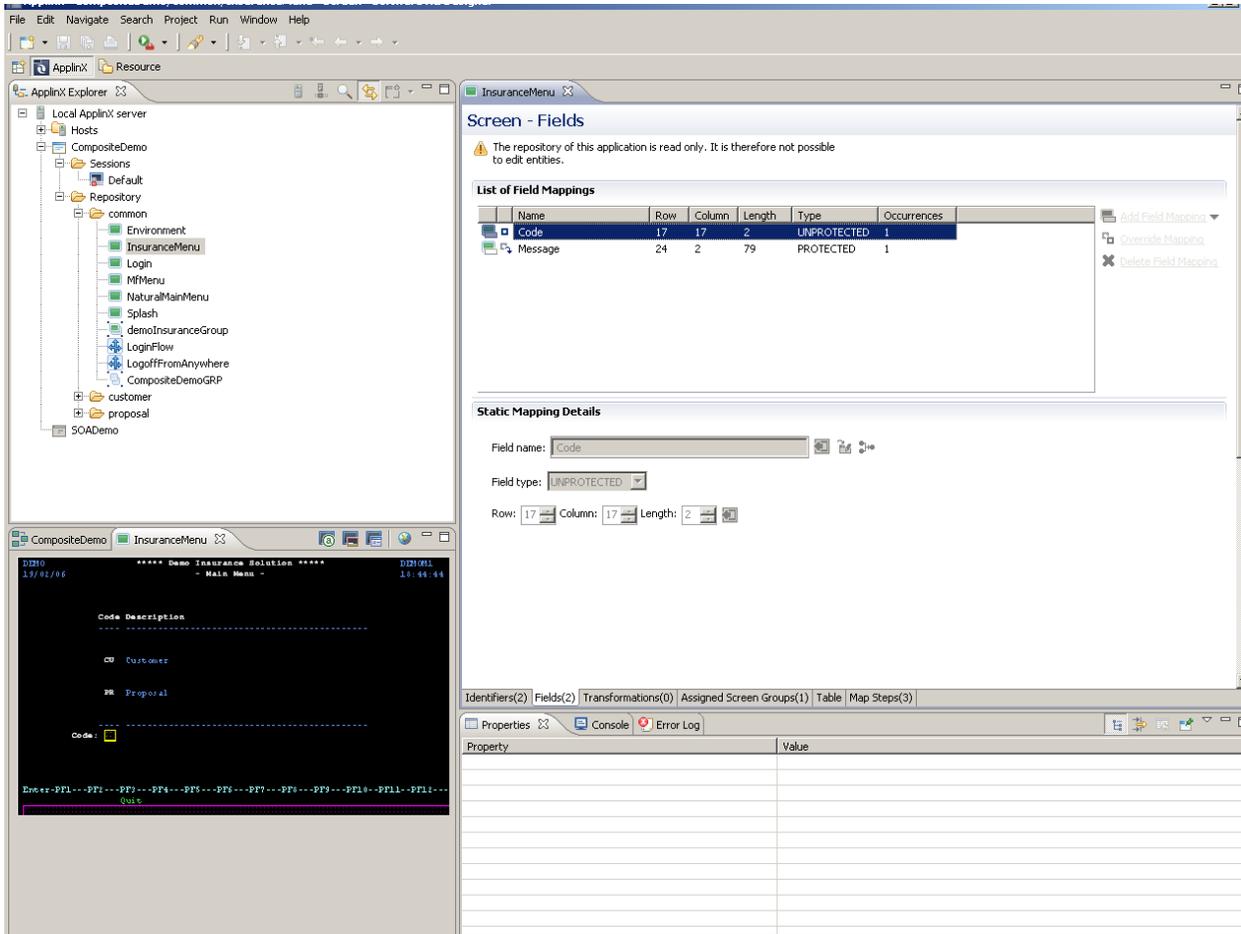
Additional connections to servers can be added. To add an additional server connection, in the **Server** menu, select **Add ApplinX Server Connection...**. The wizard is displayed. Enter the name of the server, the relevant address (IPv4 and IPv6 address formats are supported) and port. Also determine whether the connection will be a secured connection.

For further details regarding the server configuration, refer to *Configuring the Server* in the *Administration* documentation.

The Designer Layout

- ApplinX Related Menu Bar Commands
- ApplinX Explorer
- Editor
- Application Map
- Properties
- Console
- Error Log
- Status Bar

By default the layout is as displayed below, but as possible in Eclipse, the layout can be changed.



ApplinX Related Menu Bar Commands

Detailed below are the ApplinX related commands that are included in the menu bar.

File Menu

New > ApplinX Application...	Creates a new ApplinX application.
New > ApplinX Host...	Creates a new ApplinX host.
New > Entity	Creates a new entity from the list of entities.
New > Session	Creates a new display session or printer session (according to the selected option).
Import....	Enables importing ApplinX entities or importing/replacing an ApplinX application.
Export...	Enables exporting ApplinX entities or application.

Session Menu

Duplicate Session Definition	Creates another session with the same definitions as the selected session.
Connect	Connects to the host using the selected session definitions.
Disconnect	Disconnects the session from the host.
Open HTTP Session	Opens the session as an instant web application. The session is displayed by default in the Editor area of Eclipse. It is possible to display the session in your system's browser by changing the web browser definitions in Eclipse (Window > Preferences > General > Web Browser). This is convenient for checking transformations. This is only available when you have a web enablement license.

Entity Menu

Show WSDL	Displays the WSDL in the Editor area.
Copy WSDL URL	Copies the WSDL URL to the Clipboard.
Open HTML Preview	Displays an HTML preview of the screen. Useful to preview transformations that are used in this screen.
Generate > JSP Page from Screen...	Generates a JSP page from the current screen.
Generate > JSP Page from Screen Group...	Generates a JSP page from the current screen group.
Generate > Procedure Client...	Generates a Procedure Client from a procedure group.
Generate > JSP Page from Procedure...	Generates a JSP page from a procedure.
Convert > Screen to Screen Group	Converts a screen to a screen group.
Convert > Screen Group to Screen	Converts a screen group to a screen.
Convert > Path / Path Wrapper to Path Procedure	Converts a path to a path procedure.

Application Menu

Open Application	Opens the application.
Close Application	Closes the application.
Reload Application	Reloads the repository.
Import Predefined Transformations	Imports into the application a list of predefined transformations.
Import Host Screen Maps	Opens a wizard which enables importing screen maps.
Show Application Map	Displays the Application Map view.
Automatically Identify Screens/Steps using Trace Files...	Opens a wizard which enables to automatically identify screens.
Extract Business Process Activities	Opens a wizard which enables to extract business activities from trace files.

Web Application Manager...	Initiates the Web Application Manager wizard which enables creating a web application.
Deployment Manager for J2EE...	Initiates the Deployment Manager wizard which enables deploying.
Open in File System	Opens the file system and allows viewing the files in the relevant folder.

Server Menu

Add ApplinX Server Connection...	Adds an additional server connection.
Connect to Server	Connects to the server.
Disconnect from Server	Disconnects from the server.
Reload	Reloads the server configuration, i.e. the configuration of all applications connected to the server.
Show Web Services	Displays a list of the available web services.

ApplinX Explorer

The ApplinX Explorer is comprised of the ApplinX Explorer toolbar and the ApplinX Explorer tree.

ApplinX Explorer Toolbar

	Adds a new server connection.
	Connects to the selected server.
	Disconnects from the selected server.
	Enables finding an entity.
	Enables creating new entities in the currently selected application.

ApplinX Explorer Tree

	Local ApplinX server
	Hosts
	<Host name>
	<Application name>
	Session (disconnected)
	Session (connected)
	Printer Session

	Repository
	Screen
	Screen Group
	Transformation
	Connection Pool
	Connection Information Set
	Procedure Group
	Flow Procedure
	Path Procedure
	Program Procedure
	Data Structure
	Database Connection
	External Web Service

Editor

The various entity editors are displayed in the editor area: Screen, Screen Group, Transformation, Connection Pool, Connection Information Set, Path Procedure, Procedure Group, Flow Procedure, Program Procedure, Data Structure, Database Connection and External Web Service. This area also displays the HTML preview of a screen, the list of all relevant web services and a specific WSDL.

Application Map

The **Application Map** view displays application screens through which the user has navigated. Refer to *Application Map* in the *Designing and Developing an Application* documentation for further details.

Properties

This area displays basic details regarding the object selected in the ApplinX Explorer tree. For example, when selecting the Server, the **Properties** area displays a number of parameters such as the server address, port, state and version. Another example is when selecting the Repository, or a folder within the repository, the **Properties** area displays parameters such as the application name, a detailed list of all the entities, and the number of each type of entity.

Console

This area displays the console messages.

Error Log

This area displays the error log.

Status Bar

Online/Offline indication	Name of GCT trace file/application name.
Screen name	UNKNOWN when the screen has not been identified.
Cursor position	Row and column, and position in the host.
Mouse position	Row and column, and position in the host.
Selection start	Starting position of the currently selected area (in host units).

10

Creating a New Application

An application is a project which can contain an internal system within an organization, such as human resources, billing, customer services etc. In this task, you will create a new ApplinX application. Every ApplinX application is connected to a host and has a repository containing all the ApplinX entities. The steps detailed in this task include the basic steps required to create an application. Advanced configuration is explained in *Application Configuration Parameters* in the *Reference Guide*.

» To create an ApplinX application

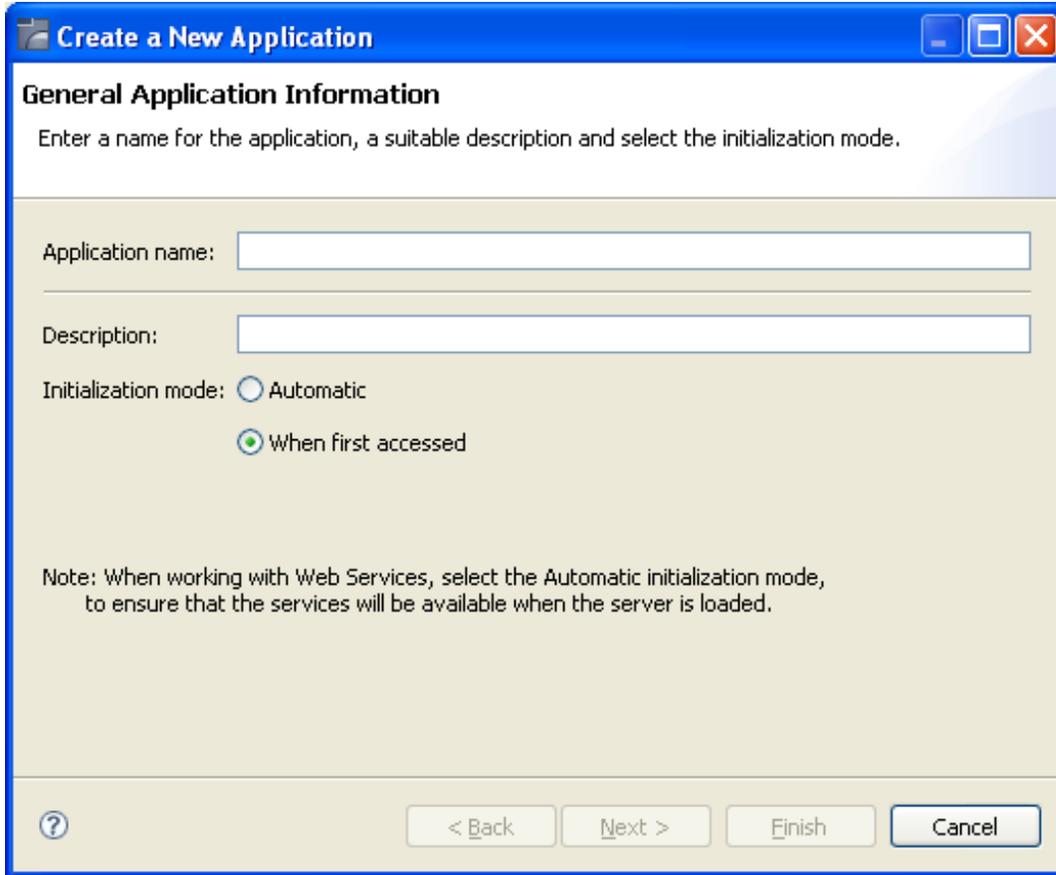
Before creating a new application, ensure that ApplinX server is available and **running**, and start the **Designer**. Log in to the relevant server or add an additional server as required (refer to the sections *Logging on* and *Adding a New Server Connection*).

- 1 In the ApplinX Explorer, right-click on the ApplinX server and select **New Application...** .

Or:

Select the menu item **File > New > Other...** In the resulting window, expand the **Software AG** node, then the **ApplinX** node, select **Application**, and then click **Next**.

The **Create a New Application** wizard is displayed.



