

COMIZOA

EtherCAT Software Development Kit

TEST & MEASUREMENT & AUTOMATION

COMIZOA S/W ETHERCAT MASTER

API REFERENCE MANUAL

- DIGITAL I/O MODULE -

JAN 2017
P/N 0126-2017-01
© 2017 COMIZOA Inc. All rights reserved

API Reference Manual

ComiSWECAT MASTER Manual

Copyright © 2017 by COMIZOA, Inc. All rights reserved.

COMIZOA owns all right, title and interest in the property and products described herein, unless otherwise indicated. No part of this document may be translated to another language or produced or transmitted in any form or by any information storage and retrieval system without written permission from COMIZOA. COMIZOA reserves the right to change products and specifications without written notice. Customers are advised to obtain the latest versions of any product specifications.

COMIZOA MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OTHER THAN COMPLIANCE WITH THE APPLICABLE COMIZOA SPECIFICATION SHEET FOR THE PRODUCT AT THE TIME OF DELIVERY. IN NO EVENT SHALL COMIZOA BE LIABLE FOR ANY INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES AS A RESULT OF THE PRODUCT'S PERFORMANCE OR FAILURE TO MEET ANY ASPECT OF SUCH SPECIFICATION. COMIZOA PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN LIFE SUPPORT APPLIANCES, DEVICES OR SYSTEMS WHERE A MALFUNCTION OF A COMIZOA DEVICE COULD RESULT IN A PERSONAL INJURY OR LOSS OF LIFE. CUSTOMERS USING OR SELLING COMIZOA DEVICES FOR USE IN SUCH APPLICATIONS DO SO AT THEIR OWN RISK AND AGREE TO FULLY INDEMNIFY COMIZOA FOR ANY DAMAGES RESULTING FROM SUCH IMPROPER USE OR SALE.

Information contained herein is presented only as a guide for the applications of our products. COMIZOA does not warrant this product to be free of claims of patent infringement by any third party and disclaims any warranty or indemnification against patent infringement. No responsibility is assumed by COMIZOA for any patent infringement resulting from use of its products by themselves or in combination with any other products. No license is hereby granted by implication or otherwise under any patent or patent rights of COMIZOA or others.

COMIZOA software and its documentation are available only under the terms of a Master Software Use and Support Agreement.

Trademarks

The COMIZOA logo is a registered trademark. All other brand names, product names, trademarks, and registered trademarks are the property of their respective owners.

Visit our web page at <http://www.comizoa.com>

For support requests, contact us at csteam@comizoa.com

For documentation suggestions, corrections, or requests, contact tech@comizoa.com

고객(顧客) 기술 지원

전자 우편 : csteam@comizoa.com

파일 서버 : [ftp.comizoa.com](ftp://ftp.comizoa.com)

웹 사이트 : <http://www.comizoa.com>

본사 안내

대전 광역시 유성구 테크노 2로 314 번지 (탑립동 914 번지)

전화번호 : 042-936-6500

모사전송 : 042-936-6507

COMIZOA Software EtherCAT Master API Reference

© 2017 COMIZOA

All Rights Reserved. No Part of this publication may be reproduced, stored in retrieval system or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission, in writing, from the publisher.

Table of Contents

Trademarks	2
Tble of Contents	3
Introduction	6
1 ComiSWECAT 사전 안내 사항	7
1.1 Overview	7
1.1.1 제품 보증 안내	7
1.1.2 제품 보증 규정	7
1.1.3 저작권	7
1.1.4 상표안내	7
1.1.5 주의사항	7
1.1.6 매뉴얼 용어 안내	8
1.1.7 매뉴얼 아이콘의 설명	8
1.2 Features	10
1.2.1 독립성	10
1.2.2 호환성	10
1.2.3 편의성	10
1.2.4 신뢰성	10
1.2.5 풍부한 예제와 신속한 기술 지원	10
Before working with ComiSWECAT	11
2 EtherCAT & ComiSWECAT	12
2.1 EtherCAT 소개	12
2.2 ComiSWECAT 개요	12
2.3 함수 이름 규칙 가이드	12
Development Environment for ComiSWECAT	13
3 개발 환경 별 ComiSWECAT 사용 안내	14
3.1 개발 환경 지원 안내	14
3.2 ComiSWECAT 인터페이스 구성 파일	15
3.3 각 개발 환경 별 안내	15
3.3.1 Visual C++ 6.x 개발자를 위한 안내	16

3.3.2	<i>Visual C++ 10.x 개발자를 위한 안내</i>	22
3.3.3	<i>Borland C++ Builder 개발자를 위한 안내</i>	28
3.3.4	<i>Borland Delphi 개발자를 위한 안내</i>	33
3.3.5	<i>Visual Basic 개발자를 위한 안내</i>	37
	<i>ComiSWECAT Introduction</i>	41
4	<i>ComiSWECAT 소개</i>	42
4.1	<i>함수의 명명 규칙</i>	42
4.2	<i>데이터형 표기</i>	42
	<i>DLL Functions</i>	44
5	<i>DLL Functions</i>	45
5.1	<i>함수 요약</i>	45
5.2	<i>함수 설명</i>	46
	<i>General Functions</i>	49
6	<i>General Functions</i>	50
6.1	<i>함수 요약</i>	50
6.2	<i>함수 설명</i>	51
	<i>Net Interface Functions</i>	54
7	<i>Net Interface Functions</i>	55
7.1	<i>함수 요약</i>	55
7.2	<i>함수 설명</i>	56
	<i>Slave I/F Functions</i>	57
8	<i>Slave I/F Functions</i>	58
8.1	<i>함수 요약</i>	58
8.2	<i>함수 설명</i>	59
	<i>Digital I/O Functions</i>	64
9	<i>Digital I/O Functions</i>	65
9.1	<i>함수 요약</i>	65
9.2	<i>함수 설명</i>	66
	<i>ComiSWECAT Error Code</i>	77
I	<i>ComiSWECAT Error Code 안내</i>	78
	<i>Frequently Asked Questions</i>	79
II	<i>Frequently Asked Questions (FAQ)</i>	80
II.I	<i>Visual Studio 2005</i>	80

II.II Visual Basic	82
II.III Borland C++ Builder	82
<i>Index of ComiSWECAT Functions</i>	<i>88</i>
III Quick Reference to ComiSWECAT Functions	89

Introduction

본 장에서는 제품 보증안내를 비롯한 제품 보증 규정, 저작권, 상표안내, 주의사항을 설명하고 있습니다. 다양한 모션 제어 환경에서 보다 강력하고 빠른 모션 제어를 위해, 그리고 안정적인 모션을 위해 고객(顧客)님께서 선택하신 저희 (주)커미조아 제품은 이제 고객(顧客)님에게 큰 감사와 보답으로 이바지 하도록 하겠습니다.

