

ApplinX Legacy Integration Suite[®]

A Technical Overview White Paper

Last revised: November 2003

© 2003 by Sabratec Ltd. All rights reserved.

Distribution of this document requires explicit permission from Sabratec Ltd.

Brand and product names in this document are trademarks or registered trademarks of their respective holders.

1. Overview

ApplinX Legacy Integration Suite[®] is an innovative solution for rapid integration of legacy processes into modern web applications and Web Services.

With ApplinX, legacy business logic and related data can be automatically encapsulated or extended into Web Services, any standard framework object, or any standard presentation file in a low-risk and reliable approach without the need to alter any legacy asset or to write a single line of code.

Using ApplinX's point-and-click composing tools, you can easily re-engineer legacy flows into new flow paradigms of your choice and map legacy entities into modern business processes.

ApplinX supports all common legacy hosts (IBM Mainframes, CICS, iSeries, Unix VT, Unisys, and more) and all common legacy non-invasive interfaces (direct transaction activation, direct program invocation, CICS COMMAREA, block- and character-based screen flows, printer buffers, and more).

ApplinX strictly adheres to industry standards and object models. ApplinX's open architecture gives you the flexibility to choose your standard target framework, development environment, composite object types or presentation file type (Web Services, .Net, J2EE, COM Objects, Java Beans, ASP, JSP, and others). Furthermore, ApplinX is provided with ready-made adapters for instant integration with commercial packages, Enterprise Portals, and EAI products.

As a result of ApplinX's unique features, integration projects can be completed in a matter of days or weeks, resulting in a robust, optimized, and easy-to-maintain system.

The general concept of ApplinX is illustrated in the following diagram:



2. ApplinX Modules & Architecture

2.1 ApplinX Modules

ApplinX Legacy Integration Suite's modules cover all phases of an integration lifecycle, from design to run-time processing.

With ApplinX, one suite of modules serves all types of legacy integration. Thus, saving the need to learn multiple tools and assuring a completely synchronized integration scenario. Such a scenario can include different integration methods combined together into a unified business process.

The following is a description of the various modules:

ApplinX Composer[®]

ApplinX Composer is a point-and-click easy-to-use design tool. It provides a rich set of flexible design facilities for a rapid construction of integration logic. Many of ApplinX Composer's tasks are performed automatically thus saving developers considerable time and effort.

With ApplinX Composer, developers can rapidly define and model integration scenarios, identify and map legacy entities (screens, fields, parameters, programs, etc.), and compose flow transformations using a sophisticated graphic visualizer.

ApplinX Composer allows a visual navigation through host screens (paths) and recording of such navigation for a later playback. Screen images can be saved and later be checked and synchronized against current session screens. ApplinX Composer is also responsible for the automatic generation of the corresponding Web Services and composite objects that encapsulate host entities.

ApplinX Composer allows a flexibility of two working modes for screens navigation: an online mode against a live host session or a replay mode for an offline development using a previously recorded host session.

ApplinX Composer is implemented as a Java utility and can run either as a Java Applet (within an internet browser) or as a stand-alone application.



ApplinX Server[®]

ApplinX Server is a cross-platform, multi-threaded, and scalable runtime server. It manages real-time requests from multiple users, composite objects, and Web Services wishing to access host legacy applications under specified flow conversions.

ApplinX Server includes host connectors, a load balancing mechanism, a user session manager, a Web Services operating environment, and an interface to generated composite objects.

ApplinX Server can run on any Java-enabled operation system (Microsoft Windows, Linux, Sun Solaris, HP/UX, IBM OS/400, IBM AIX, Novell Netware, etc.) and on any J2EE Application Server/container (Oracle9iAS, IBM WebSphere, BEA Weblogic, etc.).

ApplinX Repository®

ApplinX Repository is a collection of application knowledge attributes. It serves as a normalizing tier between the legacy applications, on one side, and the modern Web Services and web objects, on the other side.

The repository holds entities such as screen identifiers, field mappings, context-based navigation rules, flow conversion rules, and services definitions. Entities in the repository are defined and maintained using ApplinX Composer and are accessed by ApplinX Server at run-time.

