

Integrate^{IT} Aspect Integrator PlatformTM

Version 2.0

Integration Guideline



Integrate^{IT}
Aspect Integrator PlatformTM
Version 2.0

Integration Guideline

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Technology Products AB. ABB Automation Technology Products AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Technology Products AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Technology Products AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB Automation Technology Products AB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

© Copyright 1999-2002 ABB. All rights reserved.

Release: June 2002
Document number: 3BSE 023 960R201

TRADEMARKS

Registrations and trademarks used in this document include:

The Industrial^{IT} wordmark and all product names in the form XXXXXX^{IT} mentioned herein are registered or pending trademarks of ABB.

| | |
|----------------------------|---|
| Microsoft | Registered trademark of Microsoft Corporation. |
| Windows | Registered trademark of Microsoft Corporation. |
| Windows 2000 and MS-DOS | Registered trademarks of Microsoft Corporation. |
| ActiveX and Visual Basic | Registered trademarks of Microsoft Corporation. |
| PostScript | Registered trademark of Adobe Systems Inc. |
| Acrobat Reader | Registered trademark of Adobe Systems Inc. |
| Ghost | Registered trademark of Symantec Corporation. |
| Aspect Integrator Platform | Trademark of ABB Ltd., Switzerland. |
| Aspect Studio | Trademark of ABB Ltd., Switzerland. |
| Aspect Express | Trademark of ABB Ltd., Switzerland. |
| Process Portal | Trademark of ABB Ltd., Switzerland. |

TABLE OF CONTENTS

About This Book

| | |
|---|----|
| 0.1 General | 9 |
| 0.2 Use of Warning, Caution, Information, and Tip Icons | 10 |
| 0.3 Document Conventions | 11 |
| 0.4 Terminology..... | 12 |
| 0.5 Related Documentation | 18 |

Section 1 - Introduction

| | |
|---|----|
| 1.1 Product Overview | 21 |
| 1.1.1 Aspect Integrator Platform™ Runtime | 21 |
| 1.1.2 Aspect Integrator Platform Thin Client Add-on | 22 |
| 1.1.3 Aspect Integrator Platform SDK | 23 |
| 1.1.4 Aspect Studio | 23 |
| 1.1.5 Aspect Express | 23 |

Section 2 - Aspect Object Architecture

| | |
|--------------------------------------|----|
| 2.1 The Aspect Object Model..... | 25 |
| 2.2 Aspect Systems..... | 26 |
| 2.3 Server Aspects | 27 |
| 2.4 Afw Services | 27 |
| 2.4.1 General | 27 |
| 2.4.2 Examples of Afw Services | 28 |

Section 3 - Predefined Aspect System

| | |
|--|----|
| 3.1 Introduction | 29 |
| 3.1.1 User Interface Wrapping | 29 |
| 3.1.2 Calculation and Transfer Aspects | 30 |
| 3.1.3 Report and Message support | 30 |
| 3.1.4 NLS support | 30 |
| 3.2 File viewer Aspect | 31 |
| 3.2.1 Bookmark Aspect | 31 |
| 3.3 Web Aspect | 32 |
| 3.4 ActiveX Wrapper Aspect | 32 |
| 3.5 Windows Application Wrapper Aspect | 32 |
| 3.6 Help Aspect | 32 |
| 3.6.1 Advantages for Object Type | 33 |
| 3.6.2 Advantages for Object Instance | 33 |
| 3.7 General Properties Aspect | 33 |
| 3.8 Property Transfer Aspect | 33 |
| 3.9 Property Translation Aspect | 34 |
| 3.10 Property Signal Generator Aspect | 35 |
| 3.11 OLE DB Provider | 35 |
| 3.12 Operator Note Aspect | 35 |
| 3.13 NLS resource Manager Aspect | 35 |
| 3.14 Optional Tools | 35 |

Section 4 - Aspect Integration

| | |
|--|----|
| 4.1 Introduction | 37 |
| 4.2 Key Technologies | 38 |
| 4.3 Integration levels | 39 |
| 4.3.1 Aspect User Interface Integration Wrapping [Level 0] | 40 |
| 4.3.1.1 User functionality | 40 |
| 4.3.1.2 Tools | 40 |
| 4.3.1.3 Integration details | 41 |

| | |
|---|----|
| 4.3.2 Aspect User Interface Integration [Level 1]..... | 42 |
| 4.3.2.1 User functionality | 42 |
| 4.3.2.2 Tools | 42 |
| 4.3.2.3 Implementation details | 42 |
| 4.3.3 Aspect Navigation Integration [Level 2]..... | 43 |
| 4.3.3.1 User functionality | 43 |
| 4.3.3.2 Tools | 43 |
| 4.3.3.3 Implementation details | 44 |
| 4.3.4 Aspect Engineering Integration [Level 3]..... | 45 |
| 4.3.4.1 User functionality | 45 |
| 4.3.4.2 Tools | 46 |
| 4.3.4.3 Implementation details | 47 |
| 4.3.5 Aspect Administration Integration [Level 4] | 48 |
| 4.3.5.1 User functionality | 48 |
| 4.3.5.2 Tools | 48 |
| 4.3.5.3 Implementation details | 49 |
| 4.3.6 Aspect Data Management Integration [Level 5]..... | 50 |
| 4.3.6.1 User functionality | 50 |
| 4.3.6.2 Tools | 50 |
| 4.3.6.3 Implementation details | 50 |
| 4.4 Integration level summary | 51 |

Section 5 - Implementing an Aspect System

| | |
|---|----|
| 5.1 Visual Basic or C++?..... | 53 |
| 5.1.1 C++ | 53 |
| 5.1.2 Visual Basic..... | 54 |
| 5.1.3 Comparing Aspect Express and Aspect Objects AppWizard | 54 |
| 5.2 Aspect Framework layers | 55 |
| 5.2.1 Afw Interfaces | 57 |
| 5.2.2 Automation Models..... | 57 |
| 5.2.2.1 Aspect Automation Model | 57 |
| 5.2.2.2 Workplace Automation Model | 57 |

5.3 Thin Client 58
 5.3.1 Web-enabling aspect systems 59

Section 6 - Integration Tools

6.1 Introduction 61
 6.1.1 Aspect Integrator Platform..... 62
6.2 Aspect Integrator Platform run-time 64
6.3 Aspect Integrator Platform SDK 64
 6.3.1 SDK Tools..... 65
 6.3.2 SDK Public Interface documentation 65
6.4 Aspect Studio 66
 6.4.1 Aspect Studio Wizards..... 66
 6.4.1.1 Aspect Objects AppWizard..... 66
 6.4.1.2 Thin Client Support 68
 6.4.2 Configuration Management Tools 69
 6.4.2.1 General..... 69
 6.4.2.2 Environment set-up..... 69
 6.4.2.3 Build manager..... 69
 6.4.2.4 AutoBuild..... 70
 6.4.2.5 Product Packager 70
 6.4.2.6 Product Viewer 72
6.5 Aspect Express 74
 6.5.1 Aspect Express Builder..... 74
 6.5.2 Aspect Express Packager 77
INDEX 79

About This Book

0.1 General

The intent of this document is to serve as an introduction to users of Industrial^{IT} systems that, in some way, want to extend the scope with new functionality. This includes writing own functionality as well as integrating an already existing application.

The document describes the basic concept, different alternatives for you to extend the platform and the tools available for you to do so. It describes a number of software products and gives you references to all relevant documentation.

[Section 2, Aspect Object Architecture](#) introduces the Aspect Object Model. You need to understand it and all the terms introduced there before you can read the rest of the book.

[Section 3, Predefined Aspect System](#) introduces a number of predefined aspects that allow you to build applications without programming.

[Section 4, Aspect Integration](#) introduces the concept of integration levels and addresses the six different integration levels and their benefits. How far you can reach into the Integration level steps with the various tools described in this book is also clarified.

[Section 5, Implementing an Aspect System](#), describes the differences and advantages with different programming languages and development tools.

[Section 6, Integration Tools](#) brings you deeper into the tools that are provided and describes the content of and the help you receive from the Integrate^{IT} Products.

0.2 Use of Warning, Caution, Information, and Tip Icons

This publication includes **Warning**, **Caution**, and **Information** where appropriate to point out safety related or other important information. It also includes **Tip** to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:



Electrical warning icon indicates the presence of a hazard which could result in *electrical shock*.



Warning icon indicates the presence of a hazard which could result in *personal injury*.



Caution icon indicates important information or warning related to the concept discussed in the text. It might indicate the presence of a hazard which could result in *corruption of software or damage to equipment/property*.



Information icon alerts the reader to pertinent facts and conditions.



Tip icon indicates advice on, for example, how to design your project or how to use a certain function

Although **Warning** hazards are related to personal injury, and **Caution** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **Warning** and **Caution** notices.

0.3 Document Conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the ENTER key.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
- Press CTRL+C indicates that you must hold down the CTRL key while pressing the C key (to copy a selected object in this case).
- Press **ESC E C** indicates that you press and release each key in sequence (to copy a selected object in this case).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
 - The following convention is used for menu operations: **MenuName > MenuItem > CascadedMenuItem**. For example: select **File > New > Type**.
 - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.
- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

Entered value is not valid. The value must be 0 to 30.

You may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

TIC132

Variables are shown using lowercase letters.

sequence name

0.4 Terminology

The list contains terms and abbreviations that are unique to ABB or have a usage or definition that is different from standard industry usage.

| Term | Description |
|----------------------------|--|
| ActiveX | Microsoft standard for user interface components, based on definition of software interfaces. |
| Aspect | An aspect is a description of some properties of a real world entity. The properties described could be mechanical layout, how the object is controlled, a live video image, name of the object etc. In the Aspect Integrator Platform is an aspect residing in an Aspect Object. Some examples of aspects are circuit diagram, process display and control logic. |
| Aspect Category | A specialization of an aspect type. For example, the aspect type Graphic Display includes the categories Overview, Group and Object Display. |
| Aspect Integrator Platform | A collection of software that forms the basis for an Industrial ^{IT} System, and provides the development and execution environment for Industrial ^{IT} compliant applications. The Aspect Integrator Platform includes the Aspect Framework. |
| Aspect Objects | A computer representation of a real world entity like a pump, a valve, an order or a virtual object like a service. This computer representation is implemented by the Aspect Integrator Platform. An Aspect Object works like an information container for it's aspects. |

| Term | Description |
|-----------------------|---|
| Aspect Object Type | An Aspect Object Type defines certain characteristics that are shared between several Aspect Object instances, such as a basic set of common aspects. This makes it possible to create and efficiently re-use standardized solutions to frequently recurring problems. For example, rather than building an Aspect Object from scratch for every valve in a plant, you can define a set of valve types, and then create all valve objects of these instances. |
| Aspect Server | A server that runs the central functions of the Aspect Object architecture, such as Aspect Directory, Structure and Name Server, Cross Referencing, File Set Distribution, etc. |
| Aspect System | A software system, which implements one or several aspect types by providing one or several aspect system objects. |
| Client | Client is the part of the software that supply data to a subscriber. |
| Client/Server Network | A client/server network is used for communication between servers, and between workplaces and servers. |
| Connectivity Product | Connectivity components, up-loader, supporting aspect systems (e.g for the configuration), and graphical elements, faceplates, Aspect Object Types, etc., bundled together to provide the integration of a certain type of devices into the Industrial ^{IT} system. |
| Connectivity Server | A server that provides access to controllers and other sources for real-time data, historical data, and alarm and event data. A Connectivity Server runs services related to OPC/DA, OPC/AE, OPC/HDA and SysMag. |

| Term | Description |
|---------------------------------|--|
| Faceplate | A faceplate is an aspect that provides a graphical representation of a certain aspect object, with presentation of certain properties related to the object, and mechanism for operator interaction such as on/off, increase/decrease, etc. Aspect Object types often include several faceplate aspects, providing different presentation and interaction possibilities. |
| Graphic Display | A graphic display is an aspect that provides a visual presentation. It consists of static graphics representing for example tanks, pipes etc., and graphic elements that present dynamic information. Graphic displays are often used to present the state of a process or a part of a process, but are useful in any context where dynamic graphical information is needed. |
| Industrial ^{IT} | ABB's vision for enterprise automation. |
| Industrial ^{IT} System | A computer system that implements (part of) the Industrial ^{IT} vision. Aspect Integrator Platform is an example of such a system. |
| Node | A computer communicating on a network e.g. the Internet, Plant, Control or IO network. Each node typically has a unique node address with a format depending on the network it is connected to. |
| OPC | An application programming interface defined by the standardization group OPC Foundation. The standard defines how to access large amounts of real-time data between applications. The OPC standard interface is used between automation/control applications, field systems/devices and business/office application. |
| Plant Explorer | An application that is used to create, delete and organize Aspect Objects and Aspects within the Aspect Integrator Platform. The plant explorer organizes the Aspect Objects in structures according to functionality, location etc. You can also use it to browse and search the structures of the plant. |