- 2 Enter a name for the application and a suitable description.
- 3 Select the initialization mode:

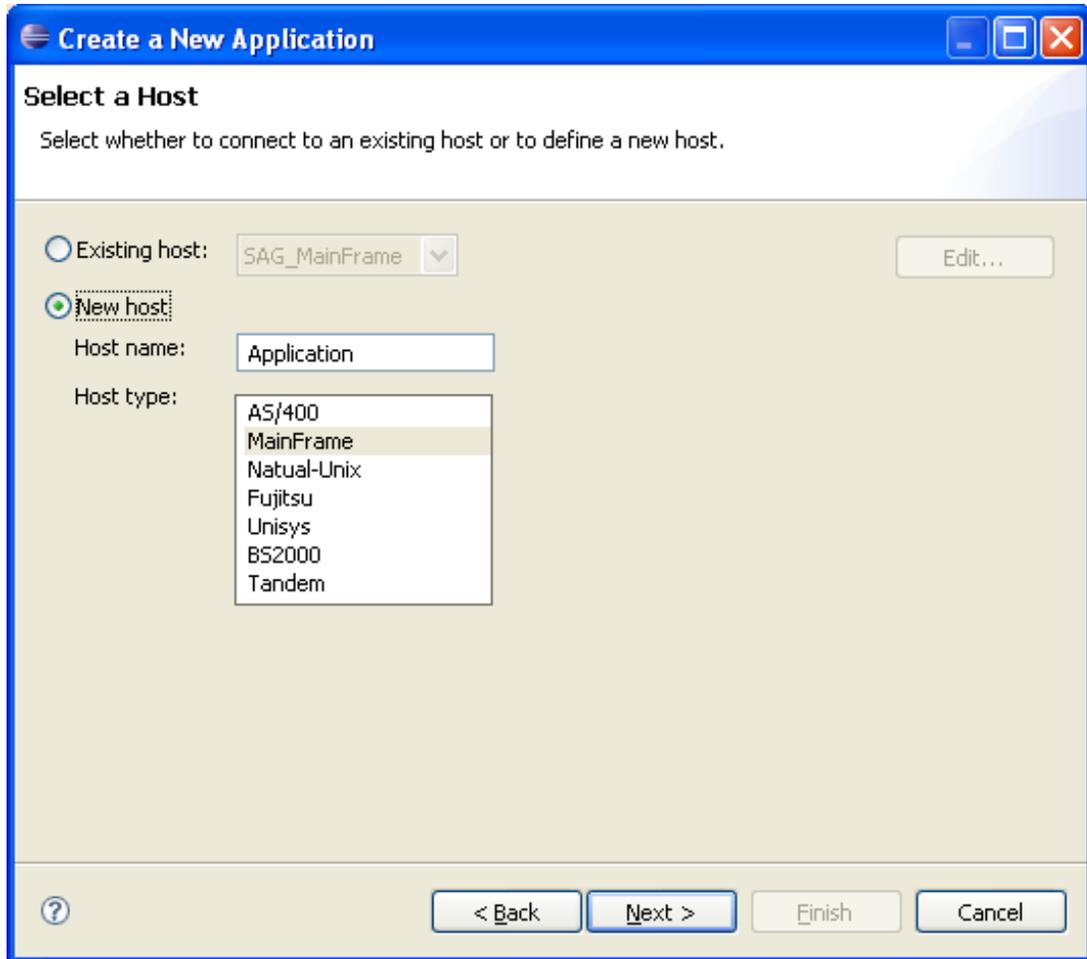
Automatic

Automatically loaded when the server is started.

When first accessed

Loaded when first accessed, in other words, when the code that initializes startup is first called (default).

- 4 Click **Next**.

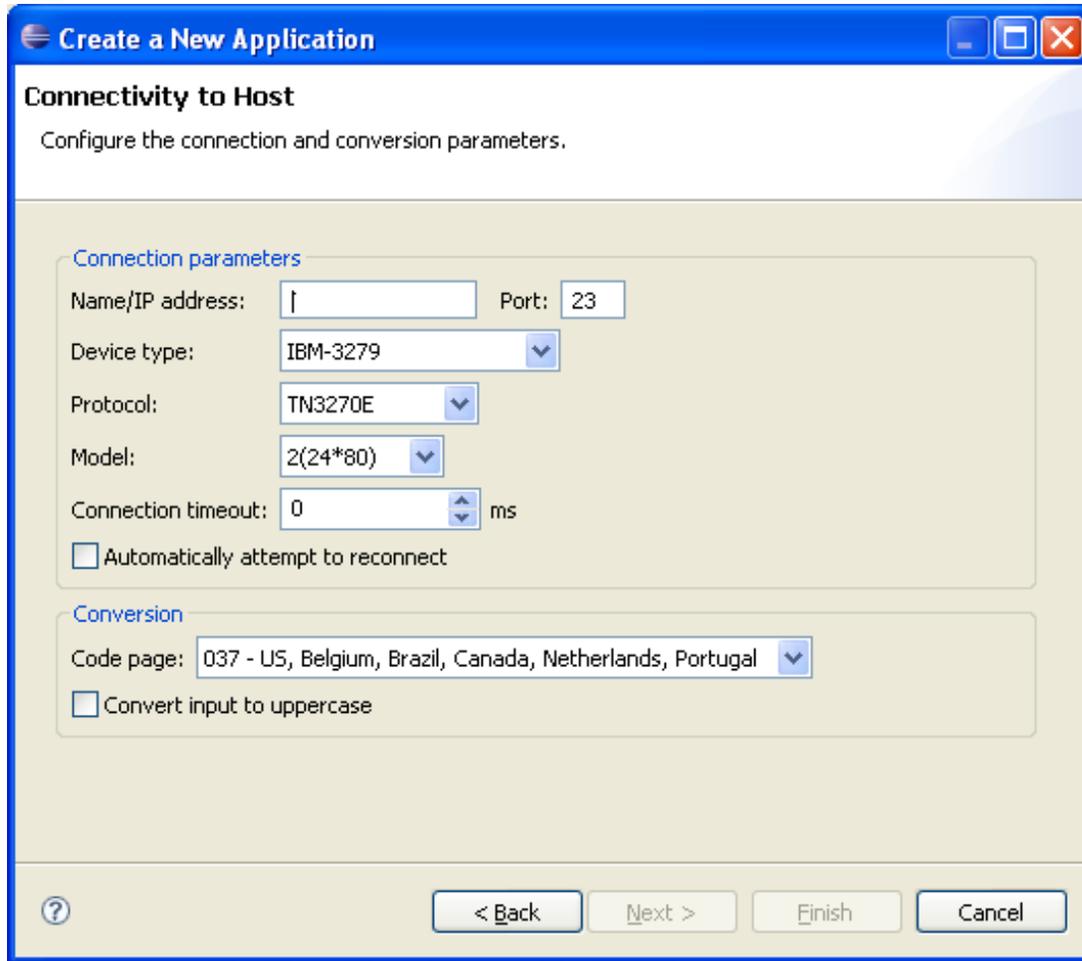


Select whether to use an existing host or to create a new host.



Note: SAG_MainFrame is accessible only from within the Software AG network. Therefore, this host can only be selected when working offline using a trace file.

- 5 When selecting to define a new host, enter a name for the host and select the relevant type of host. Click **Next**.



The host connection and conversion parameters are displayed. It is mandatory to enter the host's name/IP address (IPv4 and IPv6 address formats are supported). Configure the other parameters as required. Further details of these parameters are available in *Host Configuration Parameters* in the *Reference Guide*.

- 6 Click **Finish**. The application is displayed in the ApplinX Explorer. It is possible to further configure the application properties by right-clicking on the application name and selecting **Properties**. Refer to *Application Configuration Parameters* in the *Reference Guide* for details regarding the advanced configuration properties.

Confirming Connectivity

» To confirm connectivity to the host

- 1 In the ApplinX Explorer, expand the relevant application node.
- 2 Expand the **Sessions** node. Right-click on **Default** and select **Connect**. The first host screen is displayed and the icon of the default session changes to . Navigate between the screens

to confirm connection and host functionality. If there is a connection problem, check that the host is working and that the host definitions are correct.

Further information regarding application configuration can be found in *Defining Applications* in the *Designing and Developing an Application* documentation.

11

Creating an ApplinX Web Application

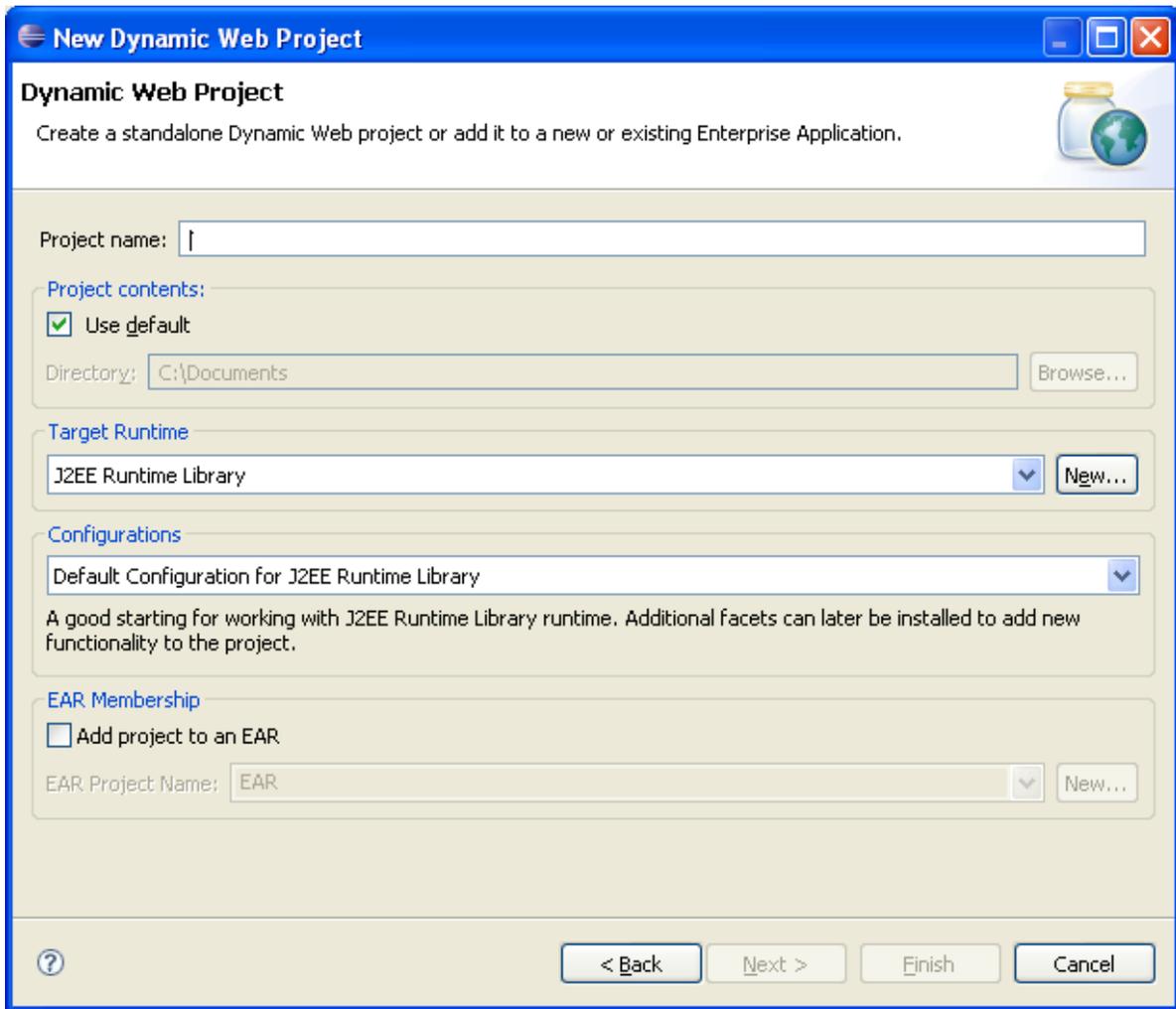
- Creating an ApplinX for JSP Web Application using the Software AG Designer (recommended) 52
- Creating an ApplinX for JSP Web Application using JDK and Tomcat (no IDE) 60
- Creating an ApplinX .NET Web Application using Visual Studio 2008 65
- Creating an ApplinX .NET Web Application using Visual Studio 2010 65
- Creating an ApplinX .NET Web Application using .NET (no IDE) 66
- Upgrading an ApplinX .NET Web Application from Visual Studio 2003 to 2005 71
- Upgrading an ApplinX .NET Web Application from Visual Studio 2003 to 2008 71
- Upgrading an ApplinX .NET Web Application to Visual Studio 2010 72

Creating an ApplinX for JSP Web Application using the Software AG Designer (recommended)

The following is based on the released Eclipse version. Additional information can be found on Eclipse website and WTP (web tool platform) site.

> To create a new Dynamic Web project

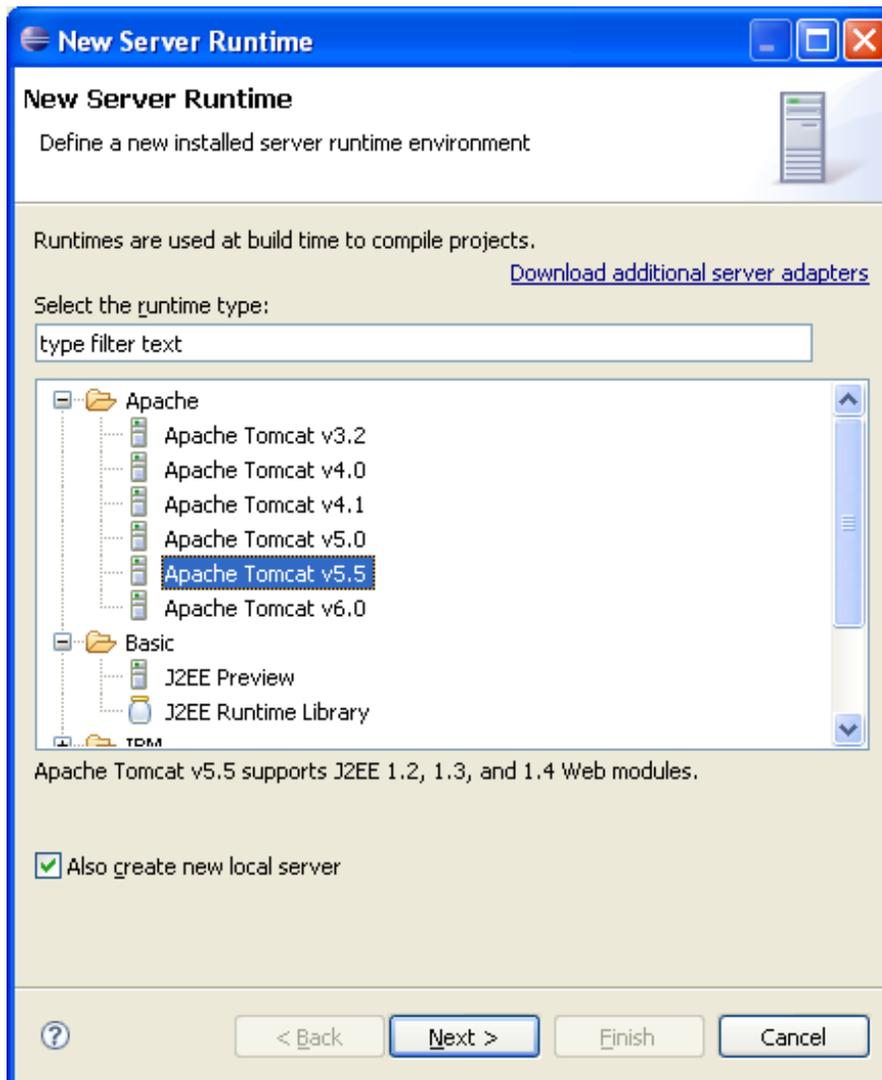
- 1 In the Eclipse workspace, select **File > New > Project** to create a new project.
- 2 Select **Web > Dynamic Web Project**.
- 3 Click **Next** to create a new web project. The **New Dynamic Web Project** dialog box is displayed.