ApplinX Repository can reside in any popular database including Microsoft Access, SQL Server, Oracle, DB2, or any other JDBC/ODBC compliant database.

ApplinX Administrator®

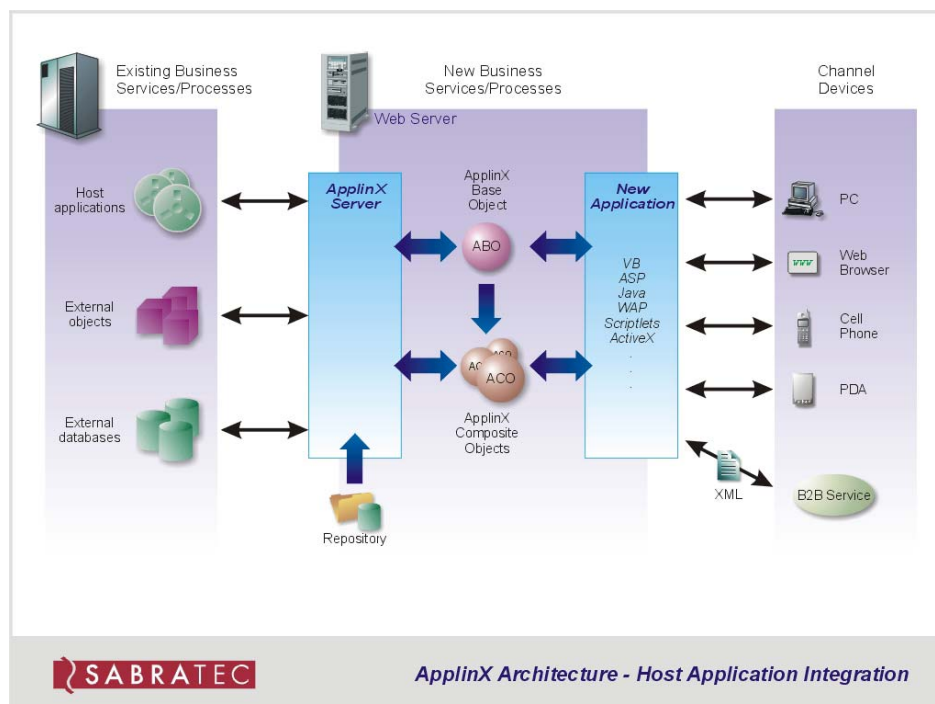
ApplinX Administrator is a server administration and management tool. Using ApplinX Administrator, you can define, monitor, and control server activities and configuration.

ApplinX Administrator schedules initiation of services, assigns priorities, and facilitates an orderly operation of a multi-user environment.

With ApplinX Administrator, it is possible to monitor traffic and to track the level of utilization of different server components in order to ensure scalability of server environment in accordance with growth in server activity.

2.2 ApplinX Architecture

ApplinX's open architecture supports a flexible set of configurations in order to adapt to any deployment need. This architecture is illustrated in the following diagram:

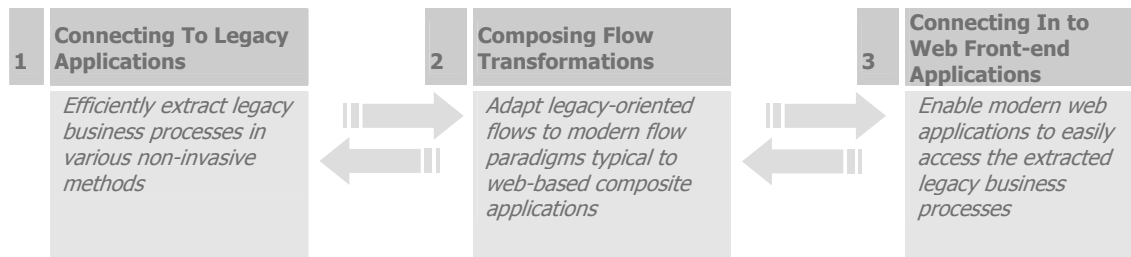


ApplinX Server uses the TCP/IP protocol to communicate with both host and web environments.