| Term | Description |
|------------------|---|
| Process Portal A | Industrial ^{IT} System Product containing functionality for efficient control and supervision of an automated process. Key functions are presentation of graphics, process dialogs and presentation of alarms and trends. |
| Permission | A permission groups a set of operations that require the same authority. For each operation defined for an aspect, the aspect category specifies the permission needed to use that interface. |
| Process Object | A process concept/equipment e.g. valve, motor, conveyor or tank. |
| Product Family | A range of products within a Product Suite, forming a scalable offering. Examples: a range of controllers, a family of I/O Products. |
| Product Suite | Product with similar functionality are kept together in a suite. Suite names have a superscripted IT-suffix. Examples: Operate ^{IT} , Control ^{IT} , Integrate ^{IT} etc. |
| Property | A data field on an aspect of an Aspect Object that can be accessed through OPC using the standard Aspect Object reference syntax. A data field on an ActiveX control accessible from the Visual Basic editor. |
| Security | Security controls a user's authority to perform different operations on Aspect Objects, depending on several parameters: <ul style="list-style-type: none"> • The user's credentials, as provided by Windows • The node where the user is logged in. This makes it possible to give a user different authority depending on where he/she is located, e.g. close to the process equipment, in a control room, or at home accessing the system through Internet. • The operation the user wants to perform the operation on. |

| Term | Description |
|--------------------|---|
| Server | A node that runs one or several Afw Services. It is the part of the software that supply data to a subscriber. |
| Structure | A hierarchical tree organization of Aspect Objects. Each structure is used to define a certain kind of relation between Aspect Object. The functional structure defines how a function can be divided into sub functions, the location structure defines how different objects are located within each other. The control structure defines how functions are executed by tasks, controllers etc. An Aspect Object can be located in several structures, for example both in a functional structure and in a location structure. |
| System Application | A software package that provides functionality in the Industrial ^{IT} System. System applications cooperate according to rules defined by the Industrial ^{IT} architecture, using mechanism provided by the Aspect Integrator Platform. They are normally bundled into System Products. To participate in Aspect Object operations, and thus be an integrated part of an Industrial ^{IT} system, a system application must present itself as an aspect system. When there is no risk for confusion with user application, the term <i>application</i> may be used instead of <i>system application</i> . |
| System Extension | A system Extension consists of one or more applications that are bundled as an extension to one or several existing System Product(s). A System extension can only be installed if (one of) the corresponding System Product(s) has been installed previously. |
| System Product | A system product consists of applications bundled together with relevant parts of the Aspect Integrator Platform. It is complete from installation point-of-view, and requires only Windows 2000. Several System Products can be installed on the same physical node. |

| Term | Description |
|------------------|--|
| Uploader | An upload is used to import a configuration from devices, to read in and build a set of Aspect Objects from information present in the devices. |
| User application | A configuration of software and hardware components that applies to a specific problem, e.g. a specific process control problem. A user application consists of a set of simple and composite Aspect Object instances, with parameter values and other configuration data for the aspects, e.g control logic, graphics, alarm and event specifications, reports etc. |
| View | An Aspect can have several ways to be presented depending on the task performed, like viewing or configuration. Each presentation form is called a view. |
| Workplace | <ol style="list-style-type: none"><li data-bbox="634 746 1292 811">1. User interactive functions that are combined for a particular use, e.g, Operator Workplace.<li data-bbox="634 811 1292 876">2. A node that runs one or several workplace applications. |

0.5 Related Documentation

| Category | Title | Description |
|--|---|--|
| Installation and System Administration | Integrate ^{IT} Aspect Integrator Platform SDK Installation Guide | This book describes how you install the Aspect Integrator Platform SDK, including the requirements on the hardware. |
| | Integrate ^{IT} Aspect Integrator Platform Run-time Installation Guide | This book describes how you install the Aspect Integrator Platform Run-time, including the requirements on the hardware. |
| | Integrate ^{IT} Aspect Integrator Platform Administrator's Guide | This book describes how you configure products built on the Aspect Integrator Platform and how you then perform maintenance. |
| | Integrate ^{IT} Aspect Integrator Platform Security Reference Manual | This book describes how to set security in products built on the Aspect Integrator Platform. |
| Software Development | Integrate ^{IT} Aspect Integrator Platform Architecture Document | This book describes the Integrate ^{IT} architecture in details. |
| | Integrate ^{IT} Aspect Integrator Platform Automation Models Programmer's Guide | This book provides a number of examples on how to use the Automation Models. |
| | Integrate ^{IT} Aspect Integrator Platform Programmer's Guide | This book contains a comprehensive description of how to program the Aspect Integrator Platform. |
| | Integrate ^{IT} Aspect Studio User's Guide | This book describes Aspect Studio excluding Aspect Express (which is described in a separate User's Guide). |
| | Integrate ^{IT} Aspect Express User's Guide | This book describes everything about Aspect Express including installation. |

| Category | Title | Description |
|----------|---|---|
| Software | Integrate ^{IT} Aspect Integrator Platform Engineering Basics Reference Manual | This book describes how to engineer the Integrate ^{IT} . |
| | Integrate ^{IT} Aspect Integrator Platform Predefined Aspects Reference Manual | This book describes how to use predefined aspects, such as property transfer and translation. |
| | Integrate ^{IT} Aspect Integrator Platform NLS Translator's Guide | This book describes how to translate the operator interface language in products built on the Aspect Integrator Platform. |
| | Integrate ^{IT} Aspect Integrator Platform Workplace and Plant Explorer Reference Manual | This book describes everything about the user interface, for example how you present information, navigate and control objects. |
| | Integrate ^{IT} Aspect Integrator Platform Alarm and Event Reference Manual | This book explains how to use the Aspect Integrator Platform Alarm and Event functions. It covers alarm and event configuration and presentation. |
| | Integrate ^{IT} Aspect Integrator Platform Graphics Reference Manual | This book describes how to configure and use graphic aspects in Aspect Integrator Platform. |
| | Integrate ^{IT} Aspect Integrator Platform Graphics Builder Reference Manual | This book covers how to use the Graphics Builder software to create graphic aspects. |
| | Integrate ^{IT} Aspect Integrator Platform Graphics Standard Building Blocks Reference Manual | This book describes the building blocks provided by ABB for the graphics in Aspect Integrator Platform. |
| | Integrate ^{IT} Aspect Integrator Platform Trends and Historian Reference Manual | This book describes how to use the Historian data logging and Trend graphs to view historical data. |

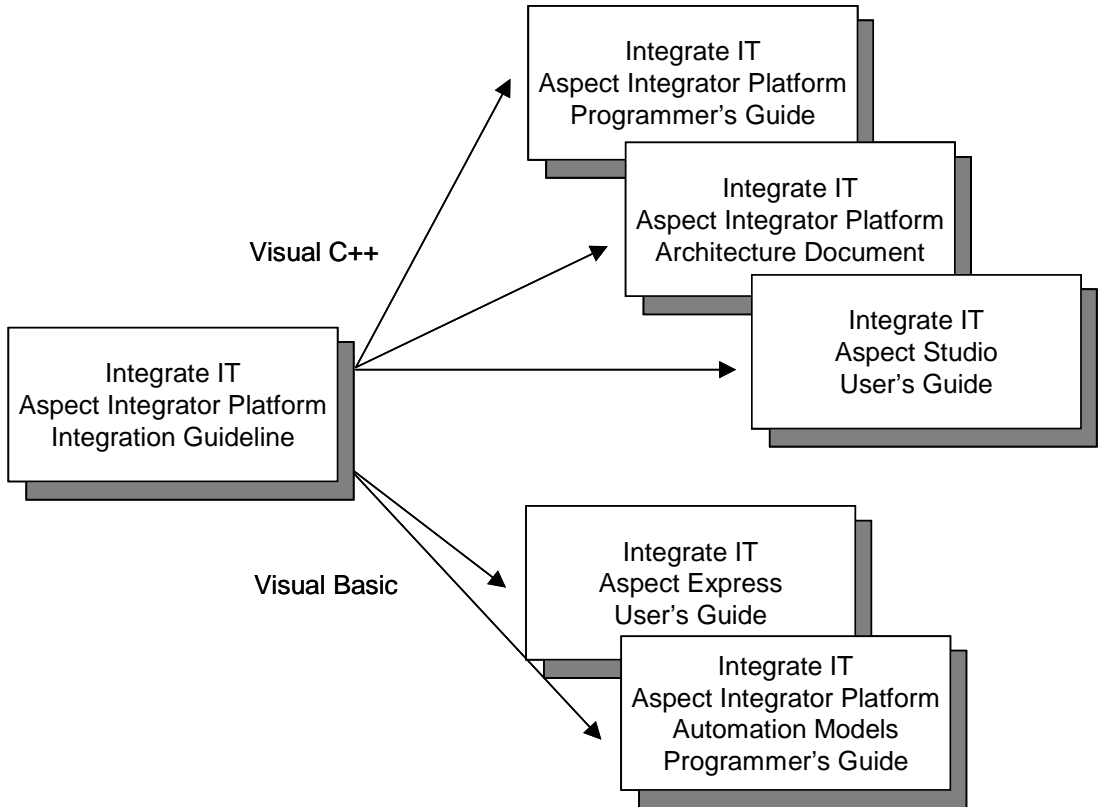


Figure 0-1. Document Overview for IIT Software Developers

Section 1 Introduction

1.1 Product Overview

The Integrate^{IT} suite contains products that are to be used, by all kinds of developers, in the different steps of Industrial^{IT} application and system product development. The five main products you use for Application/Aspect development for Industrial^{IT} systems follow.

1.1.1 Aspect Integrator PlatformTM Runtime

This provides the basic environment for test and is the base for the creation of other system products. It includes:

- Basic Services, with system supervision, diagnostics and error handling, license management, service management, national language support (NLS), timing and time management, system administration, etc.
- Communication Services, with "upward" and "downward", Ethernet-based, communication services.
- Object Handling Services, with Aspect ObjectTM management, Aspect Directory, object browsing, authority definition and control, life-cycle management and version handling, back-up and restore services, etc.
- Data Subscription Services, with data collection, cyclic subscription, OPC servers, etc.
- User Interface Management, including Workplace Manager, Plant Explorer, etc.
- The run-time portions of the Aspect Automation Model and the Graphics.
- Wrapper aspects, which make it possible to wrap Documents, HTML pages, ActiveX components and ordinary executables.

- Alarm and Event, with collection, storage and presentation of Alarms and Events.
- Transfer and Calculation aspects, that make it possible to read Aspect Object properties, to perform calculations on them, and to transfer result(s) to other Aspect Object properties, complemented with aspects for definition and storage of Aspect Object properties. A lot of applications can be configured with these aspects.
- Graphics, which manages graphic displays, faceplates, and graphical building blocks, used to monitor and control a process.
- Graphics Builder option, the tool for building graphic displays. It is an add-in to Visual Basic and the displays are ActiveX elements.
- Trend aspects, that monitor the process over a period of time, through trend traces.
- Historian option, which enables data acquisition and long term storage on disk. Trend is used to view Historian data.

1.1.2 Aspect Integrator Platform Thin Client Add-on

Thin Client add-on enables access to Aspect Objects, and corresponding aspects, via intranet/internet. It includes:

- Full workplace support including structure and object browsing.
- Viewing of Web-enabled aspects on thin clients.
- Exposure of AIP functions as web services.
- Mapping of aspects as URLs.
- Engineering and configuration made on a “rich“ environment is re-used on a Thin Client, as long as the Aspect system is Web enabled.
- Provides Thin Client functionality for WAP phones, Personal Digital Assistants (PDAs), and ordinary PCs.

Thin Client is packaged as an addition to the Aspect Integrator Platform and contains both runtime and SDK support. It requires that Aspect Integrator Platform is installed.

1.1.3 Aspect Integrator Platform SDK

- Together with the Aspect Studio™ and Aspect Express™ tools, the Aspect Integrator Platform SDK provides the development environment for the Aspect system, System Product Extension and System Product developers.
- The Aspect Integrator Platform is used by all kinds of developers in the different steps of Industrial^{IT} application and system product development.