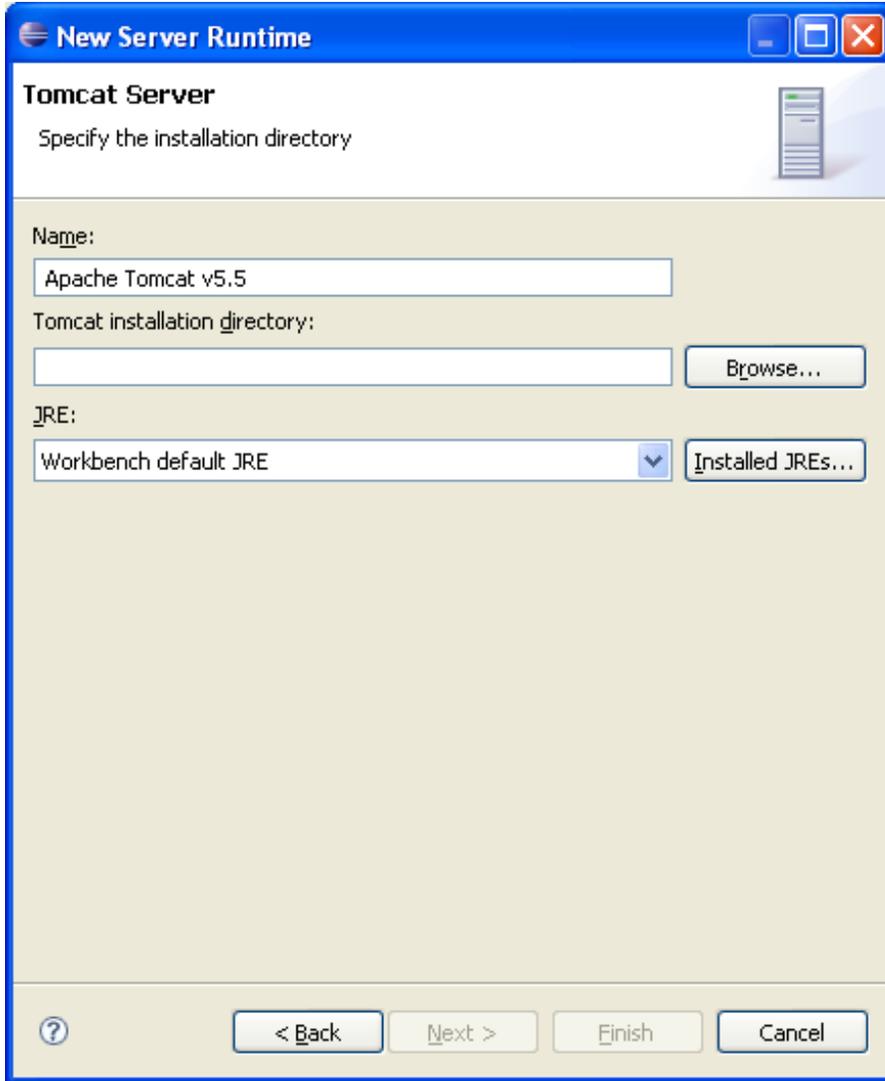
- 4 Enter a name for the project.

- 5 In the **Target Runtime** field, select the Tomcat installation: click **New...** and select the server to be used: Either **J2EE Preview** (recommended) or **Tomcat v5.5**.

 **Important:** The Restart function in the J2EE Preview server does not work properly. Instead of using the Restart function, stop and the start the server. In addition, editing files only works after stopping and starting the server.



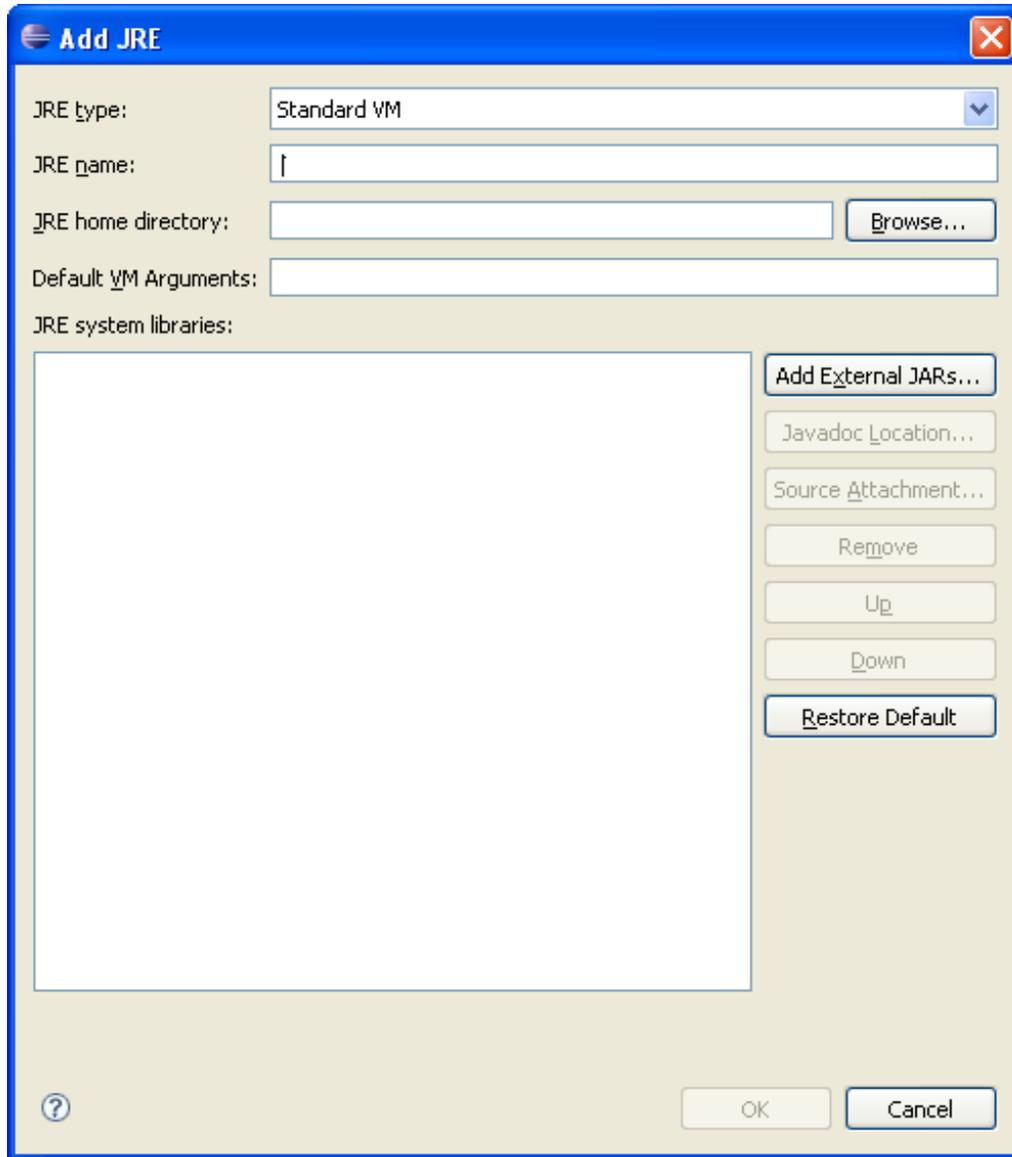
- 6 Click **Next**.



7 Browse and select the Tomcat installation.

 **Note:** Currently the ApplinX internal Tomcat cannot be used to create a web application and when selecting to use Tomcat, you are required to install an additional Tomcat 5.5 version.

8 Select a JRE or click **Installed JREs...** to add a new JRE. The **Preferences** dialog box is displayed.

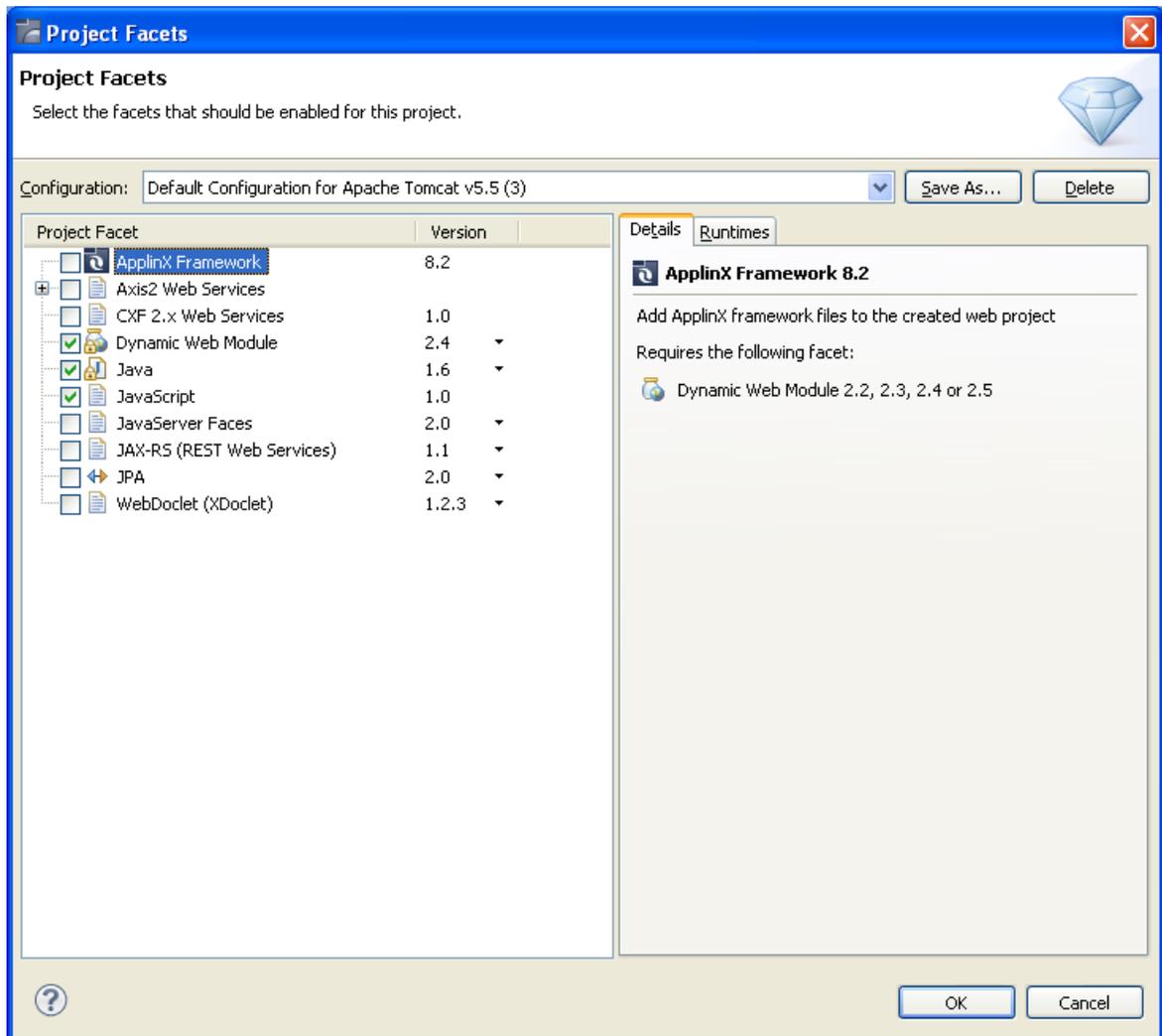


- 10 Assign a name to the new JRE.
- 11 Enter the JRE home directory (when necessary, click **Browse...** to locate the file).
- 12 Add VM arguments when required.
- 13 Click **OK** to return to the **New Server Runtime** dialog box. Select the newly created JRE. Click **OK**.
- 14 Click **Finish**. The **New Dynamic Web Project** dialog box is displayed.

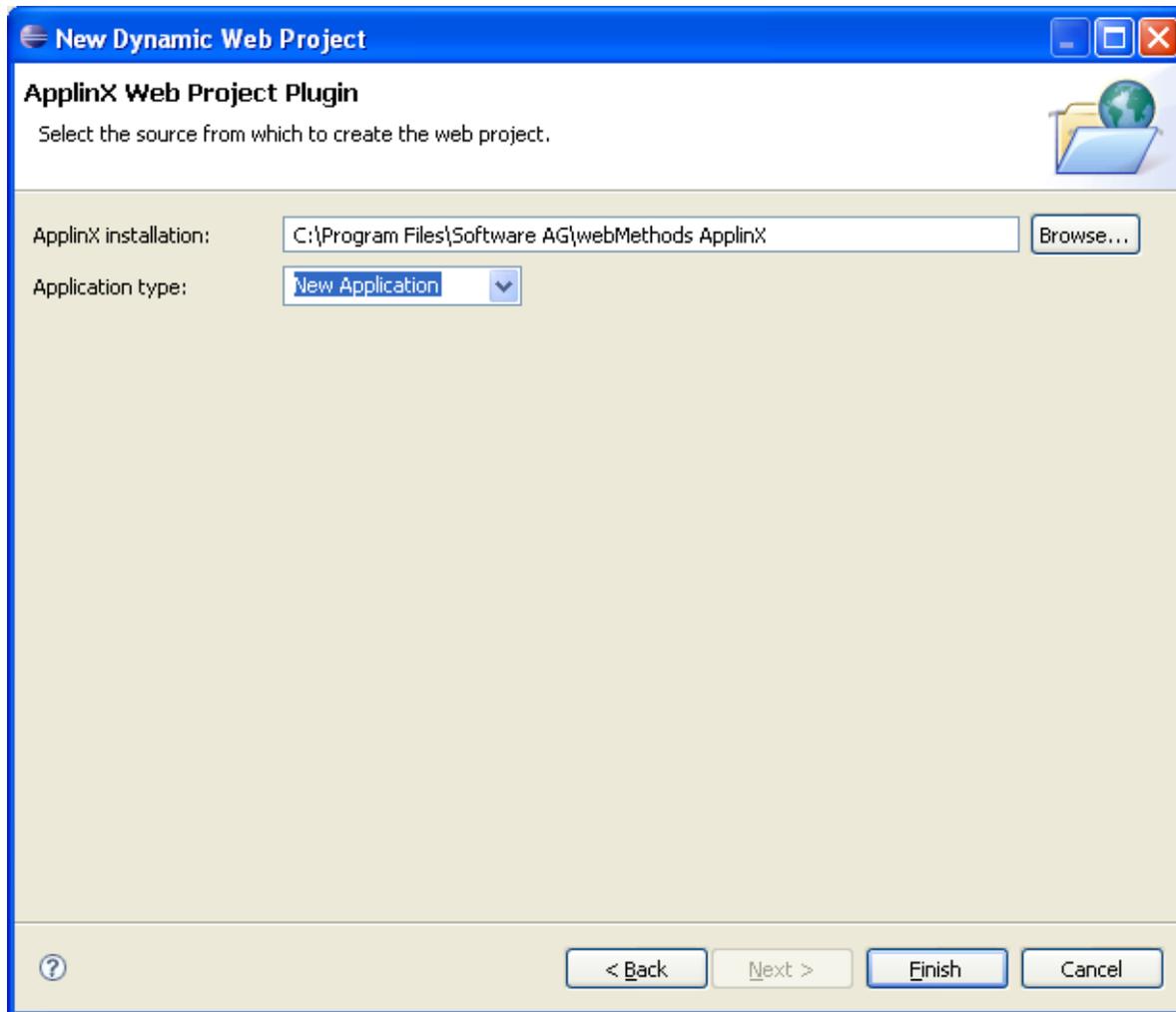
You have completed the configuration of the flexible project, the server (Tomcat) and the JDK.