For convenient interfacing, communication between ApplinX Server and the generated Web front-end components is routed via a distributed object interface called **ApplinX Base Object (ABO)**. The ABO simplifies communication by hiding server's programmatic handshake complexities. All ApplinX's modules and communications provide multilingual support and allow on-the-fly format conversions.

2.3 ApplinX Integration Scenario

The various aspects of a typical legacy integration project can be grouped into the following three methodological tasks:



These methodological tasks are described in detail in the following chapters.

3. Connecting to Legacy Applications

ApplinX’s interaction with the legacy applications supports all common legacy hosts and legacy non-invasive interfaces.

The following table depicts the various interaction options:

Legacy Hosts	Legacy Interfaces	Screen Connectors	Direct Transaction/Program Connectors
<ul style="list-style-type: none"> ▪ IBM Mainframe ▪ IBM iSeries ▪ Unix ▪ VMS ▪ HP e3000 ▪ Unisys ▪ Fujitsu ▪ Hitachi ▪ Tandem 	<ul style="list-style-type: none"> ▪ Screen Flows ▪ Direct Transaction Interface ▪ Direct Program Interface ▪ Conversational / Non-Conversational / Pseudo-Conversational Transactions ▪ Printer Buffers ▪ <i>Any mapped or unmapped streams</i> ▪ <i>Any TCP/IP-based formatable buffer</i> 	<ul style="list-style-type: none"> ▪ 3270 ▪ 5250 ▪ VT ▪ Block Mode ▪ Character Mode ▪ Mapped and unmapped 	<ul style="list-style-type: none"> ▪ CICS ▪ COBOL ▪ ADABAS/Natural ▪ RPG ▪ APPC

ApplinX’s interaction with the legacy host includes the following unique features:

3.1 Direct Transaction/Program Connectors & Automatic Parameter Mapping

ApplinX provides a comprehensive solution for direct invocation of legacy transactions and programs without the need to modify or rewrite them.

Using a variety of ready-made utilities, legacy program’s parameter definitions; such as *COBOL/COMMAREA*, *ADABAS/Natural Maps*, and *RPG*; are automatically analyzed and mapped at design time. Using these mappings, ApplinX allows full as well as partial wrappings of these programs as Web Services. These programs are then invoked directly by ApplinX at runtime.

A special connector is provided for invoking *CICS* transaction-based programs. Such programs should be IBM DPL-enabled (designed to be invoked by EXEC CICS LINK) with no presentation layer.

3.2 Host Connectivity Optimization

For assuring an efficient execution of host sessions, ApplinX includes an advanced array of connectivity optimization and monitoring features.

Connectivity optimization is handled for each Host Service. A Host Service is a packaged collection of host legacy business processes with a certain flow ([see details below](#)). Host Service optimization definitions include flexible connection pooling, stand-by threshold values, and recovery options.

Of special interest is ApplinX’s “fast-forward” initialization process, activated for each established Host Service session. This process brings the session up to the first significant navigation point. This unique feature guarantees an optimal response time when a selected Host Service session is invoked.

These capabilities ensure an optimized host access, optimal response times, an efficient connectivity resource management, a reliable performance, and automatic process recovery.

3.3 Screen Identification and Field Mapping

Using ApplinX Composer, screens and fields are identified and mapped, both manually and automatically, employing an extensive set of advanced identification algorithms. Identifiers include specific text in a certain area of a screen, screen cursor position, and various field attributes.

For character-based interaction (e.g. VT), special *wait-condition* triggers are available to identify logical screen frameworks.

3.4 Mapping of Fields to GUI Elements

Legacy application's input fields can be mapped automatically into corresponding front-end GUI elements such as combo boxes, checkboxes, and radio buttons.

3.5 Connection Keep-Alive

A Keep-Alive mechanism is provided to avoid host servers from discarding connections after a period of idle time. This is especially useful for maintaining service connection pool readiness at stand-by.

ApplinX allows configuring a Keep-Alive timeout after which a Keep-Alive Path is activated. Such a path is responsible for keeping the connection available and within the same context and host screen.