1.1.4 Aspect Studio

- Aspect Studio provides the Aspect Objects AppWizard which creates C++ templates for the most important functions of an aspect system.
- Aspect Studio includes Aspect Express, see Aspect Express below.
- Aspect Studio provides the tools that your configuration manager and developers require in order to set up the development environment, program, debug, build, package, document, deliver and install the resulting aspect systems as well as complete Industrial^{IT} system products.
- Aspect Studio is the tool for experienced and qualified users as well as for unexperienced developers who need to utilize the full functionality of the Industrial^{IT} concept.
- Aspect Studio supports development of Web enabled aspect systems. It requires the Thin Client add-on for test and execution

1.1.5 Aspect Express

- Aspect Express has an add-in for Visual Basic that provides a component sensitive integration of new or existing ActiveX-based applications into Industrial^{IT} system products.
- Aspect Express is easy to learn and easy to use. It does not require any knowledge of C++, COM or ATL.
- Aspect Express is delivered as a separate product as well as a part of Aspect Studio. An Industrial^{IT} system product is required to test the resulting product - typically Process Portal™ A or the Aspect Integrator Platform. The resulting aspect systems can be installed directly into your Industrial^{IT} system product.

Section 2 Aspect Object Architecture

2.1 The Aspect Object Model

The Aspect Object Model addresses the issue of presenting information and allowing a user to operate on information in a consistent way. The model also addresses how to integrate new functions into the system, in a natural way for the user. The Aspect Object Model is based on Aspect Objects and aspects.

The Aspect Objects in the model are the objects the user interacts with. For example, an object can be a reactor, a pump or a node (computer). The object itself is a container that holds different parts, aspects, of the object. See [Figure 2-1](#).

An aspect is one 'piece' of data and operations that are associated with an object. Typical aspects of an (process) object are its control program, operator faceplate, trend configuration, function specification etc.

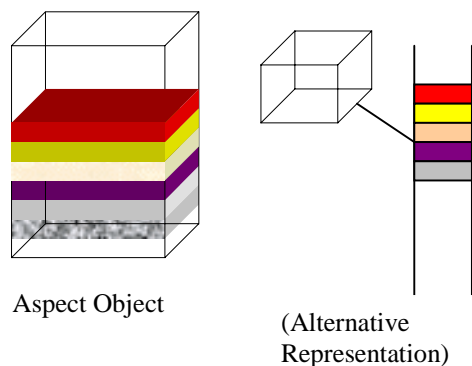


Figure 2-1. An Aspect Object is a Model of a Real World Object

Although we talk about objects, it is the aspects of the object that provide the interaction with the object. The aspects carry data and implement operations on that

data. All interactions with an object in an Industrial^{IT} system are carried out with the help of its aspects.

The Aspect Object model allows information to be presented in a consistent way and provides a way of naturally and incrementally extending a system as it evolves over time. The system is modular, as functions and components can be added or removed independently.

2.2 Aspect Systems

When a new function is added to an Industrial^{IT} system, it is done by adding a software component. If this component deals with objects (as most systems do), it has to supply an Aspect system. An aspect system makes it possible to create aspects for the Aspect Objects in the system. The user (or another software component) works with the new function by creating or using aspects. See the following example:

A history function is introduced into the system and the logging of data needs to be configured. When the history function is introduced, it includes the possibility to add and work with log aspects. These log aspects allow a user to bring up a viewer to get information about the selected configuration and change it. The aspect also allows other software components to use it to get access to the historic data etc.

Most functions in the system are centered on objects. When this is the case, the functionality should be introduced through aspect systems. Having recognized this, we may ask ourselves, what is an aspect system from the implementer's view? The answer to this question is the set of software components that allows aspects to be created and operated on. This typically includes components that implement objects that hold data and implement the operations that are directly invoked by a user, such as view, copy etc., and other operations that are used by other aspect systems or the Aspect Integrator Platform. An aspect system may introduce one object that implements all functions of the aspect system, or use several objects to do the job. Using several objects will often make the task easier to implement, and may also lead to better performance.

2.3 Server Aspects

The components, which realize an aspect system, are called ASO (Aspect System Objects). These objects are COM objects, which implement some mandatory interfaces, and should not be confused with the Aspect Objects.

Server aspects are 'normal' aspects that have aspect system objects running in the server layer, i.e. run around the clock with supervision. Such ASOs are developed in the very same way as any other ASO. Clients do not have to know if the targeted aspect is a server aspect or not.

Some functions do not lend themselves to be aspects of objects in a natural way. For example, the name server, whose primary purpose is to translate names (strings) into object identifiers, is not a function of an object and is therefore not implemented as an aspect system. Functions not suited to be implemented as aspect systems are very often implemented as Afw Services, see section [Afw Services](#) below.

2.4 Afw Services

2.4.1 General

Afw Services are functions that need to run around-the-clock. It is common for a service to interact with Aspect systems.

The service concept provides redundancy support with fault tolerance, system wide accessibility, and automatic start-up and shut down, as some of its main features.

The Aspect Studio AppWizard provides support for development of AfwServices in C++. There is no support for development of AfwServices in Visual Basic.

2.4.2 Examples of Afw Services

The Aspect Framework contains a large set of services for the implementation of the functions it provides. Some common services are:

- Aspect Directory
- System Message
- AppLog
- OPC Connector
- Structure and Name Server (SNS)
- File Set Distribution (FSD).

See also *Integrate^{IT} Aspect Integrator Platform Architecture Document*.

Section 3 Predefined Aspect System

3.1 Introduction

The Aspect Integrator Platform contains a lot of application development tools in the form of aspect systems, that make it possible to create advanced and very useful applications without any programming or complementary development tool.

The aspects systems can be sorted into groups according to their use.

3.1.1 User Interface Wrapping

These tools make it possible to reach Integration Level 1, as described in [Section 4, Aspect Integration](#). They consist of the following aspect systems:

- File Viewer and Bookmark aspects, see [File viewer Aspect](#), on page 31.
- Web Aspect, see [Web Aspect](#), on page 32.
- ActiveX Wrapper, see [ActiveX Wrapper Aspect](#), on page 32.
- Windows Application Wrapper, see [Windows Application Wrapper Aspect](#), on page 32.
- Help Aspect, see [Help Aspect](#), on page 32.

3.1.2 Calculation and Transfer Aspects

These tools make it possible to transfer Aspect Object properties from one or many Aspect Objects to properties placed at destination Aspect Object(s) and at the same time performing arithmetic operations on the data. Other aspects can be used as data containers (properties) useful for these operations or as test data sources.

- General Properties, see [General Properties Aspect](#), on page 33.
- Property Transfer, see [Property Transfer Aspect](#), on page 33.
- Property Translation, see [Property Translation Aspect](#), on page 34.
- Property Signal Generator, see [Property Signal Generator Aspect](#), on page 35.

3.1.3 Report and Message support

The OLE DB providers for History (OPC HDA) and Single point data (OPC DA) provide interfaces for reporting tools. The Operator Note facility makes it possible to save Messages and Memo for other users.

- OLE DB providers, see [OLE DB Provider](#), on page 35.
- Operator Note, see [Operator Note Aspect](#), on page 35.

3.1.4 NLS support

- The NLS resource manager, see [NLS resource Manager Aspect](#), on page 35.

3.2 File viewer Aspect

The File Viewer is used to integrate documents into AIP. An aspect can have two types of documents; a Work document and a Public document. Both documents can exist for the same aspect. Only the read-only Public document can be accessed by users of the operator category. The Work document can be the source document of the Public document, or a work in progress. If proper applications are installed on the current computer, users will be able to edit the Work document.

Work and Public documents can be stored in different formats. For example, the Work document can be a Microsoft Word document, and the Public document can be an Acrobat Adobe.

The File Viewer aspect provides full client/server support, which means that aspects/documents added on one node in the current configuration are directly available on the others. Another feature is the ability to create and use templates, making it possible to standardize look and feel on new documents that are attached to File Viewer aspects.

All documents are fully associated with the aspect. This means that a copy and paste operation creates a new set of documents. It also means that when performing an Export the documents will be contained inside the export files.



The File Viewer should be used, instead of the Windows application aspect, for viewing documents, as it has superior functionality for this kind of wrapping. The usage of the Windows application aspect should be limited to execute Windows applications that are not related to documents or data files.

3.2.1 Bookmark Aspect

The Bookmark Aspects are placed as bookmarks to a File Viewer aspect, making it possible to enter a document at a specific place.

3.3 Web Aspect

You can use the Web aspect to access any Web page or html document. It utilizes the WebBrowser ActiveX component from Microsoft. You configure the Web aspect towards a Web page/html document, which in turn may contain URLs linked to other pages on the Web. When you click on such a URL, the referenced Web page will be displayed. This is the normal way of web browsing, and the only thing you need to do is to configure the Web aspect.



ABB does not recommend that you configure links outside your own Intranet without commitments from your IT manager.

3.4 ActiveX Wrapper Aspect

The ActiveX Wrapper aspect provides you with a method to wrap any ActiveX Control. When you have configured the ActiveX Control, you simply select that aspect to execute it.

There are a number of third party and ABB applications, that include useful ActiveX Controls, that can be integrated into Industrial^{IT} Products.

3.5 Windows Application Wrapper Aspect

The Windows application aspect enables the launching of most types of Microsoft or third party tools; for example, Wordpad, Notepad, MS-Word, and Excel. These can be accessed as aspects, using the same user interface used for other features.

3.6 Help Aspect

You can use the help aspect to create online-help for an object type or an object instance. You can create files containing on-line help information, using various third party tools. You can also refer to information in the comprehensive set of Industrial IT online help files.

You can find the Industrial IT help files under, for example, ABB Industrial IT/Integrate IT/.../..

3.6.1 Advantages for Object Type

If you create on-line help aspects and place them on the object type, that help can function as help for that specific object type. If you choose to use inheritance for the help, you can even change the help later on and get an automatic update of the object instances help.

3.6.2 Advantages for Object Instance

You can create individual instructions and help for the respective object. This can be very useful for complex application objects, created in a plant. Tuning instructions, maintenance instructions, or help for emergency situations, can be part of the help.

3.7 General Properties Aspect

You can use the general properties aspect as a source or as a storage for complementary object or aspect information. It is very useful together with the Property Transfer and Property Translation aspects.

Just as the Operator Notes aspect is useful for storage of text messages, the general properties aspects are useful for data of any kind, such as integer, real, boolean and string formats. Furthermore, you can create structures of properties within one general properties aspect.

General properties are available for all AIP clients, as OPC items. This means that you can present them, for example, in Graphics, Trends and in Reports (through the OLE DB provider).

3.8 Property Transfer Aspect

You can use the Property Transfer functions for several purposes:

- As a transfer function for data from one or several data sources (Aspect Object properties) to one or several resulting Aspect Object properties.
- As a calculation function, not only reading data from Aspect Object properties but also entering these data into a suitable algorithm and delivering the results to Aspect Object properties.

The data collection as well as the data delivery can be done from/to any public Aspect Object property (including the Aspect properties from aspects such as the Structured Property aspect).

This means that object properties in controllers, are accessible both as data sources and data receivers. Object properties are made available through a Connect product or OPC Servers, connected with the Generic OPC solution.

The expression shall follow the syntax rules given by the C programming language.

3.9 Property Translation Aspect

You can use the Property Translation aspect to create text strings presenting the result of an evaluation or calculation of one or several object properties. The result is available as a text string which simplifies the presentation in displays and reports.

The data collection can be done from any public Aspect Object property (including the Aspect properties from aspects such as the Structured Property aspect). This means that object properties in controllers, are accessible both as data sources and data receivers. Object properties are made available through a Connect product or OPC Servers, connected with the Generic OPC solution.

The expression shall follow the syntax rules given by the C programming language.

There are two Property Translation aspects:

- Property Translation
In this you use a literal translation for the string translation expression
- Property Translation Extended
In extended you can also use expressions when translating to string and you can set strings both for the result **True** and **False** (of the expression).

The difference between them is that in the Property Translation Extended you can set different outputs in the same line for a given criteria, if it is true or false.

3.10 Property Signal Generator Aspect

You can use the Property Signal Generator aspect to generate a simulated value that can be used to test a display or a display element.

The Property Signal Generator aspect can generate a sinus formed wave, a ramp, a random value, or a static value.

3.11 OLE DB Provider

You can use the OLE DB provider as the access path for report tools into Industrial^{IT}. Most of the existing third party report tools can use OLE DB providers to retrieve information and then produce reports.

The OLE DB Provider is the access path into Industrial^{IT} data report tools.

3.12 Operator Note Aspect

You can use the Operator Notes aspect to leave a message to another user (from one operator to another). You can also use the aspect to store your own memo for a certain object.

The Find function in Plant Explorer makes it simple to find all the Operator Notes aspects when needed.