- 15 Change the configuration to **Custom** and click **Modify**.

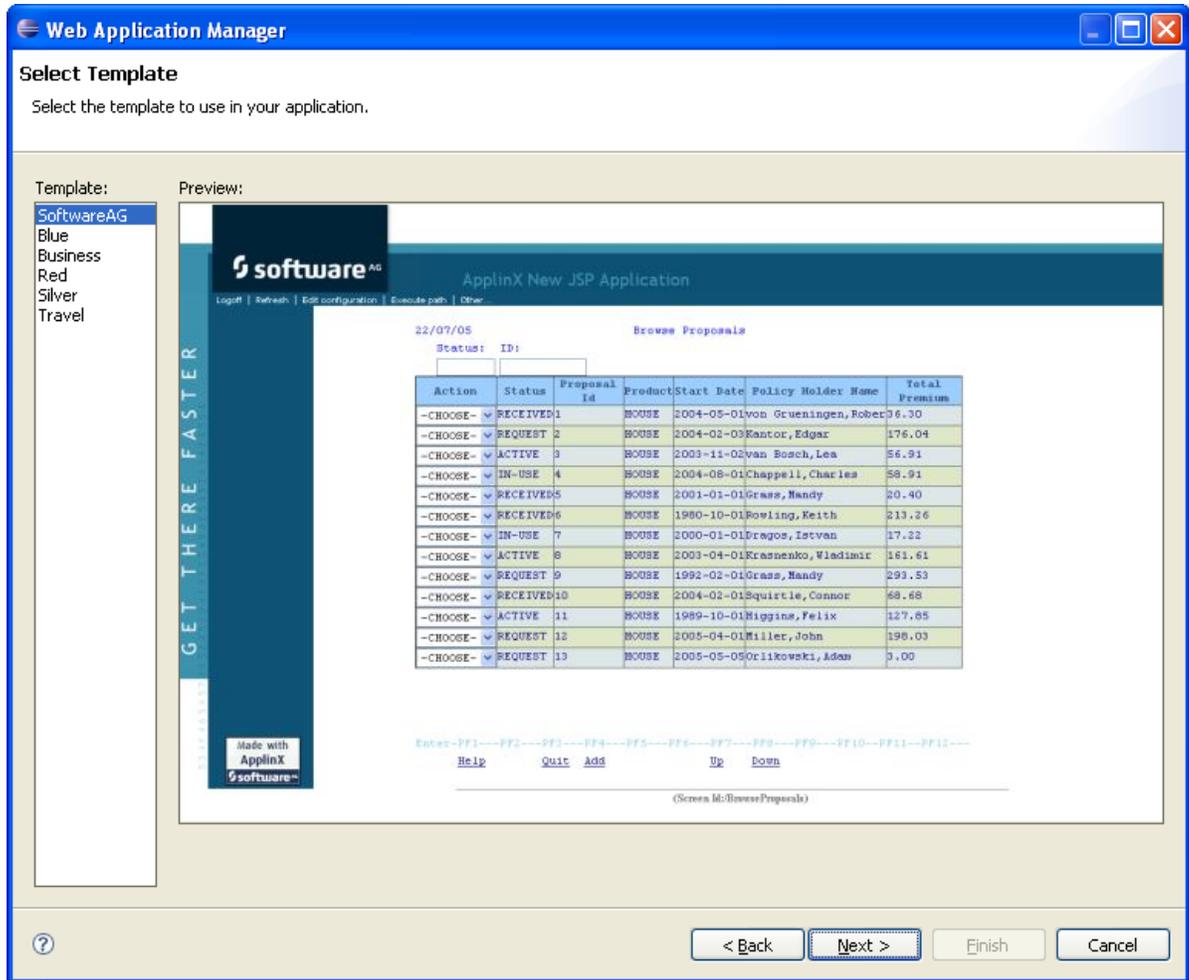
Select **ApplinX Framework**. Ensure that the **Dynamic Web Module** and **Java** check boxes are selected. Click **OK**.



- 16 Click **Next**.
- 17 Click **Next**.
- 18 Select ApplinX installation from which to create the web project and select the application type.



19 Click Next.



Select the template to use in your application.

- 20 Click **Finish**.

You have completed the creation of a standard WAR file (Java Web Archive).

- 21 Ensure that in the **Project** menu, **Build Automatically** is selected.

To run the project, right-click on the project's folder in the **Project Explorer** view and select **Run As > Run on Server...**

The first time you run a project you are required to select a default server:

Select the server you configured.

Select the **Always use this server when running this project** check box. The next time you run the application, this dialog box will not be displayed. The server you selected will automatically be used.

Click **Finish** to continue. A browser window will be opened.



Note: The browser type can be configured under **Windows > Preferences > General > Web Browser**. It is recommended to configure it to **Default system web browser**.

Using the Javadoc as a reference within the code

When using the Javadoc you will have:

- Online usage of the Eclipse Javadoc for ApplinX API.
- Suggested argument names for the ApplinX API, instead of arg0, arg1, etc.

➤ To use Javadoc capabilities when developing your project

- 1 Right-click on the project node and select **Build Path > Configure Build Path....** The **Properties** dialog box is displayed.
- 2 In the **Libraries** tab, expand the **gxframework.jar** node, click on **Javadoc location** and then click on the **Edit...** button to determine the location of the Javadoc.
- 3 You can either enter the URL `http://localhost:2380/applinx-help` or browse and select the Javadoc location path **<Software AG installation> > ApplinX > Help > API**. Then click **OK**.
- 4 Click **OK** to close the **Properties** dialog box.

Creating an ApplinX for JSP Web Application using JDK and Tomcat (no IDE)

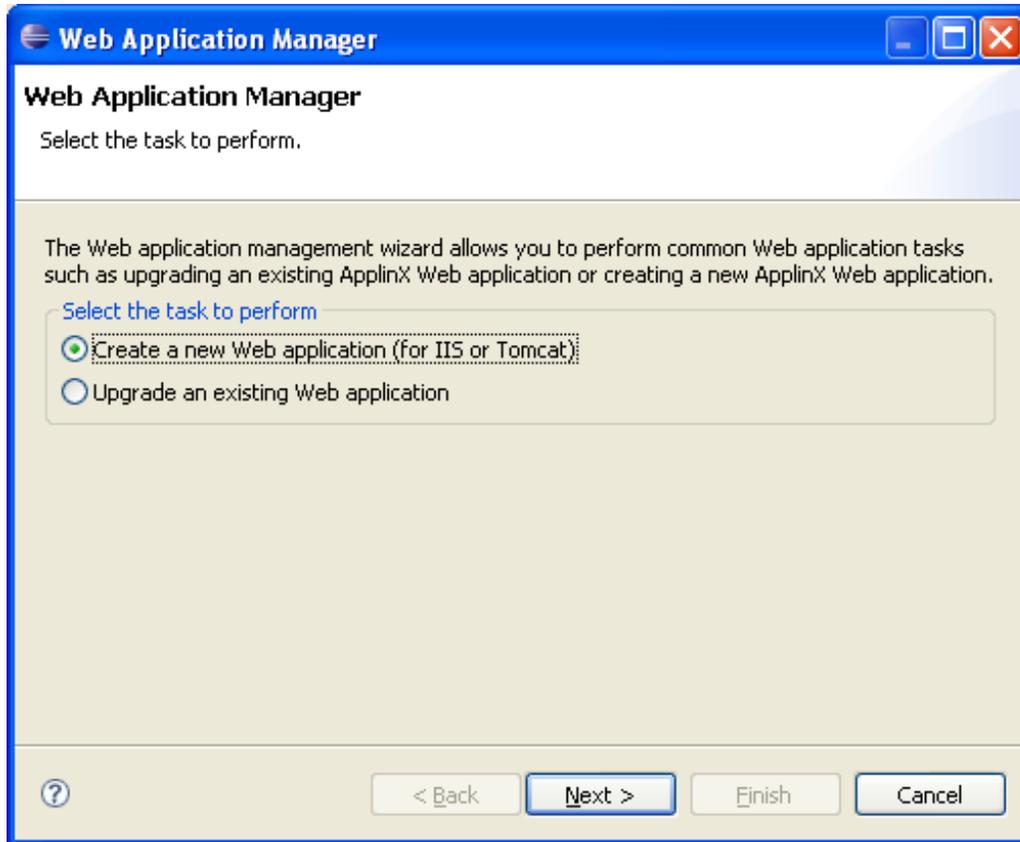


Note: This is not available when working with a SOA license.

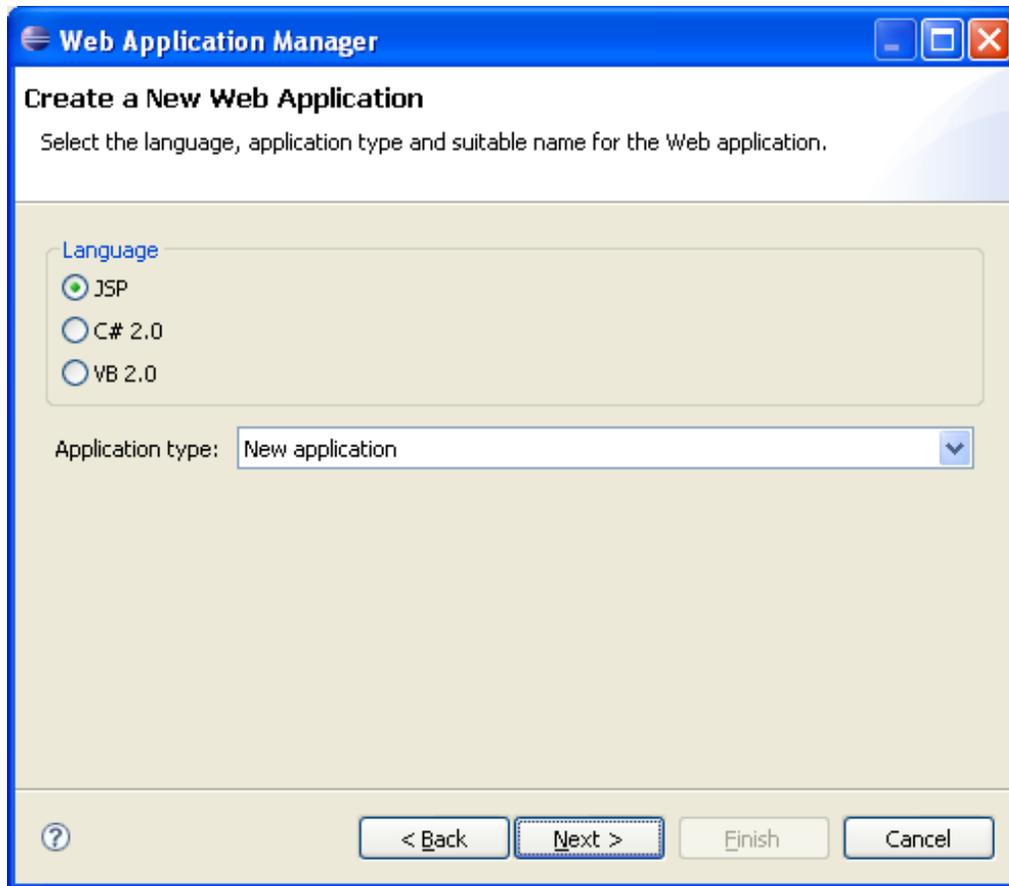
➤ To create a new JSP application without any IDE and with JDK

Refer to [ApplinX for JSP Web Application](#) in *Software Requirements*.