가 장 안정적이고, 빠르고 정확한 모션, 그리고 고객(顧客) 중심에서의 제품의 완성을 추구하는 (주)커미조아는 고객(顧客)님들을 위해 언제나 최선과 정성을 다하는 자세로 지금 이 시간에도 보다 나은 제품 개발을 위해 성실과 열정을 바탕으로 제품에 꿈을 담고 있습니다.

먼저 고객(顧客)님들께, 국내 최고의 모션 기술력을 자랑하는 저희 (주)커미조아 제품을 선택하여 주신 여러분들께 다시 한번 감사의 말씀을 드리며, 본 장에서는 제품 보증안내, 규정, 저작권, 상표 안내에 대한 내용을 설명하고 있습니다. 본 장에서 설명드리는 내용은 ECAT MASTER 운용과는 별개의 내용이나, 제품의 보증과 규정 그리고 저작권에 대한 내용을 설명 드리고 있으므로, 반드시 습득하여 주시기 바랍니다.



1 ComiSWECAT 사전 안내 사항

1.1 Overview

1.1.1 제품 보증 안내

저희 (주)커미조아는 고객(顧客) 여러분들께 가장 안정된 소프트웨어와 하드웨어를 공급함으로써, 고객(顧客) 여러분들을 만족시켜드리는 것을 최우선의 목표로 하고 있습니다. 만약 구입하신 제품에 외관상의 하자, 동작이상 또는 불량이 발견되는 경우에는 언제든지 저희 (주)커미조아를 통해 문의(問議)해주시기를 바라며, 가까운 대리점 혹은 총판점을 통해 구입하신 경우에는 해당 구입처(購買處)를 통해 문의하시면, 더욱 빠른 기술 지원을 받으실 수 있습니다.

1.1.2 제품 보증 규정

구입하신 당사의 제품은 소비자의 과실 이외의 자체 결함 및 동작이상에 대해 2 년간 그 전체 혹은 일부에 대해서 보증하고 있습니다. 당사의 제품에 대한 자세한 제품 보증 규정은 별도로 관리되는 각 제품의 '제품 보증 규정' 에 의거하며, 자세한 보증 규정을 알기 원하시는 경우 (주)커미조아 혹은 총판점(總販店) 및 대리점(代理店) 등 해당 구입처를 통해 문의해 주시기 바랍니다.

1.1.3 저작권

이 매뉴얼의 일부 혹은 전체를 무단복사, 복제, 전제하는 것은 대한민국 저작권법에 저촉됩니다.

1.1.4 상표안내

Windows 는 Microsoft Corp. 의 등록상표입니다.

Microsoft®

Visual C++ 는 Microsoft Corp. 의 등록상표입니다.

Visual Basic 은 Microsoft Corp. 의 등록상표입니다.

Borland®

C++ Builder 는 Borland Software Corp. 의 등록상표입니다.

Delphi 는 Borland Software Corp. 의 등록상표입니다.

이외의 상표는 각 회사의 등록상표입니다.

1.1.5 주의사항

(주)커미조아의 제품군에는 제품의 특성(特性)에 따라서 하드웨어 및 소프트웨어 기술 지원이 필요한 경우가 있습니다. 필요하신 경우 본사 혹은 총판 및 대리점을 통해 제품 구입 이전에 점검 또는 요청해주시기 바랍니다.

(주)커미조아의 소프트웨어 및 하드웨어 제품군은 제품 성능 향상을 위해 예고 없이 변경될 수 있습니다. 고객(顧客)님께서서는 본 매뉴얼을 읽기전에 하드웨어 및 소프트웨어의 최신 변경사항에 대한 정보를 (주)커미조아를 통해 요청하실 수 있습니다.

본 매뉴얼은 (주)커미조아 ComiSWECAT 에 대한 정보를 포함한, 실제 라이브러리를 통한 모션 사항에 대한 기반 설명을 포함하고 있습니다.

최신 내용 및 기술되지 않은 사항 혹은 누락된 사항은 (주)커미조아 제품군 구입시에 고객(顧客) 등록을 해주신 고객(顧客)님의 정보를 통해 안내해 드릴 예정이며, 기술 지원 요청시에 보다 자세하고 정확한 방법과 내용을 통해 도움 드릴 것을 약속드립니다.

본 매뉴얼에 시작하기 앞서, 본 장(Chapter)에서는 본 매뉴얼을 보다 정확하게 빠르게 이해(理解)하실 수 있도록 ComiSWECHAT 의 전체 구성과 형식에 대해서 안내해 드리겠습니다.

1.1.6 매뉴얼 용어 안내

본 매뉴얼에서는 필요에 따라 매뉴얼에 기술된 내용을 설명하기 위해 함축된 의미의 용어나 현장 용어를 사용할 수도 있습니다. 최대한 보충 설명이 필요한 내용에 대해서는 해당 단원(Chapter)에서 설명하도록 하겠습니다.

매뉴얼 전체 범위에서 사용되는 범용적인 용어의 의미는 다음 표 1 매뉴얼 용어 정리 같이 미리 안내하여 드립니다.

명칭	의미
모터 (Motor)	서보모터를 비롯한 스텝 모터
서보팩 (Servo Pack)	서보 앰프
서보 드라이브 (Servo Drive)	서보모터와 서보앰프의 편성
스텝 드라이브 (Step Drive)	스텝모터와 스텝 모터 제어회로의 편성
서보 시스템 (Servo System)	서보 드라이브와 (주)커미조아 상위 제어기를 조합한, 일련의 완성된 시스템
지령 위치 출력 신호 (Command Pulse)	(주)커미조아 모션 제어 제품에서 서보 팩 혹은 스텝 드라이브 측으로 출력하는 전기적 펄스(Pulse) 열 신호
실제 위치 입력 신호 (Feedback Pulse)	(주)커미조아 모션 제어 제품에서 ETHERCAT 네트워크를 통해 위치 검출기(Encoder) 에서 입력받는 전기적 펄스(Pulse) 열 신호
ETHERCAT NC (ETHERCAT Network Controller)	ETHERCAT 네트워크 제어를 위한 본 제품을 일컫는 명칭

표 1 매뉴얼 용어 정리

1.1.7 매뉴얼 아이콘의 설명

매뉴얼에서 안내되는 내용을 보다 신속하고 정확하게 알 수 있도록 하며, 그 의미가 바르게 전달 되고 이해(理解)를 돕기 위해 원하는 의미에서 아래와 같은 매뉴얼 아이콘을 사용하고 있습니다.