3.13 NLS resource Manager Aspect

You can use the NLS Resource Manager aspect to set national language for graphic aspects text etc.

3.14 Optional Tools

For details on the optional tools Graphics Builder and Trend, see *Integrate^{IT} Aspect Integrator Platform Graphics Builder Reference Manual*, and *Integrate^{IT} Aspect Integrator Platform Trends and Historian Reference Manual*, respectively.

Section 4 Aspect Integration

4.1 Introduction

Different aspect systems can be more or less well integrated into Industrial^{IT}. To be able to easily describe how well integrated an aspect system is, a classification is used, that specifies different integration levels.

The integration levels specify the necessary functionality and what the expectations should be of a compliant aspect system.

Today, it is typically the case that the set of functions automated in the industry are handled by separate applications, with no or very little intercommunication. Each application is a separate stand-alone entity. The implications are that the operator of the systems must log in to and interface several different systems, and the applications cannot utilize information and functions available in other applications.

Application integration is to access applications in a uniform way, both as a human user and programmatically. What we mean by access is defined below, but it usually has the implication that a user shall be able to access and interface all the applications running in a site from one screen/workstation. Application integration also includes that data and functions from the applications can be integrated and utilized programmatically. When analyzed, the following key integration points can be identified:

- User interface, e.g. GUI.
- Authentication and access to the application.
- Application execution and control.
- Application reporting, such as system messages, events, alarms and application trace messages.
- Real time data subscription and access to stored data.
- Engineering, the ability to design and build the applications with the tools that are provided by the system product.

The Industrial^{IT} architecture is designed to promote application integration. The fundamental idea is to be able to integrate heterogeneous information and operations.

4.2 Key Technologies

The key technologies for integration of an application into Aspect Integrator Platform are:

- Wrapping, for example, documents and ordinary Windows applications.
- Accessing information using the OPC interfaces.
- Implementing Aspect systems.

To begin with, many of the Industrial^{IT} products offer wrappers to simplify this task. With these you will reach the lowest level of integration.

Another method, for integration of an application, is to utilize the OPC standardized interfaces. One of the Aspect Integration Platform corner stones is to communicate with controller hardware and devices using the OPC standard. Many aspect systems also support OPC subscription to their public attributes. There are three OPC standards currently supported within Industrial^{IT}:

1. Data Access - (OPC DA) - the ability to identify, read and write to object attributes.
2. Alarm and Event - (OPC AE) - the OPC standard describes the interfaces for alarm and event related functionality.
3. History - (OPC HDA) - the OPC standard describes the interfaces for History related functionality.

Aspect systems allow you to implement virtually anything as an aspect on an Aspect Object. Implementing an Aspect system is a very powerful way of integrating an application. As described in [Integration levels](#), on page 39, you can decide upon 6 different integration levels depending on your needs and your skills in utilizing a variety of tools.

4.3 Integration levels

To be able to select the wanted integration level, you should consider how the aspect system that you intend to create shall be used. If you follow the description of the different integration levels in this document, and compare them with your view of the aspect system to be, you should be able to decide upon the integration level. This decision together with the skills set that the intended developer has, should give you all the needed arguments to choose the tool(s) to use.

The integration level that a function supports, describes how well the function is integrated into Industrial^{IT}. The integration level classification mainly applies to aspect systems (and to some extent the services that support various aspect systems). The integration levels that are defined are:

- 0 - Aspect User Interface Wrapping
- 1 - Aspect User Interface Integration
- 2 - Aspect Navigation Integration
- 3 - Aspect Engineering Integration
- 4 - Aspect Administration Integration
- 5 - Aspect Data Management Integration

Each integration level is built on the previous level. This means that an aspect system that is compliant to level 4 must also fulfill the rules of level 0 to 4¹. There is, however, nothing that prevents the developer from implementing functionality, from higher levels, that he sees useful.

The rest of this chapter describes each level according to the following:

- User functionality: The functionality the aspect system provides the end user with, on this integration level.
- Development tools: To what extent the development tools support the integration level.
- Implementation details are briefly mentioned. You can find the details in *Integrate^{IT} Aspect Integrator Platform Architecture Document*.

1. The Aspect Framework and the tools described in this document do not yet fulfill this requirement to 100%.

Note that many aspect systems do not need to do much at all, at each level. If you design the aspect systems using Aspect Express or the Aspect Objects AppWizard, many of the issues listed above are automatically taken care of.

4.3.1 Aspect User Interface Integration Wrapping [Level 0]

4.3.1.1 User functionality

This level provides all information with one mouse click, without the user having to know which application to use, or where data is stored. This level basically offers a link to a document, web page or file per Aspect. See [Figure 4-1](#) for an example.

The user interface of the integrated application is placed on an object, using, for example, a wrapper aspect. It can then be shown as an aspect in the Industrial^{IT} system. Each aspect has to be uniquely configured unless you want the same content in the next occurrences of the aspect.

4.3.1.2 Tools

To reach this level of integration you normally use one of the Wrapper aspects, see [Section 3, Predefined Aspect System](#). If you use Aspect Express or the Aspect Objects AppWizard, you can get a more sensitive and intelligent wrapping which can be easily expanded into the other integration levels.

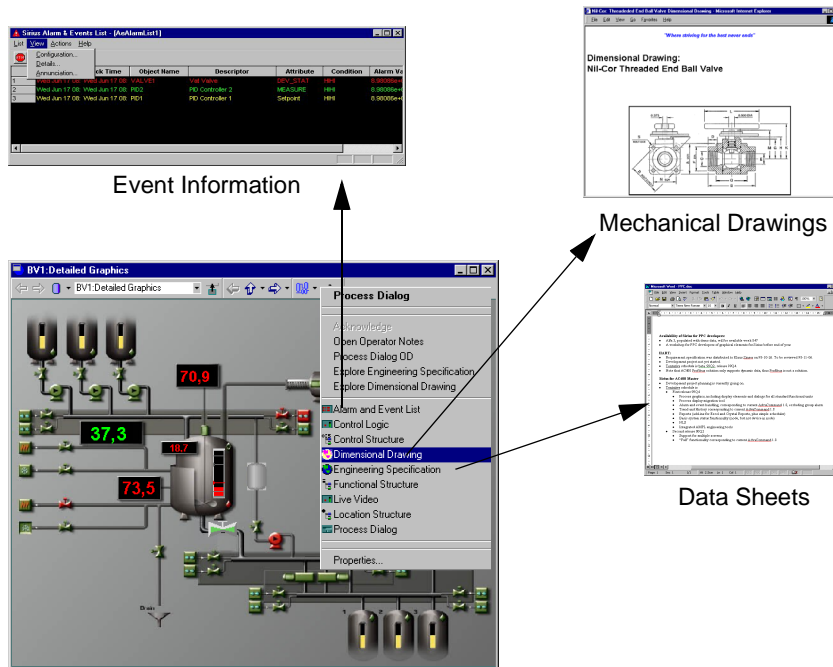


Figure 4-1. Level 0: The Information is a Mouse Click away

4.3.1.3 Integration details

The aspect system must implement its user interface as an ActiveX, ASP/HTML page, Active Document, OLE Server or Windows application.

The advantage of using this level of integration is that it does not require any programming or any software license in addition to the Industrial^{IT} System Product.

4.3.2 Aspect User Interface Integration [Level 1]

4.3.2.1 User functionality

The advantage with this level, compared to level 0, is that the configuration work to create, modify and delete the aspect instances is significantly less than for a Wrapper aspect. From an end user perspective, there are possibilities for a more intuitive navigation. Instead of just opening the aspect, as in level 0, the aspect is opened and the relevant information can directly be shown.

4.3.2.2 Tools

Aspect Express provides this integration level of an existing ActiveX component without any programming, you just run the Aspect Express Builder.

To do this integration level with the Aspect Objects AppWizard, you run the Wizard and add a very limited implementation to the created templates.

4.3.2.3 Implementation details

With this level of integration, one or more COM objects and/or ASP (Active Server Page) pages are registered with the system, to support the new aspect system. Instances initialized from the object/aspect identifiers, are supported. The function recognizes itself as an aspect system and can handle aspect create/delete etc.

An aspect system may also use an ASP or HTML page for its user interface, in which case no COM objects need to be implemented for the user interface.

Implementation details at this level are:

- Registration.
- Initialization of the aspect system.
- Providing at least one aspect view (user interface).
- Aspect instantiation (create and delete).

Most of this is taken care of by the respective development tool, see Tools above.

4.3.3 Aspect Navigation Integration [Level 2]

4.3.3.1 User functionality

This level will give the users a consistent system-wide information access, navigation and information distribution and presentation including the following:

- Context Menu, via right click, within the whole application.
- Navigation between objects/views in the application.
- Public attributes are made available for OPC access and for access using Aspect Automation Model. All configuration parameters are published as OPC properties so that the Aspect system can be configured with standard configuration tools like Bulk Data manager.
- Support for property pages.
- Support for security.
- Cross referencing (optional).
- System messages in the form of events and alarm (optional).
- Application errors, debug and trace messages (optional).

4.3.3.2 Tools

To reach this level you need to use Aspect Express or the Aspect Objects AppWizard, together with your application programming skills. See [Table 4-1](#) for details on the support from the respective tool.

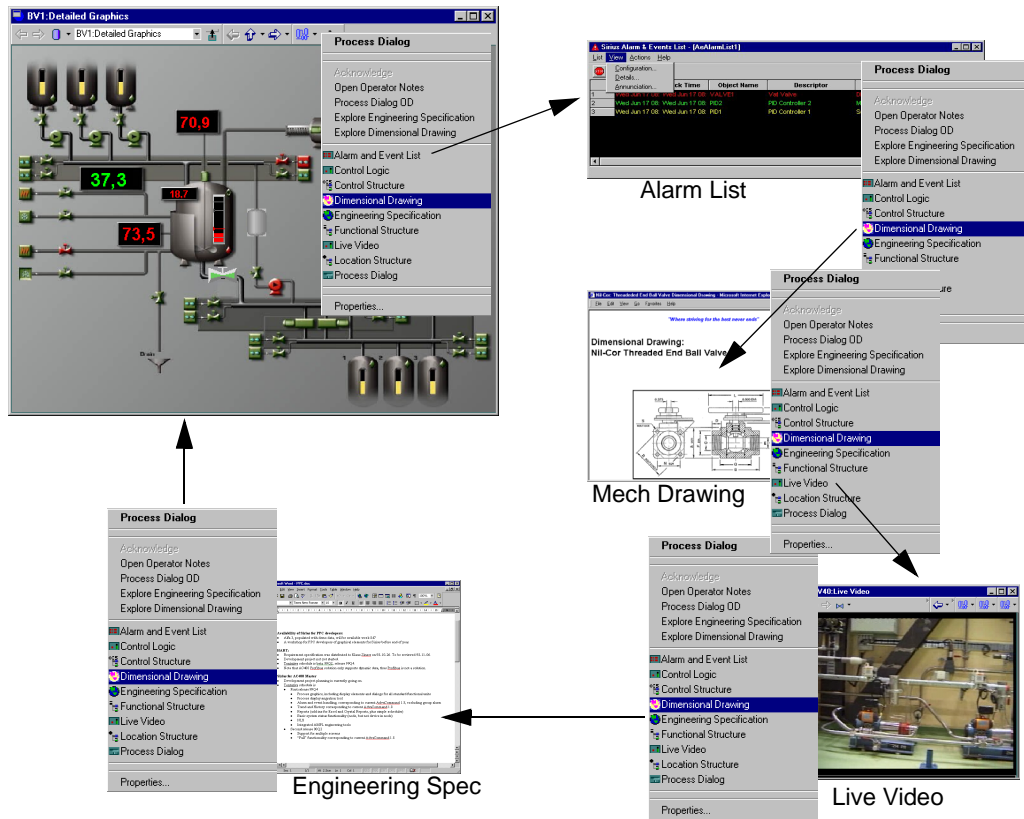


Figure 4-2. Level 2 Example from the End User Perspective

4.3.3.3 Implementation details

The aspect system is integrated well enough to support all navigational features of the system. The aspect system can also make itself heard and seen through system messages (OPC events), and debug messages/traces (AppLog).

