

## Introduction to the ike Software Development Kit (SDK)

### 1 Introduction

The present document provides a brief introduction to the ike Software Development Kit (SDK). The SDK provides a powerful and flexible Application Programmers Interface (API) that allows developers to create fully customised Data Capture Applications (DCAs) for ike, and Desktop Synchronisation Applications (DSAs) that synchronise information captured using ike with a central database.

### 2 Intended audience

The SDK is aimed at developers with experience in Microsoft Visual Basic (VB) or Microsoft Embedded Visual Basic (eVB). The SDK can also be used with Microsoft Embedded Visual C++ 4.

### 3 SDK Overview

The SDK includes an Embedded Framework, and a Desktop Framework. Figure 1 illustrates how the ike Embedded Framework is used to create DCAs, and similarly how the ike Desktop Framework is used as a basis for DSAs.

DCAs are developed using eVB (or ArcPad Application Builder) on a Desktop PC running Microsoft Windows XP, and then these application can be copied to, and executed on ike. DCAs generally guide the user through a series of frames that will make use of the instruments within ike to capture the required information.

DSAs are developed using VB on a Desktop PC running Microsoft Windows XP, and are executed on the Desktop PC to synchronise previously captured information.

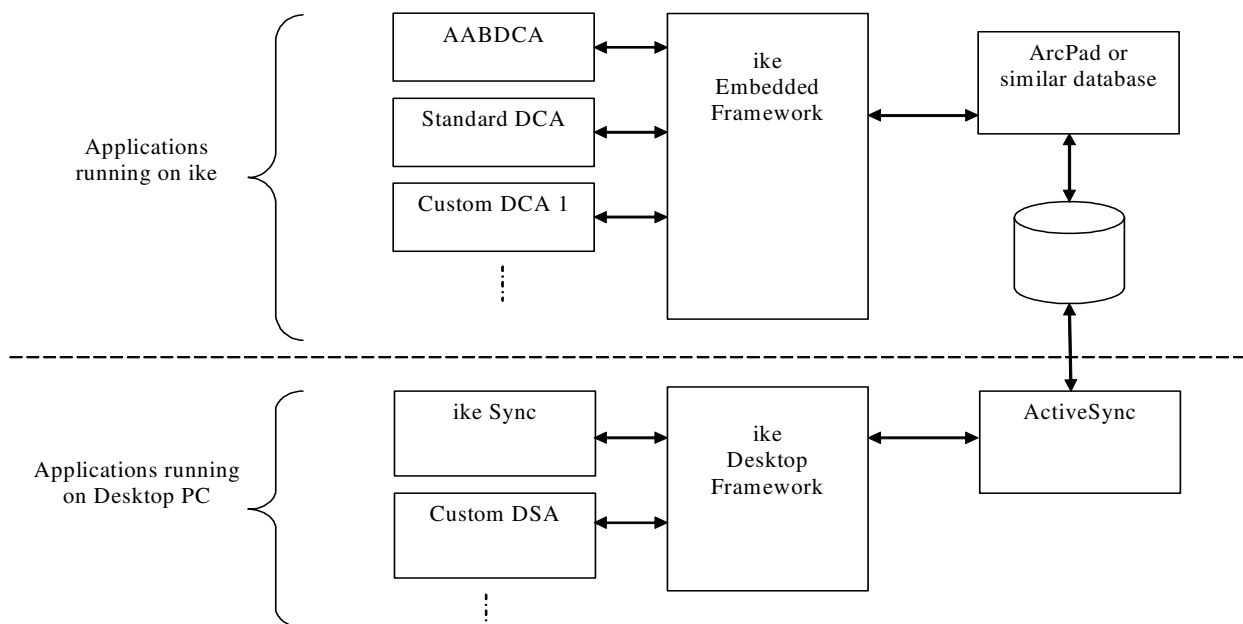
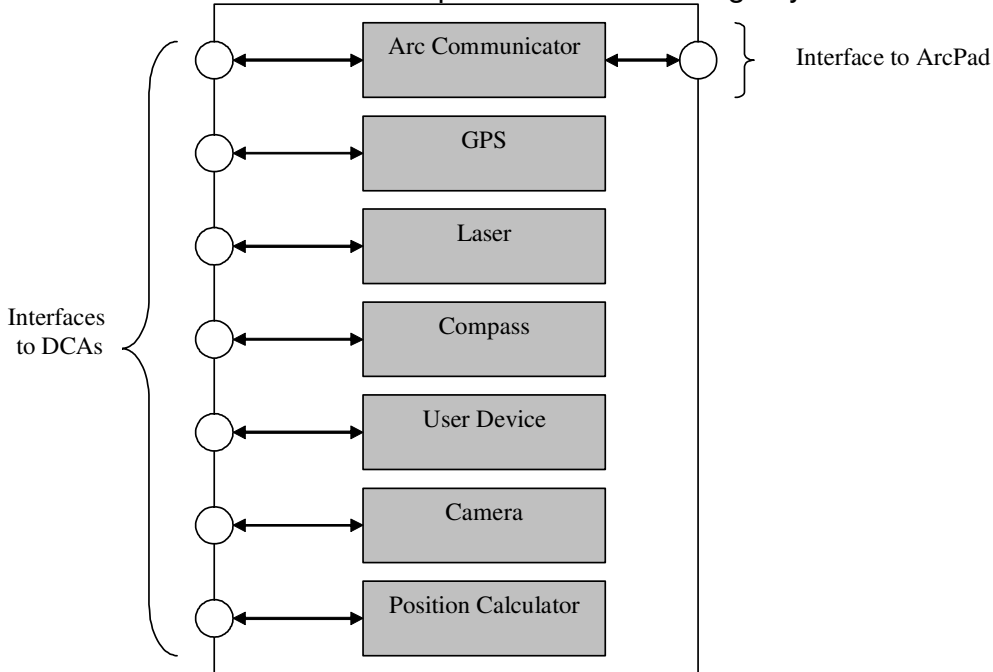