3.6 Exception Handling

When an unexpected behavior in legacy application's side is traced, an exception flag is turned on. This triggers an execution of a pre-defined exception handling based on the context of exception (e.g. presented screen).

This event-driven mechanism is especially useful in cases where the host server prompts a user with a bursting message screen.

3.7 Automatic Legacy Change Management

Integrated legacy applications might be modified with time. These modifications could affect screen structures and other interfaced legacy objects. Most frequently, such modifications are carried out, without a proper notice, by development groups other than those who are responsible for integration. As a result, existing legacy integration processes may fail to function properly upon legacy modifications.

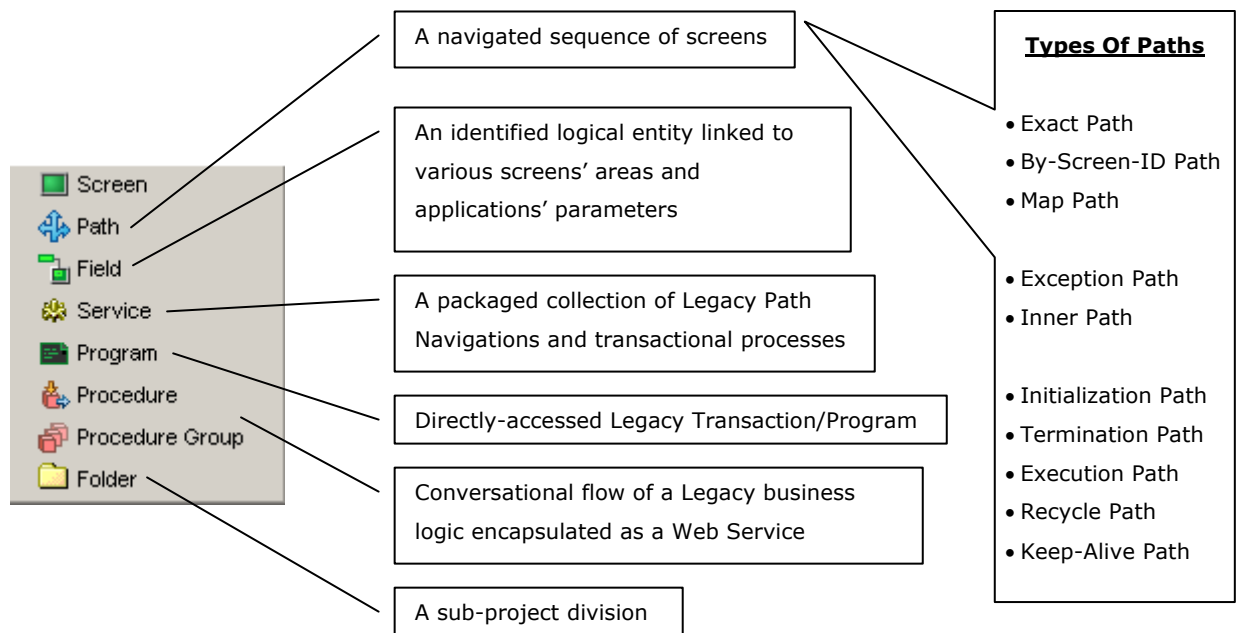
To avoid such cases, ApplinX provides an automatic legacy change management facility. Using this facility, legacy screen images are saved in ApplinX Repository and later are checked and synchronized automatically against current session screens for possible traced mismatches.

4. Composing Flow Transformations

A modern Web application flow presents a contradiction to a host's hierarchical screen structure and transactional program interaction.

To allow a new Web flow based on a transformation and enhancement of a legacy flow, ApplinX provides flow modeling facilities. These facilities map and then make use of legacy hierarchical screen navigations and transaction/program entities.

ApplinX allows a flexible identification and manipulation of the various navigation entities. These are illustrated in the following diagram:



4.1 Navigation Paths

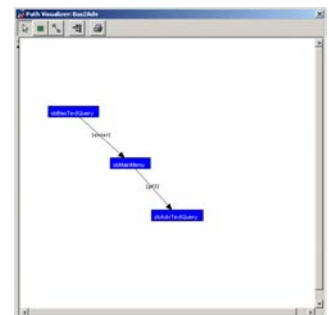
A path's primary function is to break traditional legacy flows, based on rigid screen navigations, into modern task-oriented flow patterns.