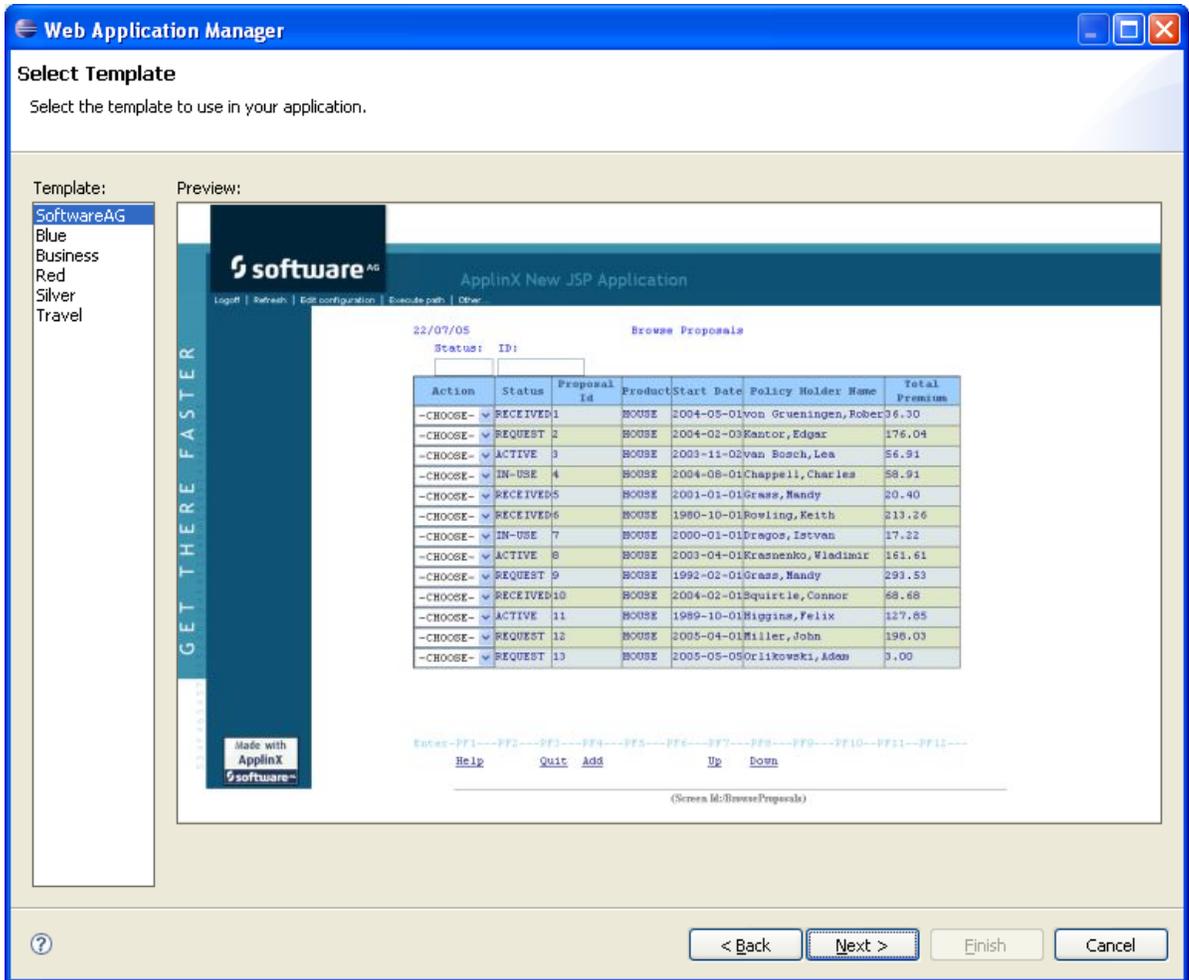
- 1 During the installation process, choose the option **ApplinX Framework for JSP**.
- 2 In the ApplinX Explorer, right-click on the relevant application and select **Web Application Manager....** The **Web Application Manager** wizard is displayed.



- 3 Ensure that **Create a new Web application (for IIS or Tomcat)** is selected. Click **Next**. The **Create a New Web Application** screen is displayed.

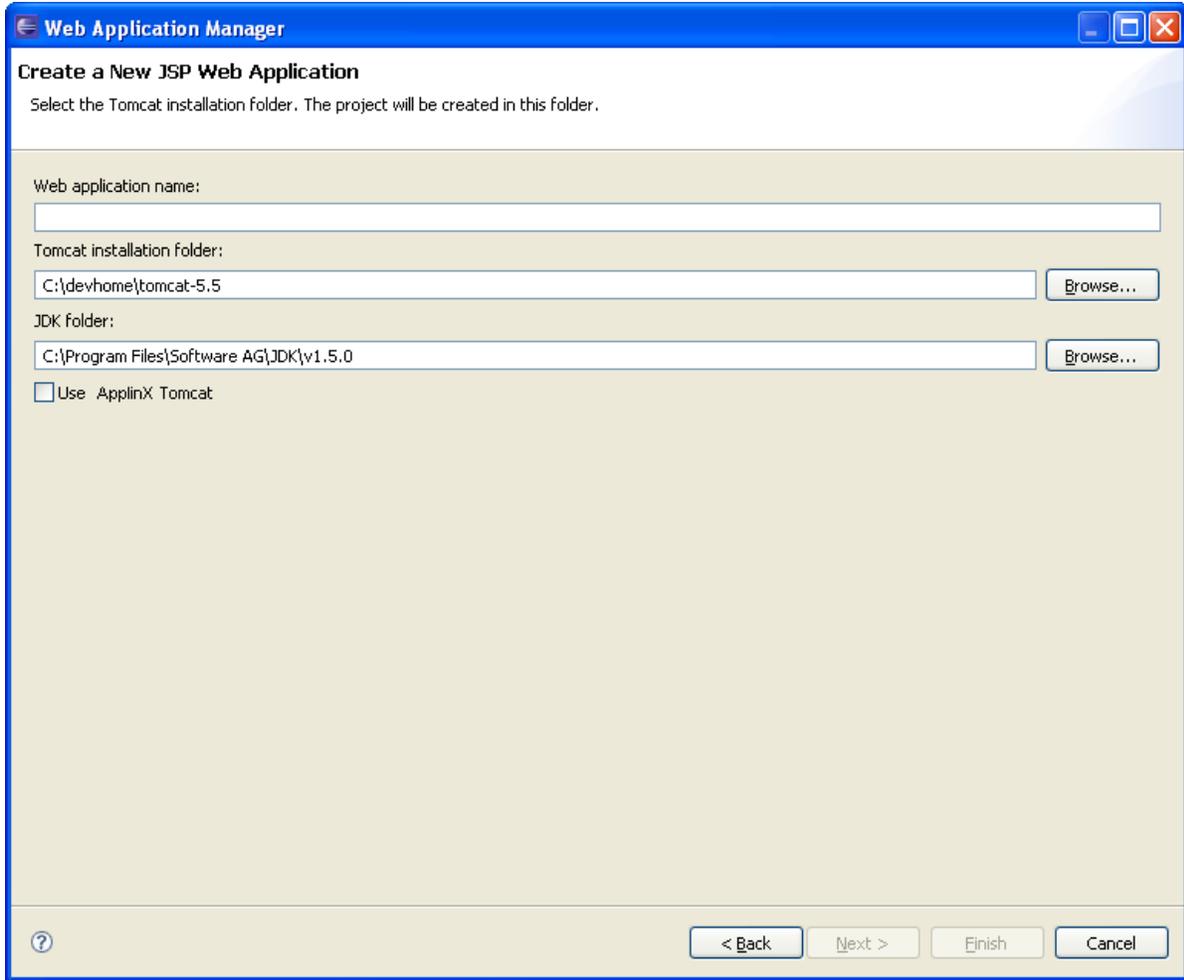


- 4 Ensure that **JSP** is selected. Select the application type: **New application** to create a basic ApplinX web application with a web look and feel or **Emulation template** to create an ApplinX web application with a host look and feel.
- 5 Click **Next**. The **Select Template** screen is displayed.



Select a template from the list on the left.

- 6 Click **Next**. The **Create a New JSP Web Application** screen is displayed.



Enter a name for the web application. By default this is the name of the ApplinX application that you selected. Browse and select the folder where Tomcat is installed. Select the location of the JDK installation.

When you wish to use the ApplinX internal Tomcat, select the corresponding check box.

- 7 Click **Next**. The **Wizard Summary** screen is displayed.
- 8 Select whether to launch the browser when the process is completed.
- 9 Select whether to restart Tomcat. This is only relevant when Tomcat is activated from a batch file. The Framework Manager does not support restarting Tomcat when Tomcat runs as a Windows Service (in this case, you need to manually restart Tomcat).
- 10 Click **Finish**. A message will be displayed indicating whether the process succeeded or failed.



Note: Any change in the Java file requires recompilation of your new application. Changes in JSP files do not require recompilation. Furthermore, Tomcat requires reloading: `http://localhost:8080/manager/reload?path=/<YOUR_APP_NAME>`.

Creating an ApplinX .NET Web Application using Visual Studio 2008

➤ To create a new ApplinX web application using Visual Studio 2008

- 1 Install J# 2.0 redistributable package Second edition.
- 2 Install Microsoft Visual Studio 2008 on your computer (install Visual Studio before installing ApplinX so that when installing ApplinX, the ApplinX web application templates will be added to the available Visual Studio New Web Sites).
- 3 Open Visual Studio 2008.
- 4 In the **File** menu, select **New** and then **Web Site...** to create a new web site. The **New Web Site** dialog box is displayed.
- 5 Select the relevant ApplinX web application type (Visual C# or Visual Basic).
- 6 Determine whether you would like to upgrade to .NET framework 3.5 or continue working with .NET framework 2.0 (both options are acceptable).
- 7 Define that *index.aspx* will be the start page (right-click on *index.aspx* and select **Set as Start Page**).
- 8 Run the web site.
- 9 In the Framework Configuration Editor, define the connection parameters as follows:
 1. Click on the **Framework Configuration** link. The Configuration Editor will be displayed.
 2. In the **Session** node, enter the application name. By default, the application name is "InstantDemo". In the **Session** node, it is possible to edit and configure additional connection and project properties.
 3. Click **Save** to save your changes.
 4. Click **Close** to return to the web application.

Creating an ApplinX .NET Web Application using Visual Studio 2010

➤ To create a new ApplinX web application using Visual Studio 2010

- 1 Install J# 2.0 redistributable package.
- 2 Install .NET Framework 3.5 SP1.
- 3 Install Microsoft Visual Studio 2010 on your computer (install Visual Studio before installing ApplinX so that when installing ApplinX, the ApplinX web application templates will be added to the available Visual Studio New Web Sites).

- 4 Open Visual Studio 2010.
- 5 In the **File** menu, select **New** and then **Web Site...** to create a new web site. The **New Web Site** dialog box is displayed.
- 6 Select to work either with .NET Framework 2.0 or 4.0.
- 7 Select the relevant ApplinX web application type (Visual C# or Visual Basic).
- 8 Define that *index.aspx* will be the start page (right-click on *index.aspx* and select **Set as Start Page**).
- 9 Run the web site
- 10 In the Framework Configuration Editor, define the connection parameters as follows:
 1. Click on the **Framework Configuration** link. The Configuration Editor will be displayed.
 2. In the **Session** node, enter the application name. By default, the application name is "InstantDemo". In the **Session** node, it is possible to edit and configure additional connection and project properties.
 3. Click **Save** to save your changes.
 4. Click **Close** to return to the web application.

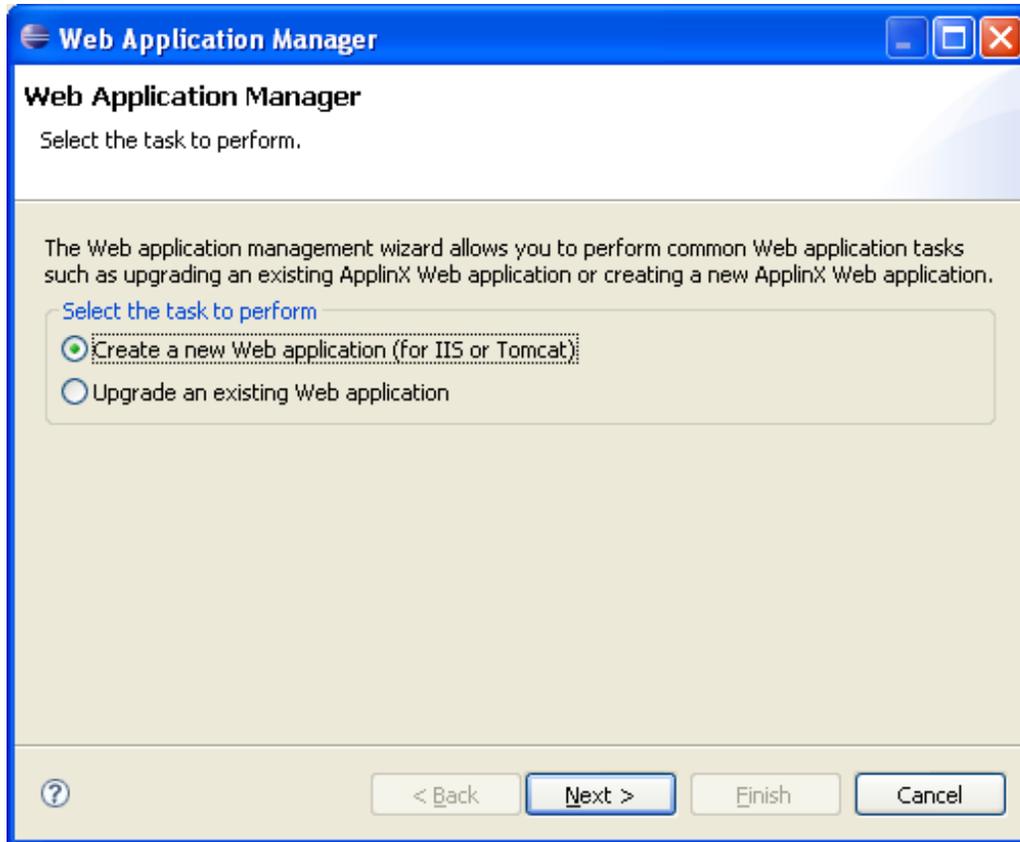
Creating an ApplinX .NET Web Application using .NET (no IDE)



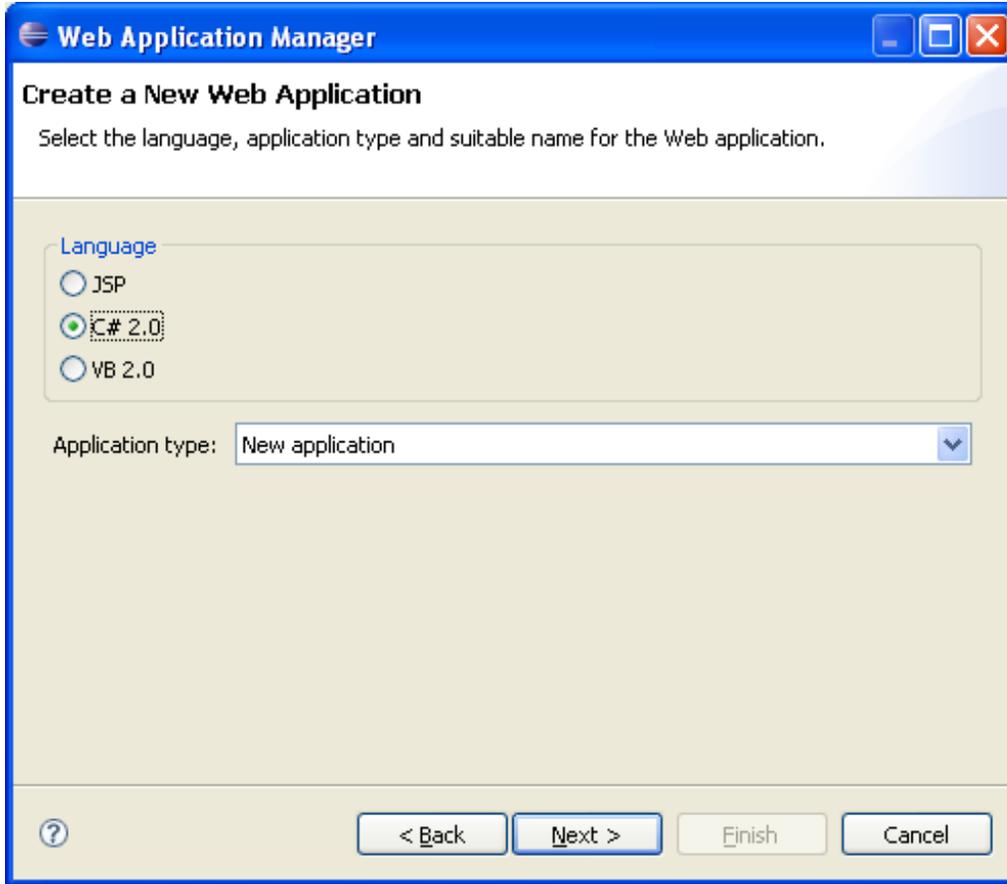
Note: This is not available when working with a SOA license.