 <p>보충</p>	이해(理解)하기 어려운 용어나 보충(補充)이 필요한 용어, 해설 및 사전에 설명 없이 새로 나온 용어를 설명합니다.
 <p>안내</p>	고객님의 편의와 기능의 자세한 내용에 대해 부가적으로 안내되는 사항입니다.

<p>주의</p> 	<p>이 동작이나 실행 함수(Function) 에 있어서 그 동작의 주의를 요망하는 내용을 나타냅니다.</p>
<p>경고</p> 	<p>이 동작이나 실행 함수(Function) 에 있어서 그 동작의 이상이나 문제가 발생할 경우 이 사항에 대해서 경고의 의미를 나타냅니다.</p>

표 2 매뉴얼 아이콘의 설명

1.2 Features

(주)커미조아의 ETHERCAT 라이브러리인 ComiSWECAT 의 장점은 다음과 같습니다.

1.2.1 독립성

ComiSWECAT 은 Microsoft 社의 표준라이브러리인 DLL(Dynamic Link Library) 형태의 독립된 라이브러리 인터페이스를 제공하며, 라이브러리 자체의 독립된 장치 관리의 편의성을 제공하고 있으며, DLL 형태의 라이브러리 장점을 통해 유지 보수와 귀사의 제품 구현에 보다 간편하게 하고 신뢰성 있는 독립형 동적 연결 라이브러리를 제공합니다

1.2.2 호환성

우수한 소프트웨어 개발 도구를 이용하여 전통적인 개발 방법보다 더 적은 시간과 비용으로 더 좋은 품질의 소프트웨어를 개발하는 방법을 이야기하는 최신 RAD(Rapid Application Development) 를 지향하고 있으며, 이에 맞는 최신 소프트웨어 개발 환경을 지원하고 있습니다.

고객(顧客)님께서서는 언제나 ComiSWECAT 를 통하여 귀사의 제품에 보다 신속하고 정확한 시스템을 구현하실 수 있습니다.

1.2.3 편의성

인터페이스 함수 명명 규칙의 통일화와 의사 코드 주제어를 매뉴얼과 라이브러리 인자(Parameter) 에 부각시켜, 보다 빠른 시간내에 숙련된 라이브러리 사용자로 만들어드립니다. 특히, ComiSWECAT 의 모든 함수 명에는 의미적 명명 규칙을 내포하였으며, 이것은 분명 실무 개발 환경에서 많은 부분 이점으로 작용할 것입니다.

1.2.4 신뢰성

(주)커미조아의 라이브러리 제품군은 오랜 시간 산업현장에서의 현장 경험을 바탕으로 형성된 신뢰성있는 제품군입니다. ‘부드럽고, 정확하고, 빠르고, 쉬운’ 모션제어 기능을 통해 모션의 기본에 충실하였으며, 각 동작에 대한 입력과 출력력을 ComiSWECAT 인터페이스 전역에 걸쳐 사용하기 쉽도록 안내해 놓았습니다.

또한, 응용프로그램에 기반하지 않는 자체 디버그 기능을 바탕으로 응용프로그램에 의한 오류를 최단시간내에 해결할 수 있도록 합니다. 디버그 모드의 지원과 향상된 디버그 정보를 바탕으로 오류발생에 대한 원인을 신속히 분석하실 수 있도록 도움을 드리며, 저희 (주)커미조아는 고객(顧客)님께 항상 정직과 신뢰를 드릴 수 있도록 최선을 다할 것입니다.

1.2.5 풍부한 예제와 신속한 기술 지원

지금 이시간에도 (주)커미조아는 타사에 비해 보다 많은 개발 선상의 활용가능한 라이브러리 예제를 지원해드리려고 노력하고 있으며, 예제간 중복 코드와 라이브러리 구성 이외의 부분을 최소화한 예제로 빠른 시간내에 예제의 코드를 바로 적용시켜드릴 수 있도록 노력하고 있습니다.

저희 (주)커미조아는 최신 .NET Framework 상의 C # (Sharp) 부터, Visual Basic 및 현존하는 RAD(Rapid Application Development) 환경을 지원하는 ComiSWECAT 으로 보다 향상된 고객(顧客) 지원과 기술 지원을 통해 고객(顧客)여러분들의 성원에 이바지하겠습니다.

Before working with ComiSWECAT

정밀한 모션제어와 고속의 모션제어는 최상의 모션 제어가 반드시 갖춰야할 필수요건입니다. 안정적인 하드웨어와 더불어 고급 기능의 소프트웨어는 고객(顧客)여러분들의 최상의 응용 프로그램 구현을 돕고 있습니다. 이제 더 이상 고민하지 마십시오. ComiSWECAT 은 보다 견고하고 우수한 기능을 바탕으로 여러분들이 원하시는 기능을 보다 빠르고 정확하게 안내하여드릴 것입니다.

본 장에서는 EtherCAT 에 대한 정의와 저희 (주)커미조아의 ComiSWECAT 에 대한 안내를 위한 내용으로 구성되어 있습니다. 보다 신속한 기술 지원과 빠른 라이브러리 기능 습득(習得)을 위해 구성(構成)되었으며, 사전 안내사항부터 함수 이름 규칙 가이드까지 모든 내용을 쉽고 빠르게 이해(理解)하실 수 있도록 최선을 다하였습니다.



2 EtherCAT & ComiSWECAT

본 장에서는 EtherCAT 에 대한 사항과 네트워크 제어를 자세히 안내해 드리고 있습니다.

2.1 EtherCAT 소개

EtherCAT(Ethernet for Control Automation Technology)은 유연한 토폴로지 구성이 가능한 고성능, 저비용의, 사용하기 쉬운 개방형 산업용 이더넷 기술입니다.

2.2 ComiSWECAT 개요

(주) 커미조아의 ComiSWECAT 은 전 세계적으로 가장 많은 사용자 층을 확보하고 있는 마이크로소프트 윈도우용 라이브러리를 제공하고 있습니다. 제공되는 라이브러리인 ComiSWecatSdk 는 EtherCAT 에 대한 지식이 없어도 사용이 가능하도록 사용자 중심의 High-Level API 로 구성되어 있습니다.

ComiSWECAT 은 소프트웨어는 윈도우 표준 동적 라이브러리(Dynamic Link Library) 형태로 고객 여러분들께 제공되며, 마이크로소프트의 Visual C++ , Visual Basic .NET Framework 환경의 Visual C#, Visual J# 개발환경, 볼랜드 C++ Builder 전제품, Delphi 전제품 인텔 컴파일러, 기타 Visual Fortran 등의 다양한 개발환경을 지원하고 있습니다.