- The aspect system provides the standard context menu:
 - for selection of other aspects of the current Aspect Object.
 - with Verbs for activation of functions in the application.
 - with knowledge on how to start up aspects/functions, for example, as overlaps.
- The aspect can also add object and aspect property pages if it holds data that is appropriate to show in a property window.
- Aspect systems integrated at this level should also publish their properties through the subscribe interfaces. This makes the properties available to all other clients, through a standardized access mechanism, OPC/DA. The properties can then also be accessed through Aspect Automation Model. (There are other internal access methods as well, for example, the ASO interfaces).
- To fully support navigation between objects and aspects it may be desired to support cross-referencing. Cross-referencing allows objects to show in what other aspects it is shown, that is, on aspects of other objects.
- Aspect systems can utilize the System message feature, which makes it possible to define NLS compliant event messages, and send them as OPC events, to the AIP Alarm and Event server, for presentation in the corresponding System message lists.
- Aspect systems can utilize the AppLog interfaces, to send error messages or debug/trace messages useful for test, debug and maintenance.

4.3.4 Aspect Engineering Integration [Level 3]

4.3.4.1 User functionality

This level will give the end user support for effective engineering. When you integrate to this level, the aspect can be included in an Object Type and instantiated with an instance of the Object Type.

The aspect can also be copied and pasted either by itself or together with the object it resides in. If the end user changes the aspect in the Object Type, all the instances of the aspect will be changed.

This means that re-use of solutions, with all aspects included, can be utilized and libraries with typical solutions can be built up. The engineering efficiency is taken to a higher level.

4.3.4.2 Tools

Aspect Express supports most of this level, see [Table 4-1](#) for details.

You can also use the Aspect Objects AppWizard together with your C++ programming skills to achieve level 3. See [Table 4-1](#) and the Aspect Integrator Platform Programmers Guide for details.

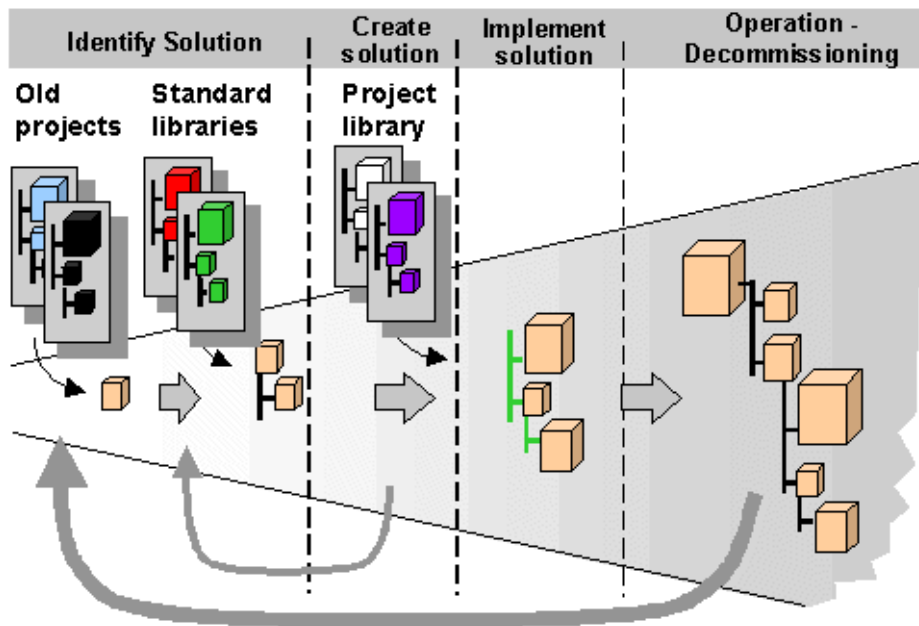


Figure 4-3. Level 3 Example: Re-use Engineering Work

4.3.4.3 Implementation details

The aspect engineering integration level requires that the aspect system understands the object type concept, inheritances, the security model, and other vital concepts used in the Aspect Framework.

The aspect system should also recognize the standard system structures and libraries and extend these structures and libraries where necessary.

The aspect system should support the following functions:

- Relative addressing
- Copy and paste
- Import and export
- Backup and restore

Many aspect systems are able to support these functions without any special considerations. If your aspect system stores the data into the Aspect Directory (default for both Aspect Express and Aspect Objects AppWizard), you do not normally need to do anything to get a functioning copy, paste, import, export and bulk support. Though, if you store data outside the Aspect Directory, you must address this issue.

Inheritance is automatically supported by the Aspect Framework if the following rules are remembered:

- References to the object should not be to the GUID but to the object name or to a combination of these.
- References to other objects in a structure should be made with a relative reference.

When you create aspect systems at this level, it is important you document the following (and perhaps also consider how to package new object types etc., so that everything is included):

- the object types the aspects use, what object types the aspect system brings to the system and generally how the aspect system interacts with different object types.
- how the aspect system supports inheritance (what levels), and also what operations it uses and brings to the system. This is the basis for how authority can be configured in the system.

4.3.5 Aspect Administration Integration [Level 4]

4.3.5.1 User functionality

This level gives you support for Aspect administration. You can deliver your application, in the form of an aspect system, to multiple sites in several countries (multilingual), and the application can be backed up to preserve critical data.

4.3.5.2 Tools

Both Aspect Express and Aspect Studio provide you with the tools to create a complete installation and deployment kit for the developed aspect system. Aspect Studio can handle more complex products and structures.

NLS support is also supported by both tools (just follow the examples and recommendations) while Backup/Restore can be handled using the import/export tool, or backup/restore tool.

4.3.5.3 Implementation details

The Aspect Administration Integration implies that an aspect system will handle its installation, registration, configuration, and data security, according to the principles defined for aspect systems.

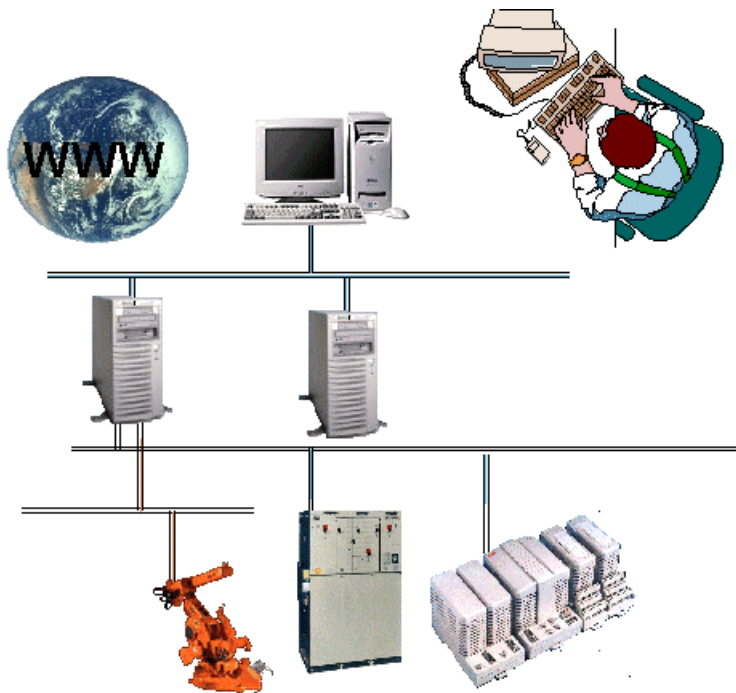


Figure 4-4. Level 4 Deployment, Installation, Multilingual Support and Backup

4.3.6 Aspect Data Management Integration [Level 5]

4.3.6.1 User functionality

The user on this level will have increased data security provided through transaction handling. He can control and manage the versions of the aspect systems he uses, and he can monitor the versions of aspect systems, and all other system and application software, that have executed for each specific time.

4.3.6.2 Tools

Aspect Express automatically supports transaction handling, provided that the aspect system stores all the data in the Aspect Directory.

Aspect Objects AppWizard, together with C++ programming, also provides a transaction handling.

4.3.6.3 Implementation details

Transaction handling means that you, as a user, can change your mind (undo/rollback) until you have committed the changes you have been doing.

Aspect version handling and Life cycle management are not yet supported by the Aspect Framework.

4.4 Integration level summary

The different levels of integration are met by Aspect Express and Aspect Objects AppWizard according to the following table. The Wrapper Aspects support level 0 only, and are thus not included in the table.

Table 4-1. Integration Level Summary

| Integration Level | Support Detail | Aspect Express | Aspect Objects AppWizard |
|--|--------------------------------------|----------------|--------------------------|
| 0: Aspect User Interface Integration Wrapping | | Yes | Yes |
| 1: Aspect User Interface Integration | | Yes | Yes |
| 2: Aspect Navigation and Broadcasting Integration | Standard context menu | Yes | Yes |
| | Context menu verbs | Yes | Yes |
| | Property pages | Yes | Yes |
| | Support public properties (OPC) | Yes | Yes |
| | Support system messages (OPC events) | Yes | Yes |
| | Support AppLog | Yes | Yes |
| | Cross referencing | No | Yes |

Table 4-1. Integration Level Summary (Continued)

| Integration Level | Support Detail | Aspect Express | Aspect Objects AppWizard |
|--|---|---|--|
| 3: Aspect Engineering Integration | Relative addressing | Yes | Yes |
| | Copy and Paste | Yes, as long as the data is not stored outside the Aspect Directory | Yes |
| | Import and Export (Backup/Restore automatically supported). | Yes, as long as the data is not stored outside the Aspect Directory | Yes |
| 4: Aspect Administration Integration | Resulting in a complete installation and deployment kit | Yes | Yes, Aspect Product Packager |
| | NLS support | Yes, if you program according to example and guideline for NLS | Yes, if you program according to example and guideline for NLS |
| 5: Aspect Data Management Integration | Transaction handling | Yes | Yes |
| | Version management | No | No |
| | Life cycle management | No | No |

Section 5 Implementing an Aspect System

5.1 Visual Basic or C++?

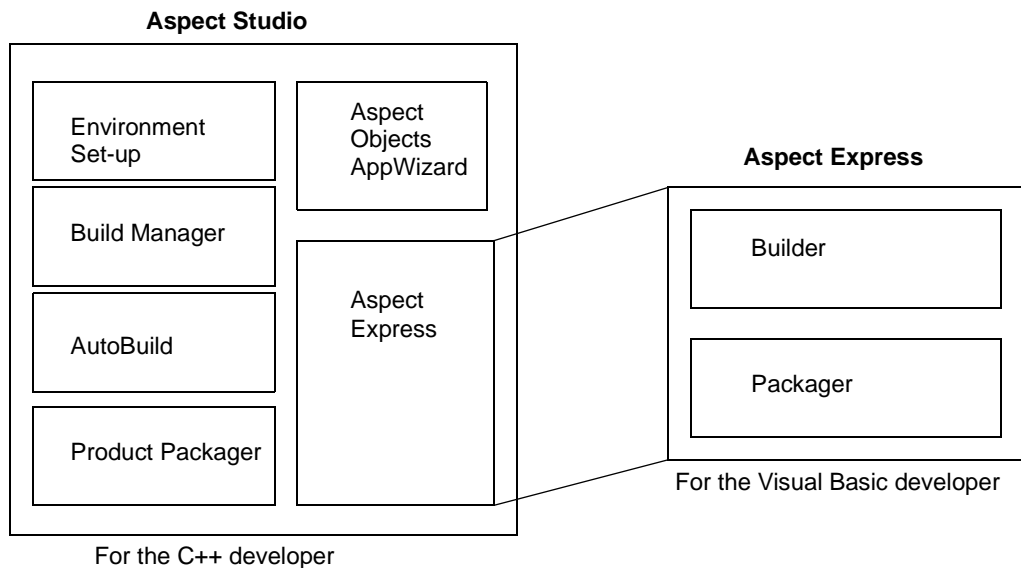


Figure 5-1. Aspect Studio and Aspect Express Developer Packages

5.1.1 C++

C++ offers a lot of freedom but requires an experienced programmer. You need a good knowledge of C++, COM and ATL to be able to do implementation using the Aspect Objects AppWizard. C++ works best with custom interfaces and 'C++-like' parameters (programming OLE Automation from C++ is not recommended).

Aspect Studio provides the Aspect Objects AppWizard, which is a starting point for aspect implementation. The Aspect Objects AppWizard creates C++ templates for the most important functions of an aspect system. It significantly simplifies aspect creation, for example, creation of Aspect System, System Extension, and Service and Test Project applications. Comprehensive functionality for an aspect system is provided, and the developer has control over the execution. Direct implementation in C++ enables aspect systems with high performance.

5.1.2 Visual Basic

Visual Basic offers very fast development, even for inexperienced programmers. Aspect Express combined with Aspect Automation Model and Visual Basic provides tools, a high level language and a lot of features that do not even require programming. Visual Basic can use custom interfaces, but has problems with some ‘C++-like’ parameters.

Aspect Express is an excellent, easy to use tool, for rapid aspect system development. A lot of aspect system functionality is provided with no or little effort. Existing and new ActiveX controls, developed for example in Visual Basic, are turned into aspect systems by a step-by-step wizard. Only a basic knowledge of the Aspect Object architecture is needed. The aspect systems can then be packaged and distributed as installable products.