» To create a new ASP.NET application without Visual Studio .NET with .NET SDK

- 1 During ApplinX installation, select the ApplinX ASP.NET Web Framework C#/VB .NET option.
- 2 In the ApplinX Explorer, right-click on the relevant application and select **Web Application Manager...** The **Web Application Manager** wizard is displayed.



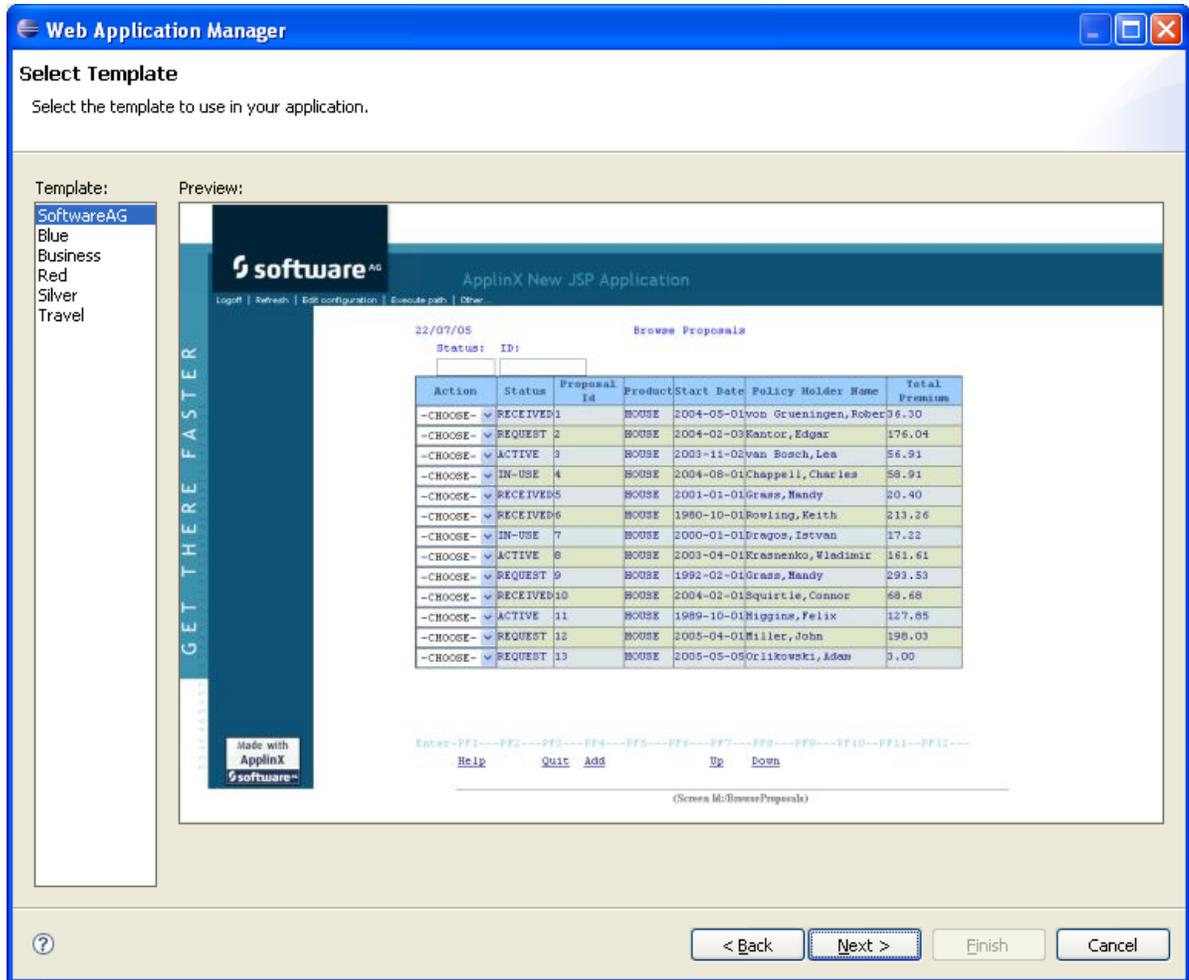
- 3 Ensure that **Create a new Web application (for IIS or Tomcat)** is selected. Click **Next**. The **Create a New Web Application** screen is displayed.



- 4 Ensure that **C# 2.0** or **VB 2.0** is selected. Select the application type: **New application** to create a basic ApplinX web application with a web look and feel or **Emulation template** to create an ApplinX web application with a host look and feel.

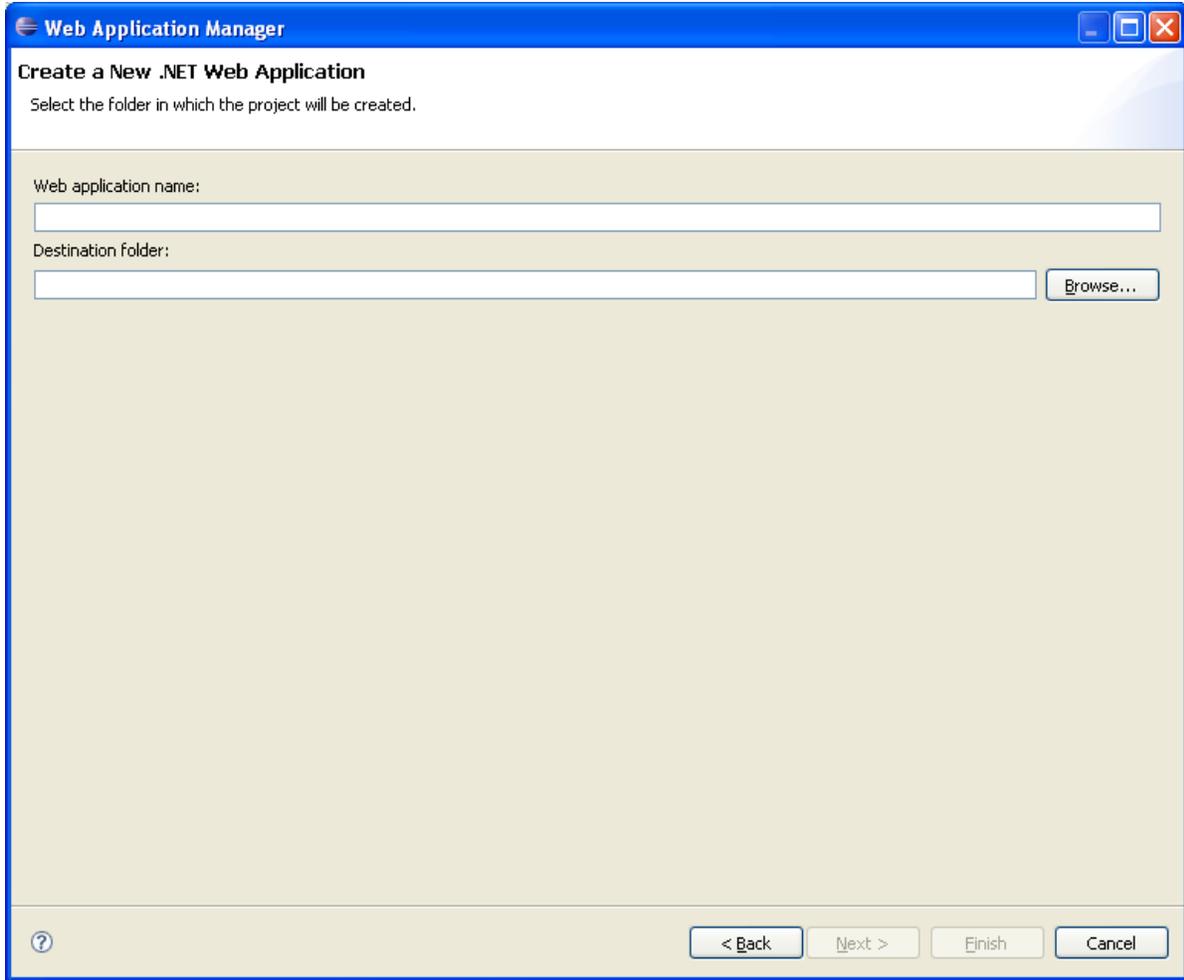
 **Note:** If you wish to create a new web application based on .NET 4.0, you need to use Visual Studio 2010 (refer to [Creating an ApplinX .NET Web Application using Visual Studio 2010](#)).

- 5 Click **Next**. The **Select Template** screen is displayed.



Select a template from the list on the left. Click **Next**.

- 6 The **Create a New .NET Web Application** screen is displayed.



- 7 Enter a name for the web application. By default, this is the name of the ApplinX application that you selected. Browse and select the destination folder.
- 8 Click **Next**. The **Wizard Summary** screen is displayed.
- 9 Select whether to launch the browser when the process is completed.
- 10 Click **Finish**. A message will be displayed indicating whether the process succeeded or failed.

 **Note:** When using the Application Manager on IIS 7.5 or 8.0, you need to convert your application from a virtual directory to a web application (in the IIS manager, right-click on your application and select **Convert to Application**).

Upgrading an ApplinX .NET Web Application from Visual Studio 2003 to 2005

➤ To upgrade a project from Visual Studio 2003 to Visual Studio 2005

- 1 In Visual Studio 2005, open the web application by clicking **File > Open > WebSite...** and selecting the web application.
- 2 The Visual Studio Conversion Wizard is displayed. Click **Next**.
- 3 Determine whether you would like to create a backup before converting.
- 4 Click **Finish**.
- 5 The conversion process is run. Once completed, errors are displayed in the conversion log. When necessary, make the required changes to correct these errors (error lines relating to the project file do not need to be dealt with, as from this point, you are working with a web site and not a web project). For example, an error "Unable to move file" should be corrected by manually moving the file.
- 6 Build the web site. If errors are displayed, check what these errors are and make the necessary changes.
- 7 Close the web site (**File > Close > Solution**).
- 8 In ApplinX Designer, upgrade your web application using the ApplinX Web Application Manager (refer to *Upgrading a .NET Web Application* in *Web Application Development*).
- 9 Reopen the web application in Visual Studio 2005 (**File > Open > WebSite...** and select the web application).
- 10 Build the web site. Correct errors.
- 11 Define that *index.aspx* will be the start page (right-click on *index.aspx* and select **Set as Start Page**).
- 12 You can run the web site to ensure that it performs as expected.

Upgrading an ApplinX .NET Web Application from Visual Studio 2003 to 2008

➤ To upgrade a project from Visual Studio 2003 to Visual Studio 2008

Ensure that you have J# 2.0 redistributable package installed.

- 1 In Visual Studio 2008, open the web application by clicking **File > Open > WebSite...** and selecting the web application.
- 2 The Visual Studio Conversion Wizard is displayed. Click **Next**.
- 3 Determine whether you would like to create a backup before converting. Click **Next**.

- 4 Determine whether you would like to work with .NET 2.0 or .NET 3.5.
- 5 Click **Finish**.
- 6 The conversion process is run. Once completed, errors are displayed in the conversion log. When necessary, make the required changes to correct these errors (error lines relating to the project file do not need to be dealt with, as from this point, you are working with a web site and not a web project). For example, an error "Unable to move file" should be corrected by manually moving the file.
- 7 Build the web site. If errors are displayed, check what these errors are and make the necessary changes.
- 8 Close the web site (**File > Close > Solution**).
- 9 In ApplinX Designer, upgrade your web application using the ApplinX Web Application Manager (refer to *Upgrading a .NET Web Application* in *Web Application Development*).
- 10 Reopen the web application in Visual Studio 2008 (**File > Open > WebSite...** and select the web application).
- 11 Build the web site. Correct errors.
- 12 Define that *index.aspx* will be the start page (right-click on *index.aspx* and select **Set as Start Page**).
- 13 You can run the web site to ensure that it performs as expected.

Upgrading an ApplinX .NET Web Application to Visual Studio 2010

➤ To upgrade a project to Visual Studio 2010

Ensure that you have J# 2.0 redistributable package Second edition installed.

- 1 When upgrading to Visual Studio 2010 from Visual Studio 2003, you need to first perform the upgrade task above: either [Upgrading an ApplinX Web Application from Visual Studio 2003 to 2005](#) or [Upgrading an ApplinX Web Application from Visual Studio 2003 to 2008](#).
- 2 In Visual Studio 2010, open the web application by clicking **File > Open > WebSite...** and selecting the web application.
- 3 Determine whether you would like to work with .NET 2.0 or .NET 4.0.
- 4 In ApplinX Designer, upgrade your web application using the ApplinX Web Application Manager (refer to *Upgrading a .NET Web Application* in *Web Application Development*).
- 5 Define that *index.aspx* will be the start page (right-click on *index.aspx* and select **Set as Start Page**).
- 6 You can run the web site to ensure that it performs as expected.