A path allows the gathering of information from several screens into one flow scenario; it represents a transaction / "closed" process / navigation rule in which you enter data, move through screen sequences, and retrieve new data.

Each path consists of **Steps**. Each Step may contain Input, Wait, and **Flow** actions.

Flow actions allow conversational interaction, conditions, testing, and data retrieval.

Using **ApplinX Path Visualizer**, paths can be easily defined and managed.



4.2 Host Services

ApplinX Host Services provide a flexible packaging of new composite flows. This packaging is built on top of pre-defined host navigation entities such as paths and screen definitions.

A host service definition may contain an **Initial Screen**, an **Initialization Path**, a **Termination Path**, an **Execution Path**, a **Recycle Path**, and service connectivity specifications for an optimized execution ([see details above](#)).

4.3 Procedures and Procedure Groups as Web Services

A Procedure is a well-defined business-logic (with input and output arguments). A Procedure Group is a container of Procedures. Procedure Groups are exposed as Web Services. Procedures invoke host services, paths, and legacy transactions/programs under various specified conditions.

5. Connecting into Web Front-End Applications

In order to integrate with front-end web applications, ApplinX automatically generates web front-end components.

These components can be standard Web Services, web presentation code, standard objects (called **ApplinX Composite Objects – ACO**), or ready-made adapters for instant integration with popular commercial packages, Enterprise Portals, and EAI products.

All common web development frameworks and tools are supported as front-end environments.

The following table depicts ApplinX’s generated front-end component types and related frameworks, supported development tools, and supported web application servers:

Generated Component	Front-End Framework *	Development Tool **	Target Web Application Servers
<ul style="list-style-type: none"> ▪ Web Service ▪ .Net Component ▪ COM Object ▪ Java Bean ▪ ASP file ▪ ASPx file ▪ JSP file ▪ XML Document ▪ XML data-binding object ▪ WAP/WML file 	<ul style="list-style-type: none"> ▪ J2EE ▪ .Net 	<ul style="list-style-type: none"> ▪ Visual Basic ▪ Visual Studio .Net ▪ Borland JBuilder ▪ Oracle JDeveloper ▪ IBM WebSphere Studio Application Developer ▪ Any Java IDE 	<ul style="list-style-type: none"> ▪ Microsoft IIS + MTS ▪ IBM WebSphere AS ▪ Oracle9iAS ▪ BEA Weblogic ▪ Any J2EE or Java/JSP Compliant Application Server ▪ Any .Net or COM/ASP Compliant Application Server

* A working framework is provided with ApplinX Server.

** An ApplinX ‘Legacy Connection’ add-in is provided.

The following table lists the ready-made adapters provided by ApplinX for instant integration with popular products on the front-end side:

EAI Products	Magic iBolt, Microsoft BizTalk, IBM WebSphere MQ, Webmethods, Microsoft Host Integration Server
Enterprise Portals	Microsoft Sharepoint Portal, Oracle9iAS Portal, Plumtree Corporate Portal, CA CleverPath Portal, Magic iBolt Portal, IBM WebSphere Portal, SAP Portal
Commercial Packages	Peoplesoft CRM, Amdocs ClarifyCRM



An ApplinX ‘Legacy Connection’ add-in is provided for each supported development tool. This add-in allows an instant exposure of ApplinX-generated Web Services within tool’s environment.

The generated components serve as advanced building blocks for a new modern application, while the ApplinX Server handles their communication with the legacy environment and performs the flow transformation.

This way, a totally new Web application can be easily created from several host applications and various resources such as databases and external objects. The

object model ensures that the new application can be implemented on any desired "channel device", such as desktop PC, Web browser, cell-phone, PDA, etc.

6. Further Readings

ApplinX documentation resources include an extensive and continuously updated set of technical as well as business White Papers, Brochures, Data Sheets, Reference Manuals, and Guides.

Documentation resources can be easily obtained either by accessing Sabratec's web site (www.sabratec.com) or by contacting Sabratec personnel or an authorized distributor.