If you develop your aspect systems with Visual Basic, Aspect Express and Aspect Automation are the preferred tools. The few exceptions might be when you use Graphics Builder and you want to create your own Graphical aspects. You could create even more advanced graphical aspects, access data, or implement other specific features, by using Aspect Automation Model and Workplace Automation Model.

5.1.3 Comparing Aspect Express and Aspect Objects AppWizard

After reading [Table 4-1](#), it may appear as though Aspect Express and Aspect Objects AppWizard give more or less the same kind of support. There are, however, some essential differences:

- Development language
 - Aspect Express is a development tool primarily supporting Visual Basic.
 - Aspect Objects AppWizard is a C++ development tool.

- Complexity of the resulting product
 - You use Aspect Express to develop System Product add-ons of low to medium complexity, at least with regard to the number of involved components and the complexity of the end product. Aspect Express is also limited by the depth of interfaces that Aspect Automation Model provides.
 - Aspect Objects AppWizard has very few limitations with regard to the complexity of the type of application and the type of interface that you implement. Furthermore, you use the Aspect Studio Tools (Environment set-up, Build Manager, AutoBuild and Product Packager) to develop System Product add-ons and System Products of low to high complexity, where the developed application can be included as one component of many in the resulting product. The result may be a complex product which needs to be structured with regard to both the end workspace as well as the final product. Aspect Studio provides support for version handling and for the creation of service packs for such products.
- Performance
 - Applications developed with Aspect Objects AppWizard have better performance than aspects developed with Aspect Express. In cases where performance is especially important, Aspect Objects AppWizard should be the preferred tool.
- The required knowledge level
 - You can use Aspect Express to integrate functionality into independent system product extensions. You only need a basic understanding of the terms and concept of Aspect Objects to achieve this.
 - Before you can use Aspect Objects AppWizard and the configuration management tools in Aspect Studio you need both training and C++, COM, and Aspect Objects knowledge.

5.2 Aspect Framework layers

The Aspect Framework is an Application Programming Interface (API) based on COM technology, and therefore independent of any specific language. With COM technology it is possible to select the language that is best suited for each problem.

The Aspect Framework has two layers:

1. The core layer contains interfaces starting with IAFw. Most of these interfaces are designed with C++ clients in mind. Visual Basic can use many of these interfaces without problems, but in some cases it can become difficult.
2. The Aspect Automation Model layer contains interfaces starting with IABB. This layer is a wrapper around the core layer and is suitable for scripting languages, such as JavaScript and VBScript.

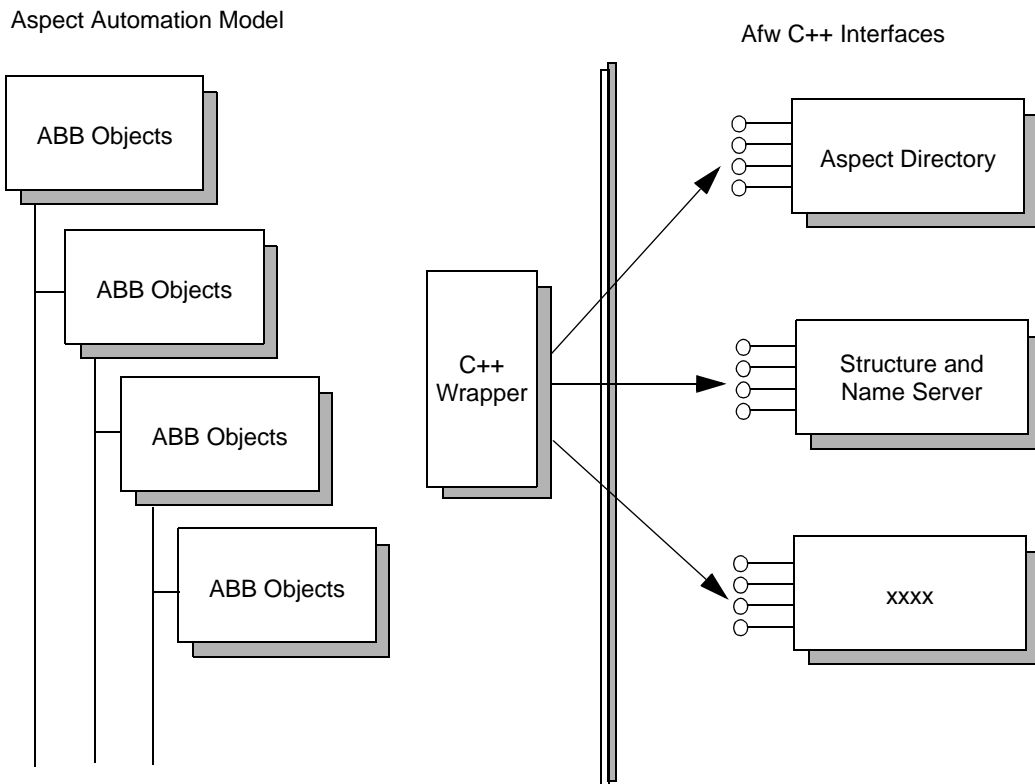


Figure 5-2. Aspect Framework Layers

5.2.1 Afw Interfaces

One way of adding functionality to Industrial^{IT} is to write an application towards the so called Afw Interfaces. With these interfaces it is possible to create almost any functionality for an aspect system.

You should always use the Aspect Objects AppWizard when programming aspects in C++. It provides templates and solutions for many problems, and saves you a lot of work. It also shows you how to use the right helper classes.



You should always write AfwServices in C++. There is no Automation Model or tools support for writing AfwServices in Visual Basic.

5.2.2 Automation Models

The interfaces starting with IAfw, like IAfwObject or IAfwTranslate, are part of the inner core (Afw Core Model). They are powerful and flexible, but also C++ oriented and require a good understanding of the system.

5.2.2.1 Aspect Automation Model

Aspect Automation Model is an easy to use layer around the Aspect Object model. It makes it easier to write code using the Aspect Integrator Platform without expert knowledge about internal architecture and interfaces. See [Figure 5-2](#). It fully supports OLE Automation, so that it can be used from Scripting Languages like VB-Script or Java-Script. Aspect Automation Model objects typically begin with ABB; e.g ABBAspect and ABBOject.

5.2.2.2 Workplace Automation Model

The Workplace Automation Model makes it possible to develop some enhanced user interface features. Extended window management and context menu handling are among these features.

The Workplace Automation Model provides the aspect system developers with an intuitive object model of the Workplace portion of the Aspect Integrator Platform API.

The Aspect Automation Model wraps the Aspect Integrator Platform API through objects, and collections of objects, rather than through a set of services.

5.3 Thin Client

Web-enabling the Aspect Integrator Platform requires three levels of support:

- Web Server, an additional machine, in the server layer, that contains the web server software, Microsoft Internet Information Server. Additional functionality is added to this web server:
 - Mapping of aspects as URLs.
 - Identification and mapping of client type.
 - Mapping of user authentication to AIP Security concepts.
 - Web services and web pages from web-enabled aspect systems.
- Web Workplace, an extension of the rich client workplace, provides the basic functionality needed to browse an Industrial^{IT} system.
- Web-enabled aspect systems, can be viewed or expose their data through the web server.

The Web Server, the Web Workplace and some web-enabled aspect systems, are all part of the Thin Client add-on. Furthermore, Aspect Studio provides support to develop additional web-enabled aspect systems.

Note that the aspect automation model is web-enabled and exposed in forms of web services for client scripting.

5.3.1 Web-enabling aspect systems

There are several ways to web-enable an aspect system:

- Creating a web service that exposes your data will provide a data view of your aspect system.
- To make your aspect system viewable on a thin client, you have two preferred choices:
 - **Reference to a .Net Control**
This is the preferred and most flexible method for a PC based Thin Client. Basically, the aspect system only answers the request to present itself with a reference to a component already installed on the web server.
 - **Reference to a asp.Net web page**
This is the preferred and most flexible method for PDAs or WAP phones. Basically, the aspect system only answers the request to present itself with a reference to a web page already installed on the web server.

It is also possible to expose your aspect system as XML or to use the provided functionality to route requests to another web server.

Web services and Web pages are best implemented using C#.

Support for web-enabling aspect systems is included in Aspect Studio.

Section 6 Integration Tools

6.1 Introduction

This section describes tools and products provided by Integrate^{IT} product suite.

The Aspect Integrator Platform provides a large set of aspect systems that make it possible to create applications without programming, see [Section 3, Predefined Aspect System](#), for more information on:

- User Interface Wrapping
- Calculations and Transfer Aspects
- Report and Message support
- NLS Support

As you have already read in [Section 5, Implementing an Aspect System](#), there are two main paths for Aspect Integration using tools and programming techniques:

- Visual C++ together with Aspect Objects AppWizard, see [Aspect Objects AppWizard](#), on page 66.
- Visual Basic together with Aspect Express and using the Automation Models to access the AIP mechanisms. See [Aspect Express](#), on page 74.

There are also some smaller paths that can be useful for small applications such as:

- Writing Java and VB scripts that utilize the Automation Models, providing an entry level to aspect integration and interfacing.
- Using the optional Graphics Builder - a tool designed for display and display aspect building, that can be used to build new aspects together with the Automation Models.

To be able to build complete system products, you need Aspect Integrator Platform and Aspect Studio.

Aspect Studio includes a large number of tools to support your configuration management and programming efforts. See [Aspect Studio](#), on page 66 for details.

6.1.1 Aspect Integrator Platform

Aspect Integrator Platform is composed of two packages:

1. Package 1: Aspect Integrator Platform SDK
2. Package 2: Aspect Integrator Platform Run-time (only)

Package 1 contains a complete environment for development and run-time testing of end-customer products based on the Aspect Integrator Platform.

Package 2 is distributed with the end-customer product and will, when installed on the end-customer site, provide the run-time functionality, of Aspect Integrator Platform, necessary for the end-customer product that was developed using package 1.

The Aspect Integrator Platform contains the core functionality that you need in order to develop new Industrial^{IT} System Products or System Product Extensions.