12

Installing ApplinX Demo Applications

- Installing the Demo Web Application using the Software AG Designer 74
- Installing the Demo Web Application using Tomcat 74
- Installing the Demo Web Application (Other Java Application Servers) 75
- Installing the Demo Web Application using Visual Studio 76

ApplinX provides several demo applications: Instant demo, Composite demo and SOA demo, based on a typical legacy Natural mainframe application. These applications contain typical legacy screens such as update screens, task screens, windows screens and more. The web applications of the demos contain various examples of common tasks needed to be performed in a typical web enabling/SOA application (in .NET or JSP). Such examples include external login page, path procedures, and tables combining a few screens to one.



Note: When working in Windows OS, by default the repository is loaded from the database provided with ApplinX. When working with UNIX, the repository of the demo application is loaded from an external gxz file and therefore changes cannot be made to entities (read-only).

The Insurance applications (Instant, Composite and SOA) manage customer details and proposals of a fictitious insurance company. They store and display information about the current customers and their proposal details. The applications contain modules for updating, deleting, and displaying customer records and proposals. There are three Insurance based applications:

- The Instant demo application includes the entities required for an instant web application.
- The Composite demo application includes, in addition to the entities included in the instant demo, paths and procedures used in the composite web application.
- The Service Oriented Architecture (SOA) demo application includes the entities required to expose Procedure Clients and forms that have been generated.

The ApplinX web demo applications are distributed as part of the ApplinX installation. They contain many useful examples and code snippets, and therefore it is recommended to install them, browse through them and inspect their sources.

Installing the Demo Web Application using the Software AG Designer

Refer to *Creating an ApplinX Web Application using the Software AG Designer* and select the desired application type.

Installing the Demo Web Application using Tomcat

➤ To install the demo web application

- 1 Right-click on the relevant application (make sure that it is one of the demo applications: Instant, Composite or SOA) and select **Web Application Manager...**
- 2 Ensure that **Create a new Web application** is selected. Click **Next**. The **Create a New Web Application** screen is displayed.

- 3 Ensure that **JSP** is selected as the language.
- 4 Select the relevant demo name as the **Application type**. Click **Next**.
- 5 Enter a name for the web application.
- 6 Browse and select the folder where Tomcat is installed.
- 7 Select the location of the JDK folder.
- 8 Click **Next**. The **Wizard Summary** screen is displayed.
- 9 Select whether to launch the browser when the process is completed.
- 10 Select whether to restart Tomcat.
- 11 Click **Finish**. A message will be displayed indicating whether the process succeeded or failed.

Installing the Demo Web Application (Other Java Application Servers)

» To install the demo web application

- 1 Right-click on the relevant application (make sure that it is one of the demo applications: Instant, Composite or SOA) and select **Deployment Manager for J2EE...**
- 2 Select **ApplinX for JSP Web application** and click **Next**. The **Select ApplinX Web Application** screen is displayed.
- 3 Select **Project in other location**. Browse and select *<ApplinX installation>\ApplinX\new jsp*<demo application name>**.
- 4 Select **Include source files** to include the Java files as well as the compiled classes. Click **Next**. The **Select Package Parameters** screen is displayed.
- 5 Select the Java Application Server, the WAR file name and the destination folder where the output file will be placed.
- 6 Click **Next**. The **Select Target ApplinX Server** screen is displayed.
- 7 Select the name/IP address of the server, the port number and the ApplinX application name. Click **Finish**.
- 8 In the console area, it is possible to see whether the process succeeded or failed. The WAR file created should be placed in your web server directory.

Installing the Demo Web Application using Visual Studio

➤ **To install the sample application in Visual Studio**

- 1 Open Visual Studio and open a **New Web Site**.
- 2 In the **New Web Site** dialog box make sure Visual C# is the selected language.
- 3 Choose one of the following templates:
 - ApplinX C# Composite Demo Application.
 - ApplinX C# Instant Demo Application.
 - ApplinX C# SOA Demo Application.
- 4 Right-click *index.aspx* and select **Set as Start Page**.
- 5 Click **Start** to run the project.

13 Working with the Demo Applications

- The Instant Demo 78
- The Composite Demo 86
- The SOA Demo 92

The Instant Demo

- [template.jsp/template.master](#)
- Screen Name: Login
- Screen Name: InsuranceMenu
- Screen Name: BrowseCustomers



Note: When using a replay file, it is possible to navigate from one screen to the next by pressing ENTER in each web screen. Clicking on buttons which execute paths will work only if the replay file contains the relevant path navigation recorded and the session screen is in the relevant starting point to execute the path. If this is not the case, the path will fail and the user will receive an error from the web application.

When using transformations it is important to notice the order in which the transformations appear in the screen editor (**Transformations** tab). Transformations can change the way ApplinX perceives a host screen. If, for example, the first transformation hides a certain text on the screen and the second transformation searches for that text pattern in order to replace it with a link then the second transformation will never find a match for that pattern since it has been removed by the first transformation. So always keep in mind, the order of the transformations in the screen editor will be the order in which they will be executed.

template.jsp/template.master

Refer to *Customizing the Default Template* in the *Web Application Development* documentation for a detailed explanation on how to change the default template.

- The host keys relevant for this screen are displayed on the left of the screen. This is defined in the *template.jsp/template.master* (.NET web apps) file in the host key control. The host key control displays all the host keys for the current host screen. It is possible to define whether they should appear horizontally or vertically, in the form of buttons, links or a pre-defined template (see "gx:GXHostKeysControl" in .NET web apps or "gx:hostKeys" tag in JSP applications). The default host key style is defined in the *css/styles_instant.css* style sheet in the "gx_hky" class.
- The Page footer section contains links that perform various actions such as logging off, refreshing the page, submitting a terminal key and so on. These links can be defined in the *template.jsp/template.master* file (look for "pgfooter" control in the instant demo app).

Screen Name: Login**Host Screen**

```
13:32:54      TID   14      -DEMOCO-      User      30.12.08
                                     ULGO
-----  COM-LETE  System Logon  -----

      For hints on how to change your password, please read:

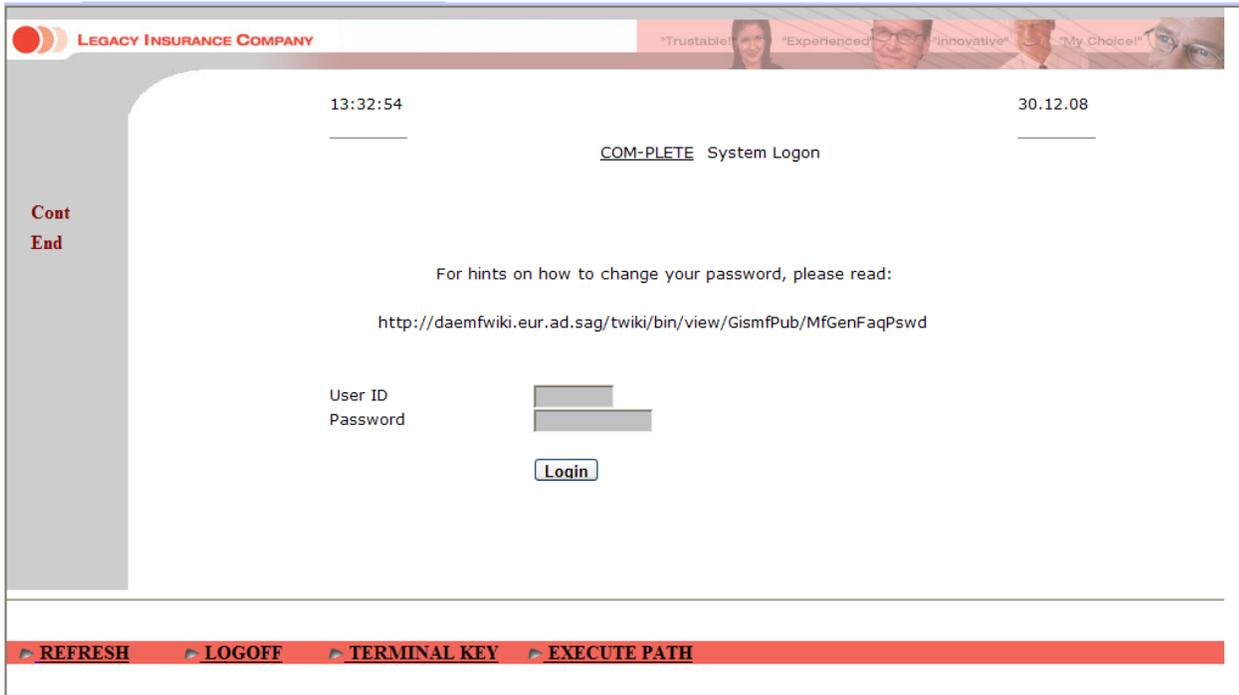
      http://daemfwiki.eur.ad.sag/twiki/bin/view/GismfPub/MfGenFaqPswd

User ID .....: 
Password .....: _____
New password .....: _____
Group (RACF) .....: _____

Session reconnect..: X

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Cont          End
```

Screen in Web Application



 **Note:** The sections that are referred to below can be found in the *Designing and Developing an Application* documentation.

- Identify the screen using the "-DEMOCO-" and "COM-LETE System Logon" texts. Refer to *Define Identifiers*.
- Map the **Message**, **UserID**, **Password** and **Reconnect** fields. Refer to *Map Fields*.
- The "HideTextTODeI" transformation hides text in a defined area. Refer to *Transforming a Text Pattern to Text* and select the "Hide" option.



- The "RemoveDots" transformation removes dots followed by a colon ("...:"). Refer to *Transforming a Text Pattern to Text*.
- The "HideInputField" transformation hides input fields in a defined area. Refer to *Transforming an Input Field to a Text Field* and select the "Hide" option.
- The "HideText" transformation hides text in different areas in the screen. Refer to *Transforming a Text Pattern to Text* and select the "Hide" option.

- The "DashesToLine" transformation changes all the dashes in the screen to lines. Refer to *Transforming a Repeating Characters Pattern to a Line*.
- The "AddLoginButton" transformation adds a button used to login. Clicking on this button executes the "login" path procedure. Refer to *Transforming a Text Pattern to a Button*.

The "login" path procedure receives the user ID and password as inputs.

Screen Name: InsuranceMenu

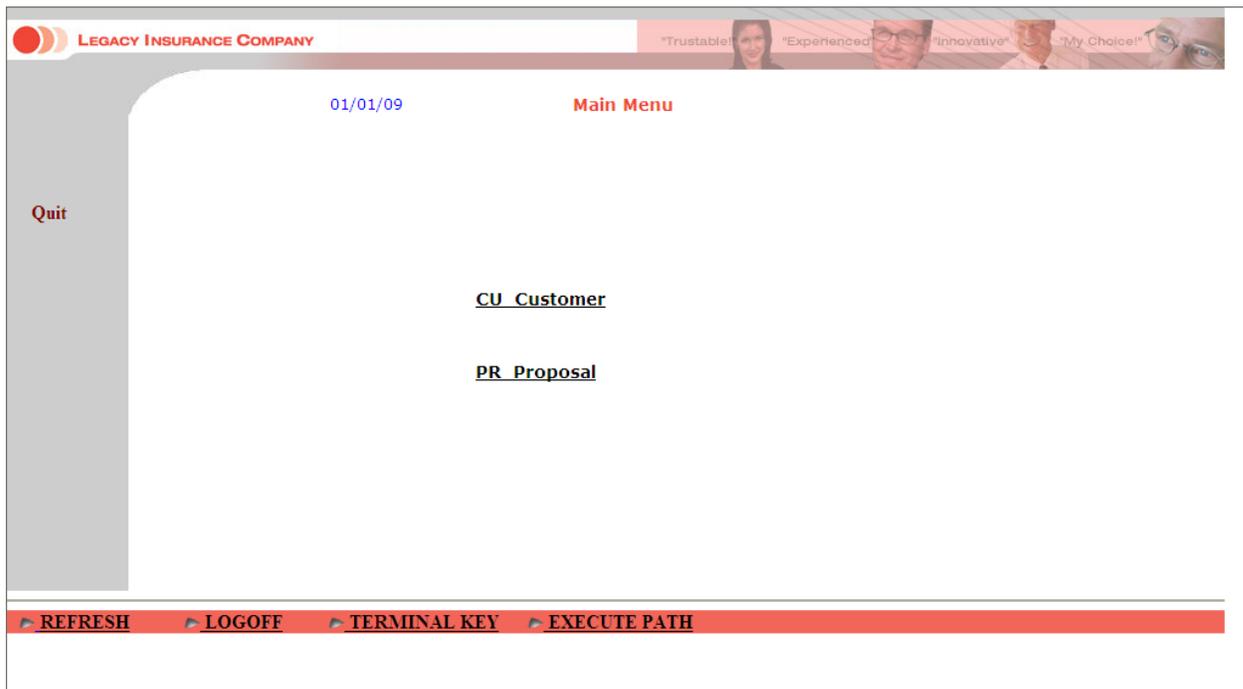
Host Screen

```
DEMO          ***** Demo Insurance Solution *****          DEMOM1
01/01/09      - Main Menu -                                     12:53:36

Code Description
-----
CU Customer
PR Proposal
-----
Code: 

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Quit
```

Screen in Web Application



 **Note:** The sections that are referred to below can be found in the *Designing and Developing an Application* documentation.

- Identify the screen using the "DEMO" and "DEMOM" texts. Refer to *Define Identifiers*.
- Map the **Code**, **Time** and **Message** fields. Refer to *Map Fields*.
- The "CleanMenuScreen" transformation hides text in a number of defined areas in the screen. Refer to *Transforming a Text Pattern to Text* and select the "Hide" option.
- The "AllGroup" screen group includes all the screens that include in row 1, column 23 the content "*****Demo Insurance Solution*****". The "AllGroup" screen group includes various transformations:
 - Removing the first line in the screen (the "RemoveFirstLine" transformation). Refer to *Transforming a Text Pattern to Text* and select the "Hide" option.
 - Formatting the screen name (the "FormatScreenNameHeader" transformation).
 - Removing dots that are followed by a colon "...:" (the "RemoveDots" transformation). Refer to *Transforming a Text Pattern to Text*.
 - Hiding the time that appears in the top right corner (the "HideText" transformation). Refer to *Transforming a Text Pattern to Text* and select the "Hide" option.
- The "MenusGroup" screen group includes all screens that include text: "code", and text "menu". This screen group includes a transformation which transforms menus to links (the "MenuToLinks" transformation). Refer to *Transforming a Menu to Hyperlinks*.