Figure 1 - Overview of SDK environment

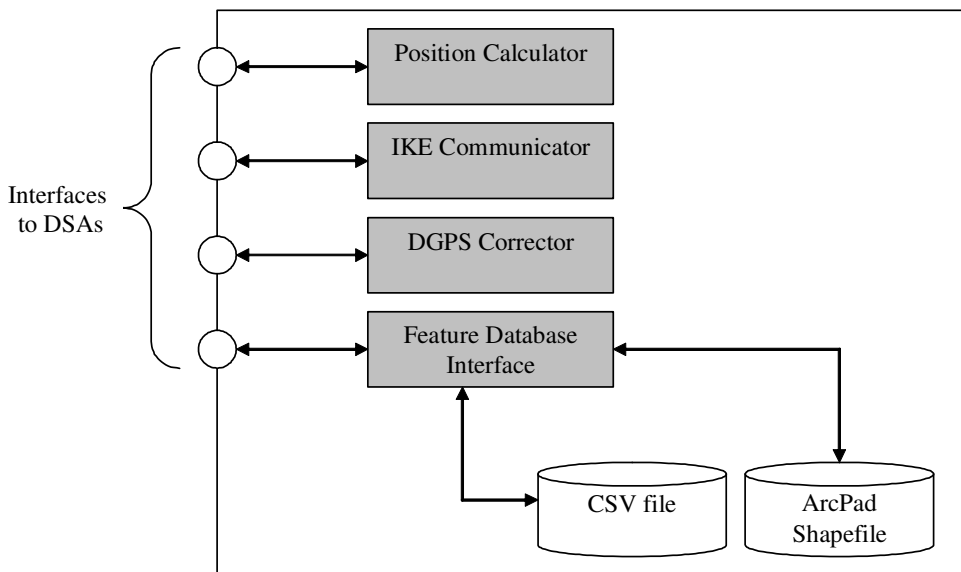
#### 4 ike Embedded Framework Overview

The ike Embedded Framework provides the following key modules. Each module is an ATL COM control.