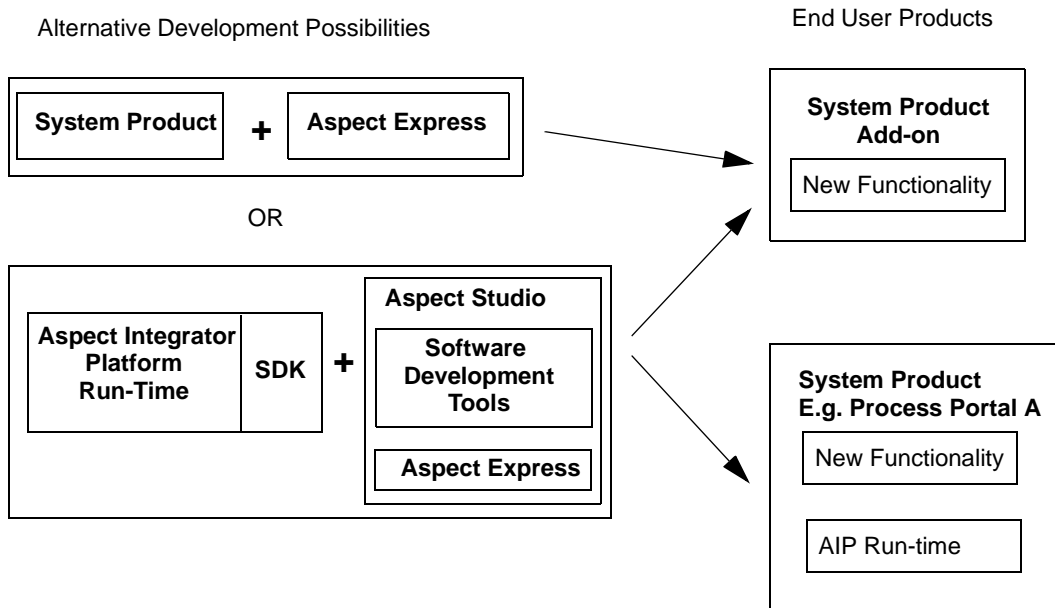


Figure 6-1. Creating AIP Products

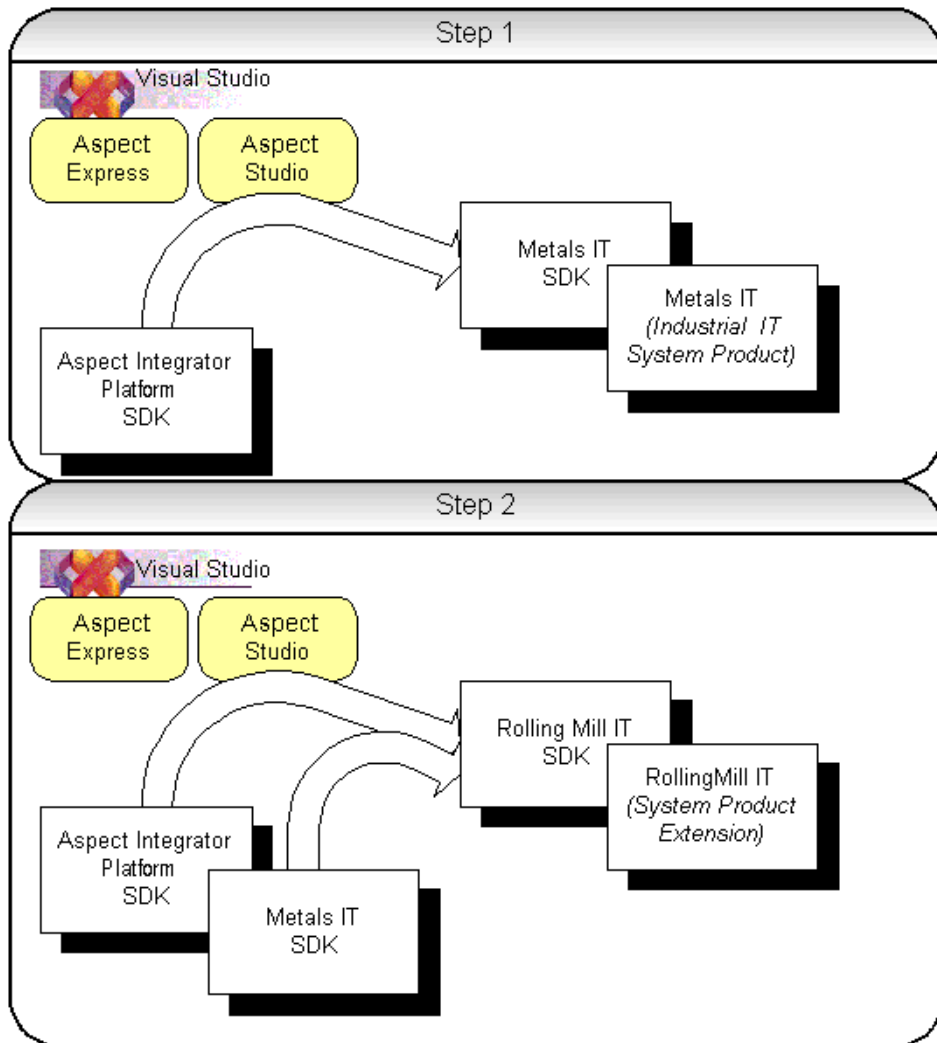


Figure 6-2. Utilizing SDK:s to Create new System Products and System Product Extensions

The following example refers to [Figure 6-2](#):

You have Aspect Integrator Platform SDK installed on your development system. As a first step you develop a new system product “Metals^{IT}” that extends the functionality of Aspect Integrator Platform.

As a second step the Rolling Mill people will further extend the functionality by using AIP SDK and Metals^{IT} SDK.

More generally, any Industrial^{IT} system product (AIP, Metals^{IT} etc.) will be accompanied with an SDK so that you can extend the functionality of it with additional components. See also [Figure 6-1](#).

6.2 Aspect Integrator Platform run-time

Aspect Integrator Platform run-time contains the executable environment for Industrial^{IT} system products.

As already described in [Section 3, Predefined Aspect System](#), AIP also contains a lot of predefined aspects and tools, making it possible to develop applications without programming, just configuration. For details, please see *Integrate^{IT} Predefined Aspects Reference Manual*.

AIP provides the Automation Models, useful on the run-time platform for access from applications implemented with Aspect Express or from Java and VB script applications. See [Section 5, Implementing an Aspect System](#), for details on the Automation Models and *Integrate^{IT} Aspect Integrator Platform Automation Models Programmer's Guide*.

6.3 Aspect Integrator Platform SDK

Aspect Integrator Platform provides the complementary environment (headers, libraries, include files etc.) to Industrial^{IT} systems, that you need to do anything from developing a single Aspect system, in C++, to creating a complete Industrial^{IT} system product.

To automate the process of creating new products based on other products SDKs, and to maintain the <product> + <product> SDK structure of any Industrial^{IT} product, there is a set of tools available in the Aspect Studio product. An SDK consists of the following additional directories on top of the directories and files installed for each functional subsystem in an ‘executable’ Industrial^{IT} system product:

- Include: include files
- Lib: library files
- Bin: tools, debug builds of the Aspect Integrator Platform.

Aspect Studio supports both release and debug build/packaging of products including SDK and reference source code (ref_src), which means that you can debug your own applications together with the system product.

- Idl: IDL files
- Doc: technical documents

A number of functional descriptions, test descriptions and such technical documents are included in the SDK.

- Help: help files
- Samples: test examples
- Import: sometimes there can be export/import files useful for the developers.

6.3.1 SDK Tools

There are a number of tools on the SDK that can be very useful for your developers. Most of the tools are command line tools. The tools are described in the Aspect Integrator Platform Programmers Guide, as most of them are developer tools.

6.3.2 SDK Public Interface documentation

The SDK contains documentation of the public interfaces available for the developers. The documentation consists of on-line help as well as a set of word documents. See *Integrate^{IT} Aspect Integrator Platform Programmer's Guide* for details.

6.4 Aspect Studio

Aspect Studio consists of the following software functions:

- Aspect Express, see [Product Viewer](#), on page 73.
- Aspect Objects AppWizard
- Thin Client Support
- Environment set-up
- Build Manager
- AutoBuild
- Product Packager
- Product Viewer

For help with choosing between Aspect Express and Aspect Objects AppWizard tools, see [Section 4, Aspect Integration](#) and [Section 5, Implementing an Aspect System](#).

6.4.1 Aspect Studio Wizards

6.4.1.1 Aspect Objects AppWizard

Aspect Objects AppWizard creates C++ templates for the most important functions of an aspect system, and increases the productivity of C++ aspect developers. They can start with the framework of the application already implemented.

The Aspect Objects AppWizard allows you to create new Aspect systems. You do this by selecting the desired functionality in a series of dialogs.

The Aspect Objects AppWizard makes use of COM and ATL technology. Much of the interaction with the system is hidden in templates and helper classes which means that if something in the framework changes, the code produced by the wizard will still be valid.

First, run the Aspect Objects AppWizard. This creates an ATL application with the necessary include files, IDL files and a configuration file (.add) that you will use to register your aspect system.

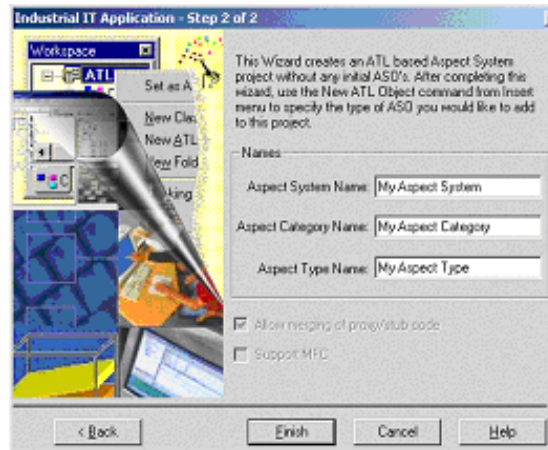


Figure 6-3. Running the Aspect Objects AppWizard

Then, continue by creating the ASOs that you need in your application. Your choice of ASOs depends a lot upon the complexity of your application. You can create the following four types of ASOs:

- **Aspect View**
This ATL object implements a view ASO that can be viewed in the Industrial^{IT} workplace. It implements the UI functionality of the Aspect system.
- **Aspect Page**
You use this ATL object when you build standard configuration (tabbed) dialogs. This includes standardized buttons and behavior for “Apply”, “Cancel” and “Help”.
- **Aspect Verb**
You use this ASO when you add verb items to the context menu of the workplace. This ASO should be as “thin” as possible. You do not want to start heavy data ASOs just to show some verbs!
- **Data ASO**
This is the ASO that handles all data storage.

From here you start refining the ASOs with more and more detailed functionality. Please read the *Integrate^{IT} Aspect Integrator Platform Programmer's Guide* for details.

There are very few limitations to the functional levels that you can achieve using the Aspect Objects AppWizard and the *Integrate^{IT} Aspect Integrator Platform Programmer's Guide* together with the Aspect Integrator Platforms public C++ interfaces which are described through the on-line help-files.

6.4.1.2 Thin Client Support

Aspect Studio supports Thin Client development as is. However, in order to test and execute thin client solutions, it is required that the Thin Client add-on SDK is installed.

The support includes handling of projects created in Visual Studio 7 (.Net), and affects most tools.

Aspect system development support is provided through two additional wizards and an extension of the Aspect Objects AppWizard:

- Aspect Object Appwizard. Extended with a thin client checkbox. When checked the created aspect system is provided with web interfaces.
- Aspect Web Pages. Supports the creation of the web page or control that is able to present the aspect system.
- Aspect Web Service. Supports the creation of the web service that is able to expose the aspect systems data.

These wizards require that Visual Studio 7 (.Net) is installed.

6.4.2 Configuration Management Tools

6.4.2.1 General

Aspect Studio contains a number of tools that will help the configuration managers and developers to do their work as effectively as possible. Please read the following short descriptions of the respective tool and then study.

6.4.2.2 Environment set-up

Environment set-up helps your developers to set-up their development environment based on the products they intend to create and the Industrial^{IT} components they intend to include in it. This saves a lot of valuable time and the development can start with a minimum of work.

This is the first step in the development cycle where you use Aspect Studio.

6.4.2.3 Build manager

Build Manager provides support for your configuration managers and for your developers. With Build Manager you can take care of version control and manage the build of families of Microsoft Visual Studio components. You can create baselines and specify build of components in a batch oriented way utilizing the Visual Source Safe capabilities.

Build Manager provides support for your configuration manager(s) and for your developers. With Build Manager your configuration managers can take care of version control and manage the build of families of Microsoft Visual Studio components. Baseline creation and build of components in a batch oriented way while utilizing Visual Source Safe capabilities as good as possible are some of the key features.

6.4.2.4 AutoBuild

Your developers and configuration managers use AutoBuild to compile and build the application and system products and produce the merge modules for the Microsoft installer (MSMs). AutoBuild is an add-in to Visual Studio.

The compilation and building of the resulting products into so called MSM packages is done with AutoBuild which is a plug-in to Visual Studio. It utilizes the definitions done with Build Manager and produces the input required by Product Packager.

6.4.2.5 Product Packager

In brief you use the Product Packager to generate software distribution packages from MSM files. The generated distribution packages contain everything that Windows Installer require during the installation, re-installation and un-installation of the generated products.

You can also use Aspect Studio to combine the aspect systems you develop using Aspect Express or Aspect Objects AppWizard with other aspect systems and Aspect Integrator Platform components into new product packages.

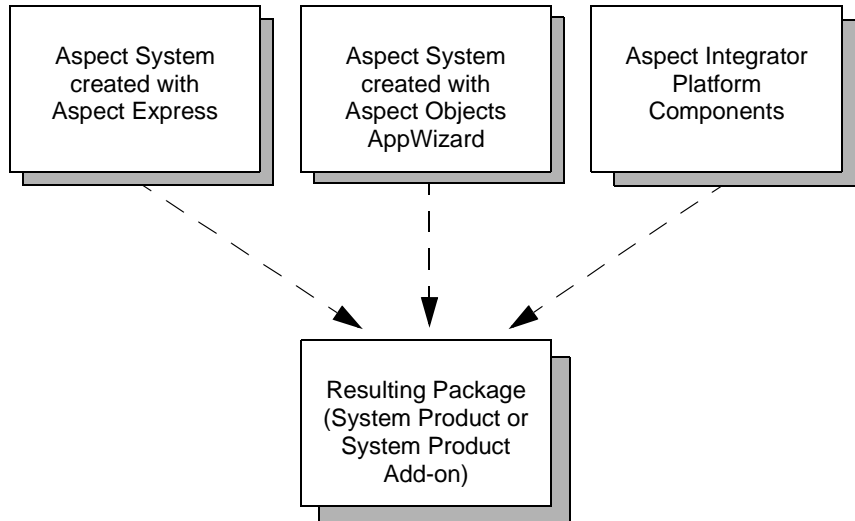


Figure 6-4. Creating new Industrial^{IT} System Products using Product Packager

When the products are distributed you can take advantage of the service pack features to package and distribute software upgrades to your customers. They in turn can install and uninstall the software and the service packs, helped by the Product Viewer to inspect the versions and service packs that are installed of the respective products. Product Packager has the following main features:

- **Creation of distribution packages**
Product Packager lets you generate distributions interactively or in batch. The result will be the same, but generating in batch mode lets you generate a product package from a script. Product Packager creates a structure that is complete in the sense that it is ready for installation on a target node by using the Microsoft Windows Installer (MSI).
 - Normally, a distribution package is delivered on CD-ROM. The distribution CD will contain a setup program that handles the installation. After you have started the setup program, you will be guided through the complete installation procedure. Sometimes there can be a need to re-install, un-install, or change a product package. The functionality required for that is provided by the Windows Installer.
- **Creation of Service Packs**
Product Packager supports building of service packs. A service pack can then be distributed via a suitable media, normally a CD-ROM, or made available through Internet. By installing a service pack you can correct a software and have it work the same way as if you install a complete new version of the software.
- **Creation of Extension Packs**
Product Packager supports building of extension packs. Extension packs can be distributed via suitable media, normally CD-ROM, or made available through Internet. Extension packs are installed for the adding of software components, such as NLS for a new language.