- Associate the "AllGroup" and "MenusGroup" screen groups to this screen.
- Refer to the template section above to see how to implement placing the host keys on the left of the screen, and placing buttons in the footer.

Screen Name: BrowseCustomers

Host Screen

```

BCUSTENO          ***** Demo Insurance Solution *****          BCUSTM01
13/07/08          - Browse Customers -                               14:40:25
                                                                1/2

```

A	Customer ID	Lastname	Firstname	Birthdate	Sex	Type
18		van de Wetering	Marco	1931-02-12	M	I
2		van Bosch	Lea	1988-06-21	F	I
22		von Grueningen	Robert	1930-08-09	M	I
1		Bachmann	Daniela	1960-03-04	F	I
3		Bloggs	Joseph	1974-04-04	M	I
4		Busley	Matthew	1920-08-20	M	I
27		Chappell	Charles	1989-04-16	M	I
6		Copperdale	Alice	1990-05-05	F	I
5		Dillenburg	Mike	1960-12-08	M	I
9		Dragos	Istvan	1911-04-19	M	I
101		Eisenstein	Armin	1969-03-14	M	I
14		Grass	Mandy	1969-07-01	F	I
16		Higgins	Felix	1980-03-11	M	I

```

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Quit Add           Up   Down   <   >

```

Screen in Web Application

04/01/09 **Browse Customers** 1/2

ID: Lastname: Firstname:

A Customer Birthday Sex Type

.CHOOSE-	van de Wetering	Marco	1931-02-12	M	I
.CHOOSE-	van Bosch	Lea	1988-06-21	F	I
.CHOOSE-	von Grueningen	Robert	1930-08-09	M	I
.CHOOSE-	Bachmann	Daniela	1960-03-04	F	I
.CHOOSE-	Bloggs	Joseph	1974-04-04	M	I
.CHOOSE-	Busley	Matthew	1920-08-20	M	I
.CHOOSE-	Chappell	Charles	1989-04-16	M	I
.CHOOSE-	Copperdale	Alice	1990-05-05	F	I
.CHOOSE-	Dillenburg	Mike	1960-12-08	M	I
.CHOOSE-	Dragos	Istvan	1911-04-19	M	I
.CHOOSE-	Eisenstein	Armin	1969-03-14	M	I
.CHOOSE-	Grass	Mandy	1969-07-01	F	I
.CHOOSE-	Higgins	Felix	1980-03-11	M	I

REFRESH LOGOFF TERMINAL KEY EXECUTE PATH

- Identify the screen using the "BCUSTBN0" and "BCUSTBN1" texts. Refer to *Define Identifiers* in the *Designing and Developing an Application* documentation.
- Refer to the Fields tab to see the mapped fields.
- Refer to the template section above to see how to implement placing the host keys on the left of the screen, and placing buttons in the footer.

The Composite Demo

- [template.jsp/template.master](#)
- Screen Name: Login
- Screen Name: Browse Proposals
- Screen Name: ModifyProposal1
- Screen Name: BrowseCustomers1



Note: When using a replay file, it is possible to navigate from one screen to the next by pressing ENTER in each web screen. Clicking on buttons which execute paths will work only if the replay file contains the relevant path navigation recorded and the session screen is in the relevant starting point to execute the path. If this is not the case, the path will fail and the user will receive an error from the web application.

template.jsp/template.master

Refer to *Customizing the Default Template* in the *Web Application Development* documentation for a detailed explanation on how to change the default template.

JSP

In this template we use these panels (that can be overridden by specific pages):

- **CssPanel**: holds the css links (has a default in the template).
- **JSPanel**: holds the JavaScript code and JavaScript file references.
- **HeaderImagePanel**: by design, an image in the top-right corner changes according to the data.
- **MenuPanel**: holds the menu that is included in the pages. In some specific pages it is empty, therefore the menu is not displayed.
- **GXPagePlaceholder**: holds the data for the page.

.NET

In this template we use these panels (that can be overridden by specific pages):

- **CssPanel**: holds the css links (has a default in the template).
- **JSPanel**: holds the JavaScript code and JavaScript file references.
- **HeaderImagePanel**: by design, an image in the top-right corner changes according to the data.
- **MenuPanel**: holds the menu that is included in the pages. In some specific pages it is empty, therefore the menu is not displayed.
- **GXPagePlaceholder**: holds the data for the page.

Screen Name: Login

LEGACY INSURANCE COMPANY "Trustable!" "Experienced" "Innovative" "My Choice!"

Login

Please Enter your UserName and Password:

User ID: Password: [LOGIN](#) [AUTOLOGIN](#) Work online

When not connected to the Software AG network, the demo application will run in offline mode (using a pre-recorded GCT file). In this mode, some of the web application functionality may not perform as it would in the actual online Web Application.

After logging in, use the ENTER key to proceed from one screen to another. Refer to the "**ApplinX Demo Applications**" chapter in the documentation to learn more about the different tasks implemented in each page

Refresh | Logoff

Software AG powered by **ApplinX**

INTRANET

Tasks

Refer to the following sections in the *Web Application Development* documentation:

- *Building an External Login Page*
- *Controlling the Connection Properties from Code*

Screen Name: Browse Proposals

LEGACY INSURANCE COMPANY "Trustable" "Experienced" "Innovative" "My Choice!"

Select Proposal

Search by: Proposal ID: OR Status: **SEARCH**

Status	Proposal ID	Product	Start Date	Policy Holder Name	Total Premium
RECEIVED	1	HOUSE	2004-05-01	von Grueningen,Rober	36.30
REQUEST	2	HOUSE	2004-02-03	Kantor,Edgar	176.04
ACTIVE	3	HOUSE	2003-11-02	van Bosch,Lea	56.91
IN-USE	4	HOUSE	2004-08-01	Chappell,Charles	58.91
RECEIVED	5	HOUSE	2001-01-01	Rowling,Keith	20.38
RECEIVED	6	HOUSE	1980-10-01	Rowling,Keith	213.26
IN-USE	7	HOUSE	2000-01-01	Dragos,Istvan	17.22
ACTIVE	8	HOUSE	2003-04-01	Krasnenko,Wladimir	161.61
REQUEST	9	HOUSE	1992-02-01	Grass,Mandy	293.53
RECEIVED	10	HOUSE	2004-02-01	Squirtle,Connor	68.68
ACTIVE	11	HOUSE	1989-10-01	Higgins,Felix	127.85
REQUEST	12	HOUSE	2005-04-01	Miller,John	198.03
REQUEST	13	HOUSE	2005-05-05	Orlikowski,Adam	3.00

Software AG powered by **Applinx**

Refresh | Logoff

INTRANET

Tasks

Refer to the following sections in the *Web Application Development* documentation:

- *Activating an Application Map from a Menu*
- *Creating Designed Web Pages*
- *Creating a Button / Hyperlink for Submitting a Host Key*
- *Using Web Application Controls in Generated Pages*
- *Exporting Data to an MS Office Application (Excel, Word)*
- *Adding the Sorting Capability to a Screen-Based Table*
- *Creating a Page with a Table*

Screen Name: ModifyProposal1

LEGACY INSURANCE COMPANY "Trustable!" "Experienced" "Innovative" "My Choice!"

Detail view proposal

Proposals
Customers
Log Out

Proposal ID: 3

Main Data | Collection | Object Data | Risk Data

Status Code:	ACTIVE	Total Premium:	
Product Code:	HOUSE	<input type="checkbox"/> External Calc	
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:	
Agent ID:	19	Agent Name:	

Software AG
powered by
Applix

Refresh | Logoff

INTRANET

Tasks

Refer to the following sections in the *Web Application Development* documentation:

- *Collecting Data from Multiple Host Screens*
- *Creating a Button / Hyperlink for Executing a Path Procedure*
- *Updating Data in Multiple Host Screens*
- *Activating an Application Map from a Menu*
- *Creating Designed Web Pages*
- *Creating a Button / Hyperlink for Submitting a Host Key*
- *Using Web Application Controls in Generated Pages*
- *Map Fields* (identify dynamic fields by their leading label)

Screen Name: BrowseCustomers1

The screenshot displays the 'Customer account' section of the Legacy Insurance Company web application. It features a search form with fields for Customer ID, Last Name, and First Name, and a 'SEARCH' button. Below the form is a table listing customer information. The table has columns for Customer ID, Last name, First name, Birthday, Gender, Type, City, and Street. The data rows are as follows:

Customer ID	Last name	First name	Birthday	Gender	Type	City	Street
22	von Grueningen	Robert	1930-08-09	M	I	Berlin, 10117, D	10 Unter d. Linden
27	Chappell	Charles	1989-04-16	M	I	London, S6C 3T2, GB	12 Cromwell Street
3	Bloggs	Joseph	1974-04-04	M	I	Manchester, M24 4AD, GB	123 Deansgate
4	Busley	Matthew	1920-08-20	M	I	Manchester, AF3 D3C, GB	97 Trafford Lane
5	Dillenburg	Mike	1960-12-08	M	I	New York, NY 10771, USA	8023 Broadway
6	Copperdale	Alice	1990-05-05	F	I	Toronto, M3J 1P3, CAN	603 W 22nd Street
66	Moreaux	Mireille	1957-23-03	F	I	Paris, 75001, F	122A Champs Elysees
7	Priester	Patrick	1980-03-22	M	I	Darmstadt, 64297, D	12 Uhlandstr.
8	Kubrey	Stanley	1928-07-26	M	I	Weiterstadt, 64331, D	24 Mainzer Str.
9	Dragos	Istvan	1911-04-19	M	I	Budapest, 1071, H	171 Balaton Street

The interface also includes a navigation menu with 'Proposals', 'Customers', and 'Log Out' options. At the bottom, there are 'Refresh' and 'Logoff' buttons, and a status bar with 'INTRANET' and several icons.

Tasks

Refer to the following sections in the *Web Application Development* documentation:

- *Activating an Application Map from a Menu*
- *Creating Designed Web Pages*
- *Collecting Data from Multiple Host Screens*
- *Creating a Button / Hyperlink for Submitting a Host Key*
- *Using Web Application Controls in Generated Pages*
- *Adding the Sorting Capability to a Screen-Based Table*
- *Exporting Data to an MS Office Application (Excel, Word)*
- *Customizing the Table's Display*

The SOA Demo

.NET and JSP implementation can be done in many ways. This demo includes examples of one way to implement the tasks. For example, in one of the tasks below, we demonstrate how to display data collected by the procedure client. To do this we use the framework. In .NET, this can also be implemented using .NET datagrid.

The following lists the different typical tasks that we have demonstrated in the SOA application included with the ApplinX installation. Next to each task, the entities involved with this task are detailed. This enables you to further explore the configuration of each of these entities.

1. Identify and define host screens in the ApplinX repository. Refer to *Screens* in the *Designing and Developing an Application* documentation.
 - INS_Browse_Customers
 - INS_Customer_Details_1
 - INS_Customer_Details_2
 - INS_Customer_Details_3
 - DEMOCO_Login
 - DEMOCO_Menu
 - DEMOCO_Splash
 - E_Environment
 - INS_Menu
2. Create a connection pool with a connection information set (for offline use). Refer to *Connection Pools* and *Connection Information Sets* in the *Designing and Developing an Application* documentation.
 - a. Set the pool size as required.
 - b. Set a connection information set when required (SOADemo includes connection information sets that allow the application to run in offline mode).
 - c. Set navigation rules:
 - i. Create an initialization path for the pool that initializes each new connection in the pool. This path should finish in the "Initial screen", otherwise the initialization will fail.
 - ii. Set the initial screen a connection should wait on.
 - iii. Create and set a recycle path to recycle connections in the pool.
 - iv. Create and set a termination path for properly terminating a connection in the pool.

As you may have noticed, the SOADemo repository contains three connection pools. This is for the sole reason of enabling the application to retrieve results in offline environments (that do not have connectivity to Software AG's network). All three pools share the same setting

- (pool size and navigation settings), but differ in their connection information sets. To enable offline execution of the application, each procedure has been executed with its own replay file (set in its connection information set), whereas, online execution requires no replay file at all.
3. Collect data from multiple screens. Refer to *Collecting Data from Multiple Host Screens* in the *Web Application Development* documentation.
 - a. Create a path procedure which collects table data (refer to the *Browse_All_Customers* path procedure). Refer to *Creating a Path Procedure* in the *Designing and Developing an Application* documentation.
 - i. Assign a connection pool to start from the relevant screen.
 - ii. Create a While loop that checks if an "End of data" message appears on the screen, marking the table's end.
 - iii. For every table instance: Collect the tabular host screen data and page down for the next instance.
 - b. Create a path procedure which collect data from multiple screens (refer to the *GetCustomer-ByID* path procedure). Refer to *Creating a Path Procedure* in the *Designing and Developing an Application* documentation.
 - i. Assign a connection pool to start from the relevant screen.
 - ii. Add an input to allow the user to specify which customer details they wish to retrieve.
 - iii. Navigate to the Details screens and map host fields to the procedures' output.
 - c. Assign procedures to procedure groups. Refer to *Assigning a Procedure to a Procedure Group* in the *Designing and Developing an Application* documentation.
 - d. Generate a procedure client from the procedure group (refer to *DemoINS* procedure group in the demo repository). Refer to *Procedure Clients* in the *Designing and Developing an Application* documentation.
 4. Bind web controls to procedure response (output). Refer to *Binding Procedure Outputs to an ApplinX Framework Based Web Page* in the *Designing and Developing an Application* documentation.
 - a. For each procedure, create a JSP/ASPX page with controls which match the procedure output.
 - b. Fill the form (using `gx_fillForm`).

14

Self-Training and Movies

A self-training, basic tutorial of ApplinX and accompanying movies can be found on the Software AG Empower website at <https://empower.softwareag.com>. After entering your user name and password, navigate to **Knowledge Center > Product Documentation > webMethods Product Line > ApplinX > Self-Training**.