#### 5 ike Desktop Framework Overview

The ike Desktop Framework provides the following key modules. Each module is an ATL COM control.



## 6 User Manual

The SDK provides a module corresponding to each instrument within ike, allowing each device to be controlled as required by the DCA. Each module includes a number of 'Interface Components', which are described in detail in the SDK User Manual.

### 6.1 Interface components

Each module provides a set of interface components (Methods, Properties, and Events) that are accessible from eVB in the embedded framework, or VB in the desktop framework. Each of these interface components is documented in detail using a common format consisting of:

1. A description of the Purpose of the interface component.
2. Advertised Requirements, or preconditions that must be met (either before the interface component is used, or restrictions on the parameters that are passed).
  - a. For Methods and Properties, the requirements usually include a description of any other methods that must be called, or configuration that must be established prior to calling the method or using the property being described. Also, any restrictions on the parameters passed to the method or property are described.
  - b. For Events, there is usually nothing that the developer using the interface can do prior to the event being generated, so there are not usually any requirements.
3. Advertised Promises, or postconditions that the interface component guarantees will be met after use of the interface component if the advertised requirements were met.
  - a. For Methods and Events, the promises generally describe events that are guaranteed to be true after the method is called, or the property is accessed.
  - b. For Events, the promises generally describe the range of valid parameter values that may be provided in the event, or any configuration that will be in place before the event is fired.

If the requirements are not met when an interface component is used, the promises usually will not be met either.

### 6.2 Example interface component

The Interface Component 'ShowMeter' from the Laser module is provided to illustrate the nature of the SDK documentation. To use this interface component from within an eVB application, the developer would use code similar to the following:

- Laser1.ShowMeter (True, False)

#### 6.2.1 ShowMeter(BOOL ShowIndicator, BOOL ShowValue)

##### 6.2.1.1 Purpose

- Enables the graphical display for the laser. The features visible are determined by the given parameters.

##### 6.2.1.2 Requirements

- *Initialise* must be called before using this method.

##### 6.2.1.3 Promises

- Displays current laser reading on graphical indicator iff ShowIndicator is true

- Displays current laser reading on numerical display iff ShowValue is true
- If the *Initialise* method has not been called, the *Error* event is fired (ModuleNotInitialised, "ShowMeter", "Laser not initialised")
- If ShowValue and ShowIndicator are both false:
  - Disables streaming
  - Clears the laser control
- Otherwise:
  - Enables streaming

## 7 Data Capture Applications

The SDK includes full Source Code for two working Data Capture Applications. All subroutines and functions in the examples are documented using the same strategy as the SDK interface components (Purpose, Requirements, and Promises). These examples are intended as tutorial examples for developers wishing to develop their own DCAs, and provide a good basis for development to get a customised DCA up and running quickly.

### 7.1 The Standard DCA

The Standard DCA provides a simple set of screens allowing the user to capture information and store it in a Comma Separated Value (CSV) file. The application illustrates how to manage a series of forms that enable all of the ike instruments, capture a point (which includes Image, GPS position, Laser distance, Compass Bearing, and calculated Target position), and store this information to a CSV file.

### 7.2 The AABDCA

The ArcPad Application Builder DCA is very similar to the Standard DCA, with the exception that the information captured is stored to an ArcPad Database. The series of screens provided are identical to those in the Standard DCA, except that ArcPad can be used to view the captured information on a map, to allow easy access to previously captured information while in the field.

## 8 Contact information

For further information, questions or comments, please contact Surveylab:

Telephone: +64 4 382 8064  
Fax: +64 4 382 8205  
Email: [support@survey-lab.com](mailto:support@survey-lab.com)  
Mail: P O Box 6529, Wellington, New Zealand