6.4.2.6 Product Viewer

You use Product Viewer to view which software, which service packs and which versions that are installed.

The Product Viewer provides you with the possibility to inspect not only the Industrial^{IT} software you have installed, but also other vendors software installed with Windows Installer.

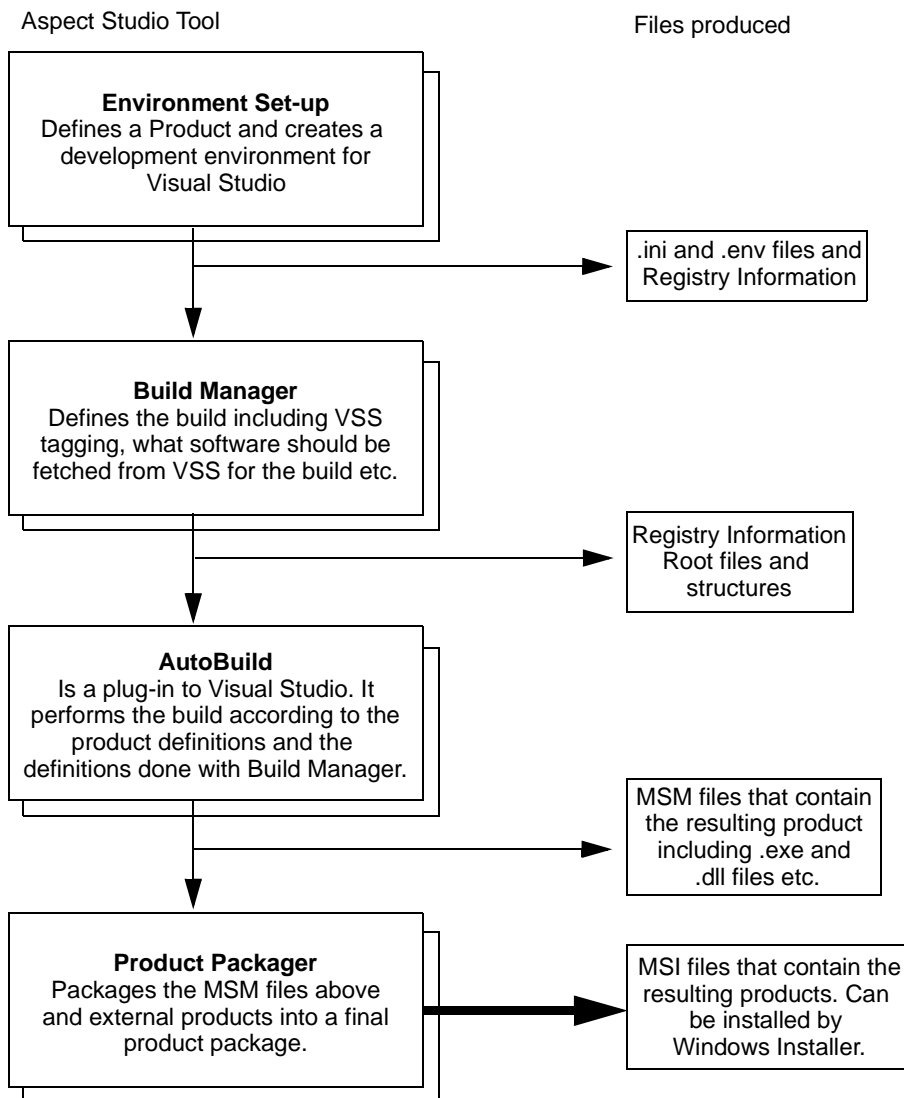


Figure 6-5. Aspect Studio Tools - Workflow

6.5 Aspect Express

Aspect Express provides a Wizard for Aspect system creation that can be run inside or outside Visual Basic. It increases the productivity and simplifies the integration of new or existing ActiveX-based applications into Industrial^{IT}. It is a smart tool, for fast and simple creation of Aspect systems, that provides data persistency, creates Aspect views and verbs and makes ActiveX properties available as OPC properties. It can also be used to configure COM components as server aspects.

Another functionality that is very useful is the ability to build and package the complete Aspect system as merge modules for the Microsoft installer (MSMs), ready to be installed on any Windows 2000 machine with Industrial^{IT} products installed.

Aspect Express, Aspect Automation Model and Visual Basic can be used to implement almost any Aspect Object functionality. The public C++ interfaces of the Aspect Integrator Platform can also be used to some extent, to further extend the functionality.

6.5.1 Aspect Express Builder

Aspect Express provides a configuration tool - Aspect Express Builder - that simplifies the integration of new or existing ActiveX-based applications into Industrial^{IT} system product extensions. You run the configuration tool from Visual Basic or as a stand-alone tool.

You can use the Aspect Express Builder to specify the following functionality (works both for third party ActiveX components as well as for the ActiveX components that you have developed yourself):

- Aspect data storage - functionality to store the Aspect properties in the Aspect directory (persistency).
- Aspect Views - functionality to visualize Aspect GUIs.

- Aspect Verbs - functionality to provide Aspect and Object Verbs, see the example in [Figure 6-6](#).
- Make ActiveX attributes available as OPC properties - any OPC client can read or write to the public attributes using the OPC standard.
- Server Aspect component - if one of the Application components needs to execute continuously.

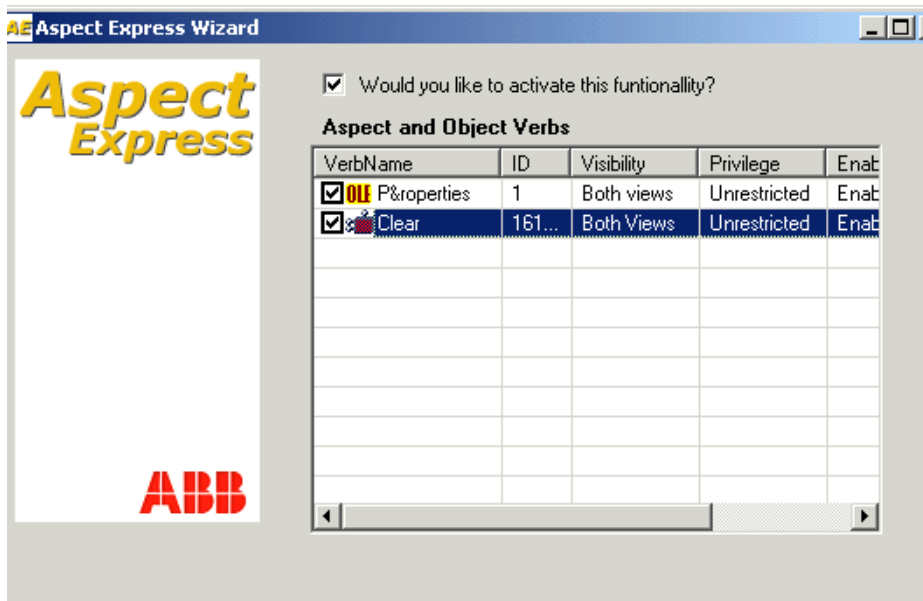


Figure 6-6. New Component Wizard - Aspect Verbs Configuration

Each of the above functions can be individually selected. Furthermore, if your application consists of several COM components, each one of them can have its individual set-up.

Aspect systems created with Aspect Express are installed as system extensions.

The following functionality is provided by Aspect Integrator Platform as long as the data storage is done in the Aspect Directory (see [Integration levels](#), on page 39 for more information):

- Support for copy and paste.
- Support for import and export.

Use Visual Basic and the ready made examples for Aspect Express to add the following functionality to the aspect system:

- On-line help on the aspect system (NLS support is provided)
- Application logs for error tracing
- System messages for information to the end users (NLS support is provided)

You can also combine Aspect Express usage with the usage of Visual Basic programming either utilizing the Aspect Automation Model or the core IAfw interfaces (many can be used from VB).

6.5.2 Aspect Express Packager

When you have finished developing your aspect system, you can use Aspect Express Packager tool to build an installation. You can then install the aspect system on any Industrial^{IT} product and it can be executed without having Aspect Express installed.

Aspect Express Packager can also produce install components that can be combined with Aspect Object AppWizard solutions implemented in C++. The combined package is produced by using Product Packager in Aspect Studio.

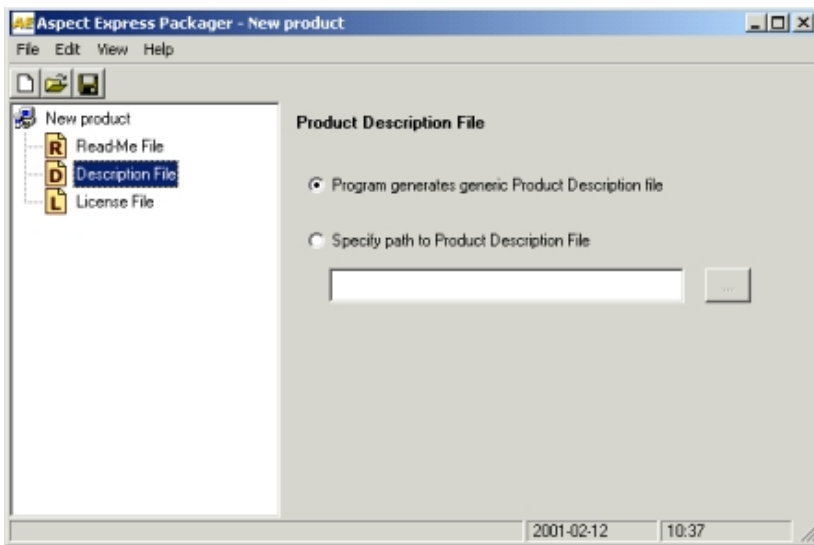


Figure 6-7. Using the Aspect Express Packager

INDEX

A

Afw Interfaces 57
Afw Services 27
Alarm and Event 38
ASO 27
Aspect Automation Model 25, 54
 OPC properties 45
 Overview 57
 With Aspect Express 74
Aspect Express 54, 73
 Details 73
 Level 0 40
 Level 2 43
 Level 3 47
 Product Overview 23
Aspect Framework 55
Aspect Integrator Platform 21, 23
Aspect Objects AppWizard 54, 57
Aspect Studio 54
 Build Manager 69
 Environment set-up 69
 Product overview 23
 Product Packager 70
 Level 4 48
Aspect System Objects 27
Aspect Systems 26

B

Bookmark Aspect 31

D

Data Access 38

G

Graphics Builder 54

H

History 38

I

IAfw 56
Integration Levels 39
 Level 0 40
 Level 1 42
 Level 2 43
 Level 3 45
 Level 4 48
 Level 5 50

O

OPC AE 38
OPC DA 38
OPC HDA 38

S

Server Aspects 27

V

Visual Basic or C++? 53

W

Web Server 58
Web Workplace 58



3BSE 023 960R201. Printed in Sweden June 2002
Copyright © 1999-2002 by ABB. All Rights Reserved
® Registered Trademark of ABB.
™ Trademark of ABB.

<http://www.abb.com/control>

Automation Technology Products
Västerås, Sweden
www.abb.com/processautomation
email: processautomation@se.abb.com

Automation Technology Products
Wickliffe, Ohio, USA
www.abb.com/processautomation
email: industrialitsolutions@us.abb.com

Automation Technology Products
Mannheim, Germany
www.abb.de/processautomation
email: marketing.control-products@de.abb.com