2.3 함수 이름 규칙 가이드

구분 명	ComiSWECAT
장치 초기화 (Device Initialize)	secGn_LoadDevices secGn_UnLoadDevices
장치 인터페이스 (Net Interface)	secNet_SetAIState secNet_GetAIState
장치 설정 (ECAT Config)	secCfg_SetNetConfig secCfg_GetNetConfig
슬레이브 설정 (Slave I/F)	secSlv_SlvIdx2PhysAddr secSlv_PhysAddr2SlvIdx
모션 단축 (Motion Single Axis)	secmSxMot_MoveStart secmSxMot_MoveToStart
모션 보간 (Motion Interpolate)	secmIxMot_LineStart secmIxMot_LineToStart

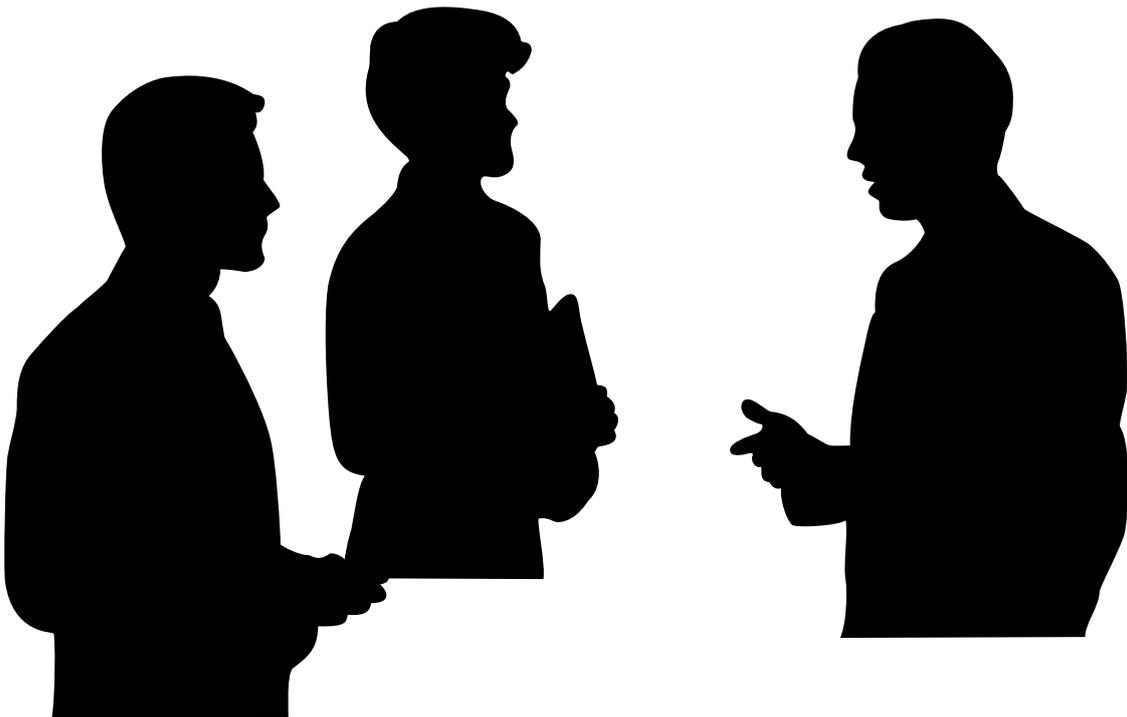
표 3 함수이름 규칙

- 본 지면 상 전체 함수에 대해서 다루지 못한 부분은 각 제품 매뉴얼을 참조해주시기 바랍니다

Development Environment for ComiSWECAT

쥬커미조아는 통합 라이브러리 ComiSWECAT 을 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 ComiSWECAT 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리에서 사용할 수 있도록 지원하여 드립니다.

쥬 커미조아 ComiSWECAT 은 다양한 고객(顧客) 여러분들의 요구에 맞추어 개발되었습니다., 가까운 대리점 혹은 총판점을 통해 구입(購入)하신 경우에는 해당 구입처(購入處)를 통해 문의하시면, 더욱 빠른 기술 지원을 받으실 수 있습니다.



3 개발 환경 별 ComiSWECA T 사용 안내

3.1 개발 환경 지원 안내

Supported Development Environment		개발 환경 별	언어별	버전 별	제품 별
		Microsoft 社 Visual Studio	Visual C++ Visual Basic C# (Sharp)	6.0 및 이하 버전을 포함한 VS2003, VS2005 환경	Enterprise Edition 을 포함한 전 제품
		Borland International 社 Borland Development Studio	C++ Builder	Builder 5 , Builder 6 및 하위 버전 포함한 2006 버전	Architect, Professional, Enterprise
			Visual C++	Delphi 5 , Delphi 6, Delphi 7 및 하위 버전 포함한 2006 버전	
		Borland International 社 Borland Turbo Series	Delphi	Turbo C++ Turbo Delphi / Turbo Delphi for .NET	전 제품
	Visual C++ .NET		Turbo C#	전 제품	
	Sybase PowerBuilder	C# (Sharp)	BDS 2006 버전	전 제품	
		C++ Builder	PowerBuilder 8, PowerBuilder 9, PowerBuilder 10.5	전 제품	
	Microsoft	Delphi			

표 4 지원 개발 환경

ComiSWECA T 은 위와 같은 개발 환경을 지원하며, 별도로 명시되지 않은 개발 환경에서도 윈도우의 Dynamic Link Library 형태를 사용가능한 경우 ComiSWECA T 을 사용하실 수 있습니다. 이 사항에 대해서 더욱 자세한 내용을 알기 원하실 경우 Appendix(부록) 편을 참고해 주시기 바랍니다.

3.2 ComiSWECAT 인터페이스 구성 파일

개발 환경	MS VC++	Borland C++ Builder	Borland Delphi	MS Visual Basic	MS C# (C Sharp)
항목					
ComiSWECAT 인터페이스 파일 명	ComiSWEcatSdk _Api.h	ComiSWEcatSdk _Api.h			
	ComiSWEcatSdk _Loader.cpp	ComiSWEcatSdk _Loader.cpp			
ComiSWECAT 상수(常數)정의 파일	ComiSWEcatSdk _Def.h	ComiSWEcatSdk _Def.h	ComiSWEcatSdk .PAS	ComiSWEcatSdk .BAS	ComiSWEcatSdk .CS
	ComiSWEcatSdk _Err.h	ComiSWEcatSdk _Err.h			
	ComiSWEcatSdk _SysDef.h	ComiSWEcatSdk _SysDef.h			

표 5 ComiSWEcatSdk 인터페이스 구성 파일 안내

Microsoft 社의 윈도우 운영체제에서 동작하는 DLL(Dynamic Link Library) 형태의 ComiSWECAT 은 상기 표에서 나타낸 것 처럼, 라이브러리 인터페이스를 위해 “ComiSWECAT 인터페이스” 파일과 라이브러리의 반환값 및 전달인자, 각 데이터 표기등을 위한 “ComiSWECAT 상수(常數) 정의 파일”을 제공하고 있습니다.

(주)커미조아 고객(顧客)님께서서는 “ComiSWECAT 인터페이스 파일” 을 통해 ComiSWECAT 의 함수를 명시적으로 응용프로그램에 포함시킬 수 있으며, 이에 따라 상기 표기한 개발 환경을 지원해드리고 있습니다. 만약 이외의 환경에서 저희 ComiSWECAT 을 사용하시기 원하신다면, 저희 (주)커미조아 고객(顧客) 지원 팀으로 연락 주시기 바랍니다.

3.3 각 개발 환경 별 안내

저희 (주)커미조아에서는 고객(顧客)님들의 개발환경에 대한 고민을 덜어드리기 위하여, 범용적인 객체지향 언어인 C++ 부터 Borland International 社의 Object Pascal 의 Delphi 제품군, 그리고 **최신 .NET 환경까지 고려한 라이브러리 인터페이스를 제공하고** 있습니다. 특히 이번 ComiSWECAT 에서는 개발 환경의 새 패러다임인 C Sharp(C#) 언어를 본격 지원하고 있으며, 보다 새로운 개발환경에서 저희 (주)커미조아 제품 군을 경험하실 수 있도록 만전을 기하고 있습니다.

안내

인터페이스 파일에 대해서 자세히 설명해 주십시오.

먼저, 저희 (주)커미조아에서 제공하는 ComiSWECAT 은 DLL 형태의 독립 라이브러리 인터페이스를 제공하는 마이크로소프트웨어의 윈도우 표준 라이브러리를 말합니다. 이것은 당사의 제품의 기능을 담당하는 가장 중요한 함수의 집합 구성입니다.

고객(顧客)님들께서는 이러한 함수, 즉 기능을 통해서 저희 (주)커미조아의 제품 군을 이용하게 됩니다.

그러나 마이크로소프트의 DLL 라이브러리 형태는 구조상 제공하는 함수들이 어떻게 써야 하는 지의 정보를 정확하게 알 수 없습니다. 다시 말씀 드려서, 함수의 매개변수의 개수나 매개변수의 형태 즉, 함수의 사용법을 알 수가 없습니다.

ComiSWECAT 에 대해서 그 사용방법을 각 개발환경에 맞게 제공해드리기 위해 [인터페이스] 파일을 제공하고 있습니다. 각 개발환경(VC++, Delphi 등)에서 개발을 하시는 고객(顧客) 여러분들께서는 반드시 이 [인터페이스] 파일을 사용하셔야만, ComiSWECAT 을 사용할 수 있습니다.

3.3.1 Visual C++ 6.x 개발자를 위한 안내

Microsoft Visual C++ 6.x 에서 ComiSWECAT 을 사용하시려면 다음의 절차에 따라 사용하시면 됩니다.

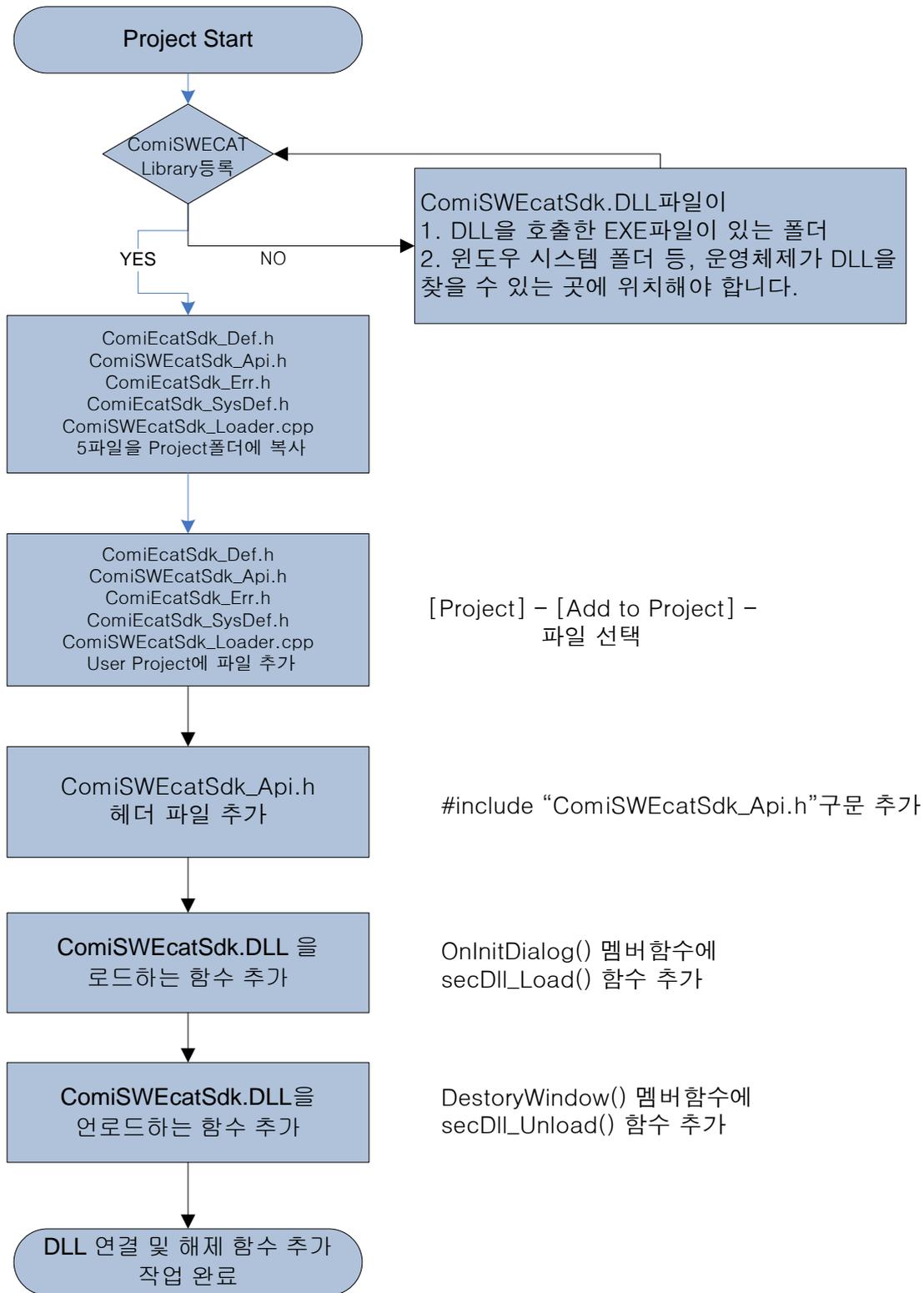


그림 3.3.1-1 Visual Studio 6.x 에서의 ComiSWECAT 사용 순서도

Visual C++ 6.x 을 실행합니다. 메뉴에서 'File'-'>'New' 를 선택하여 새로운 프로젝트 생성을 시작합니다.

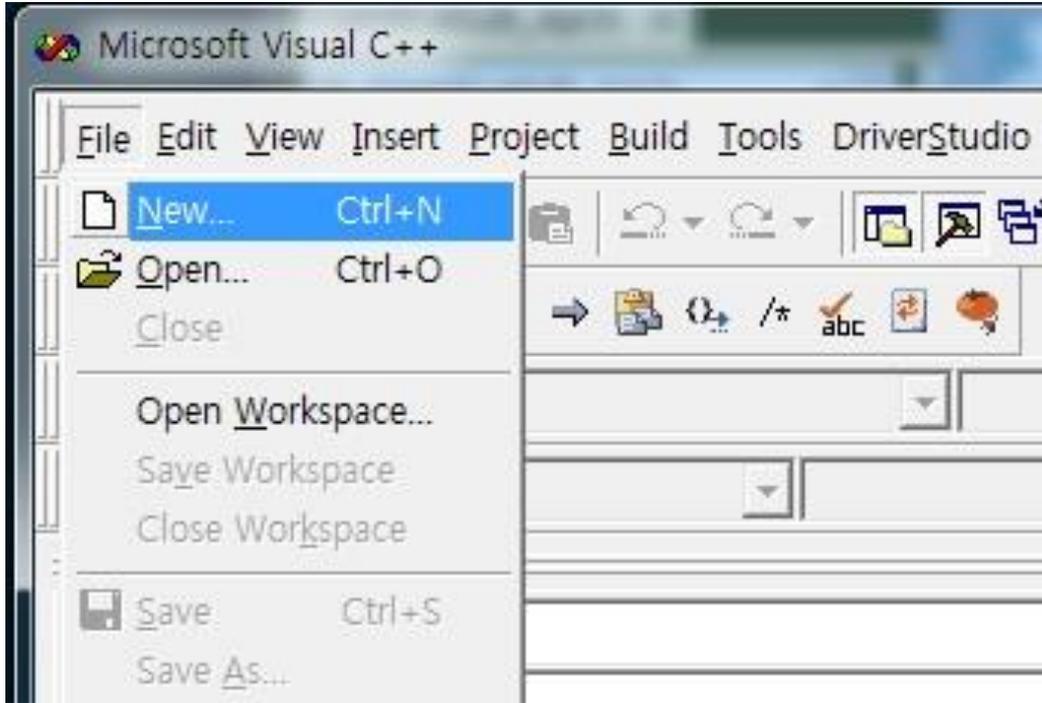


그림 3.3.1-2 Visual C++ 6.x 의 새로운 프로젝트 생성 화면

MFC AppWizard(exe)를 선택하고, 프로젝트를 생성할 위치와 프로젝트 이름을 입력한 후 [OK]버튼을 클릭 합니다.

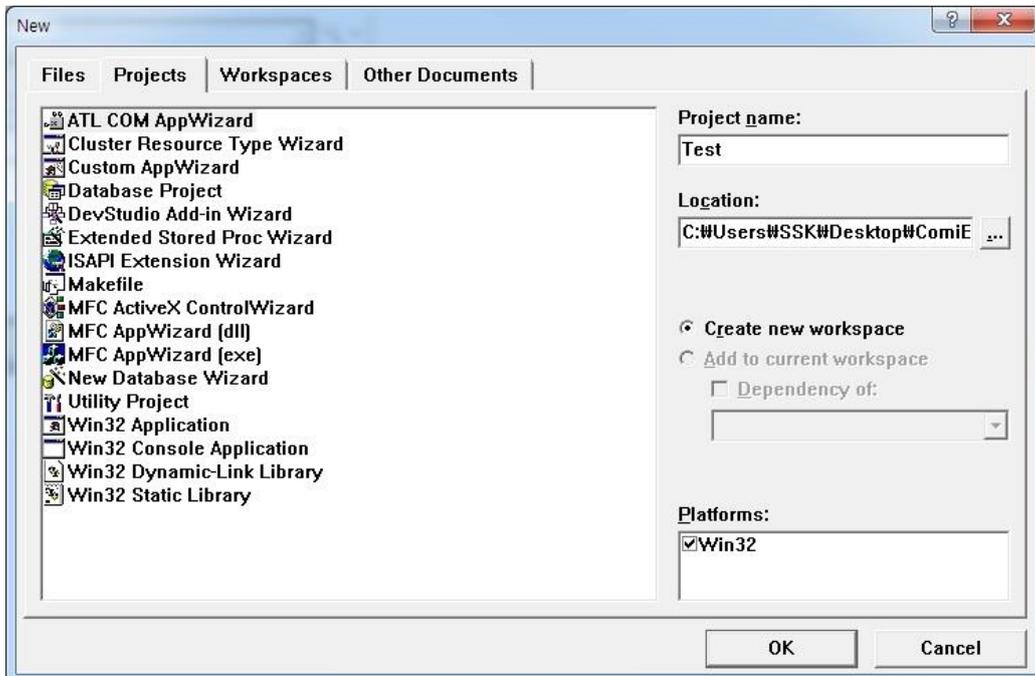


그림 3.3.1-3 새로운 프로젝트 생성 화면

MFC AppWizard 창이 나타나면 [Dialog based]를 선택하고 [Finish]버튼을 클릭합니다.

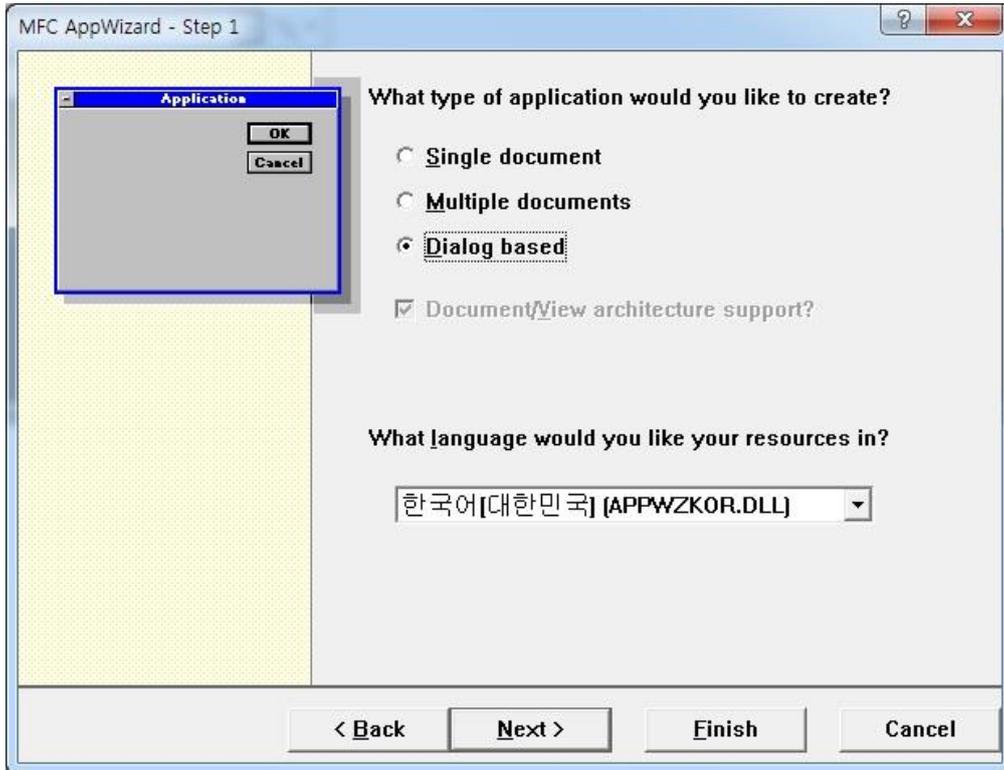


그림 3.3.1-4 MFC AppWizard 의 Application Type 선택 화면

VC++ 용 인터페이스 정의 파일인 ComiECATSdk_Def.h, ComiSWECATSdk_Api.h, ComiECATSdk_Err.h, ComiECATSdk_SysDef.h, ComiSWECATSdk_Loader.cpp 파일을 신규로 생성한 프로젝트 폴더로 복사합니다.

메뉴에서 [Project]->[Add To Project]->[Files]를 선택합니다.

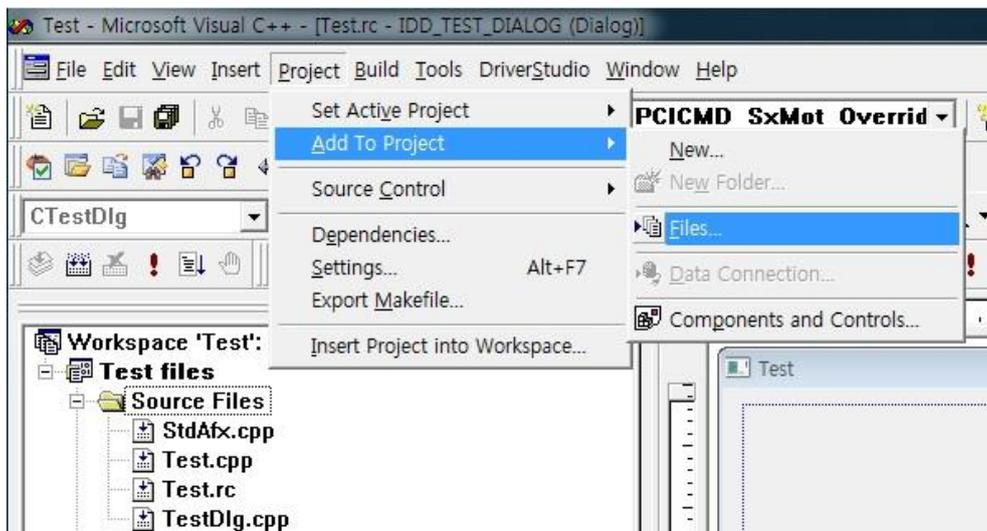


그림 3.3.1-5 프로젝트에 새로운 파일 추가 선택화면

추가될 파일을 선택한 후 [OK]버튼을 클릭하여 통합 모션 라이브러리 인터페이스 파일인 다섯 프로젝트에 추가 합니다.

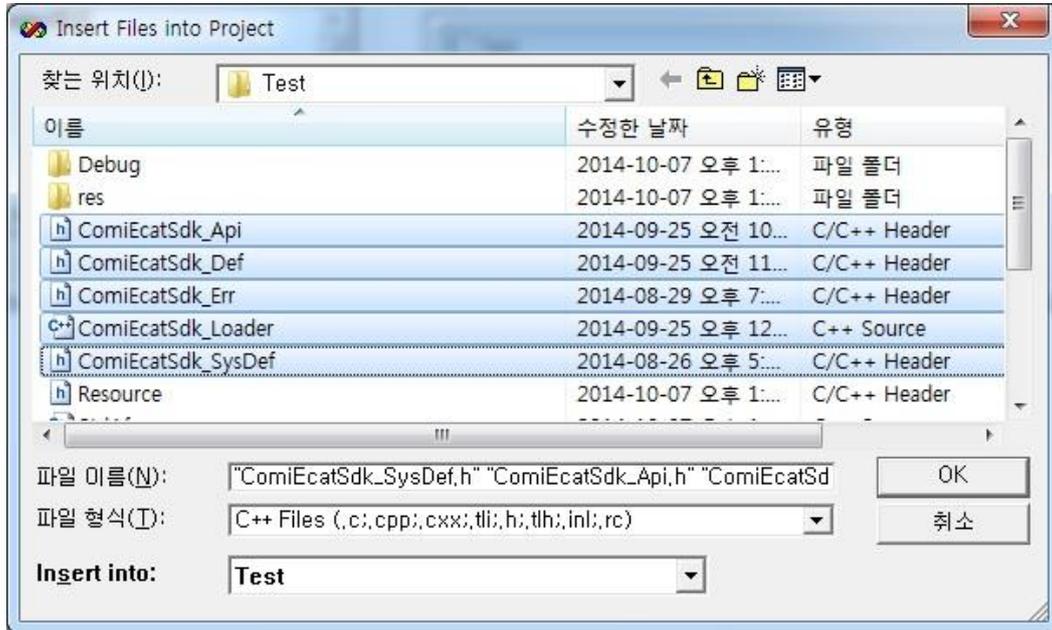


그림 3.3.1-6 프로젝트에 추가할 파일 선택 화면

Workspace 창의 FileView 탭에서 [생성한 프로젝트 이름]+Dlg.cpp 파일을 선택합니다.

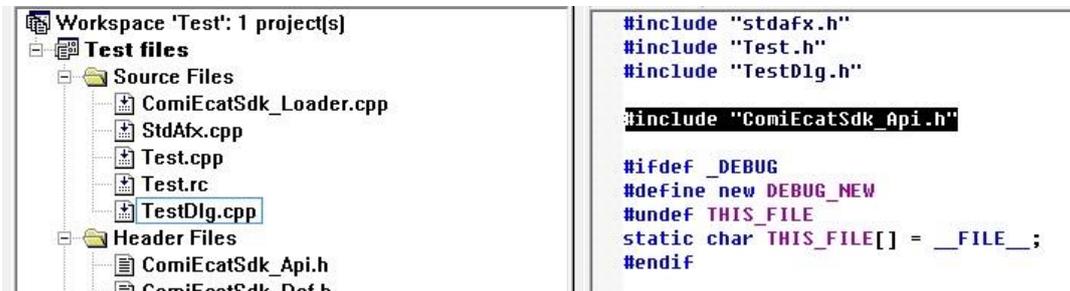


그림 3.3.1-7 사용자가 MFC AppWizard 를 통해서 생성한 소스코드에 ComiSSCNET3 파일을 추가 함

([생성한 프로젝트 이름]+Dlg.cpp) 파일의 OnInitDialog() 함수 내부의 “TODO”아래에 “secDll_Load();”을 추가 합니다.

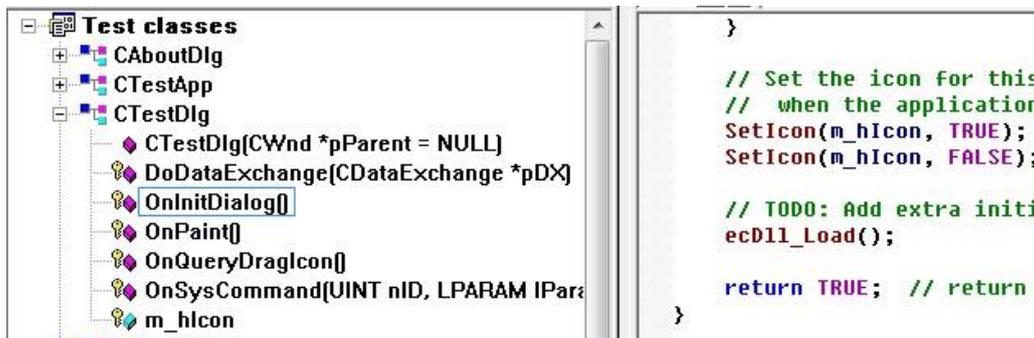


그림 3.3.1-8 DLL 로드 함수 호출 화면

사용자 작성 프로그램이 종료되면 DLL을 Unload시켜야 합니다. DLL의 Unload는 사용자 작성 프로그램의 종료시 이루어져야 하며 secDll_Unload()이라는 함수를 통해서 이루어 집니다. secDll_Unload()를 추가 하는 방법은 다음과 같습니다.

Class View 창에서 [(생성한 프로젝트 이름)+Dlg] 클래스를 마우스 오른쪽 버튼으로 클릭 합니다. 팝업 메뉴에서 [Add Virtual Function]을 선택합니다.



그림 3.3.1-9 가상 함수 추가

'New Virtual Functions'항목에서 'DestroyWindow'를 선택한 다음 [Add and Edit]버튼을 클릭합니다.

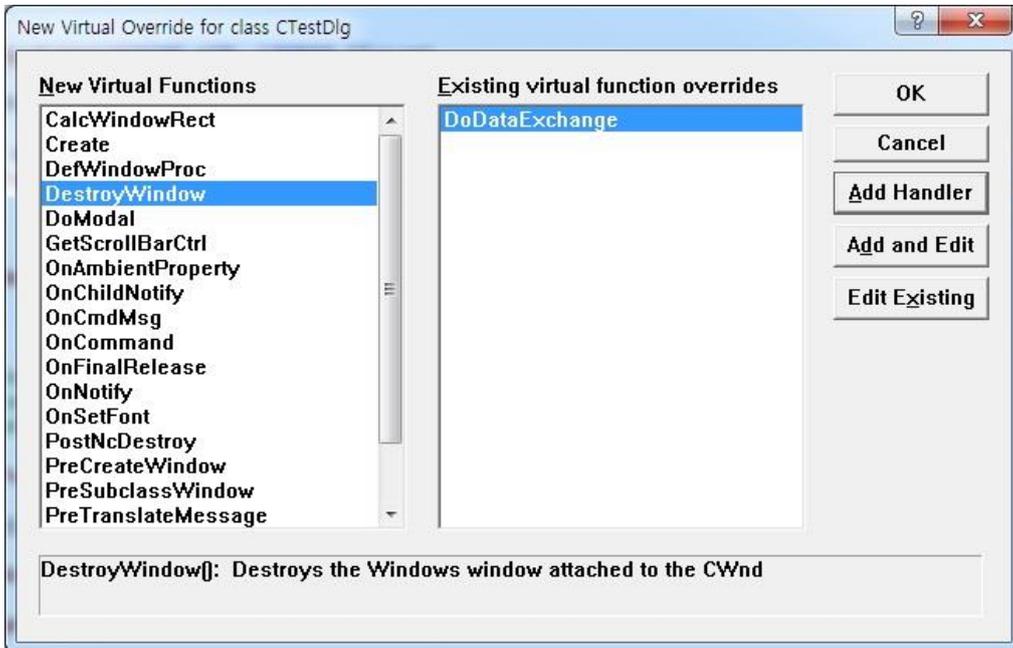


그림 3.3.1-10 DestroyWindow 함수 추가

(생성한 프로젝트 이름)+Dlg 클래스의 멤버함수인 'DestroyWindow()'에 'secDll_Unload();'을 추가 합니다.
 'secDll_Unload()'함수를 추가 하면 윈도우가 종료 될 때 자동으로 DLL도 해제 됩니다.

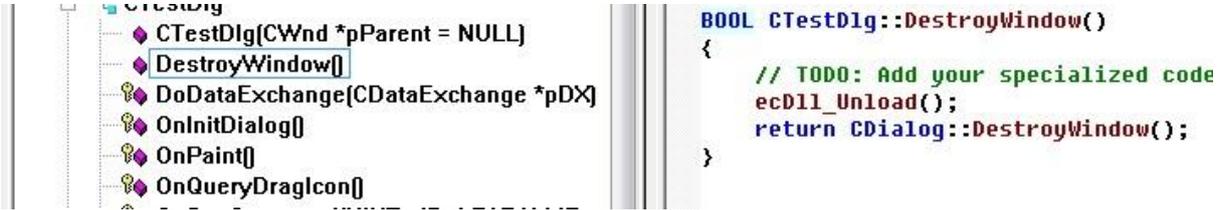


그림 3.3.1-11 DLL 로드 및 언로드 코드 추가

3.3.2 Visual C++ 10.x 개발자를 위한 안내

Microsoft 社の Visual C++ 10.x (Visual Studio 2010)에서 ComiSWECAT 을 사용하시려면 다음의 절차에 따라 사용하시면 됩니다.

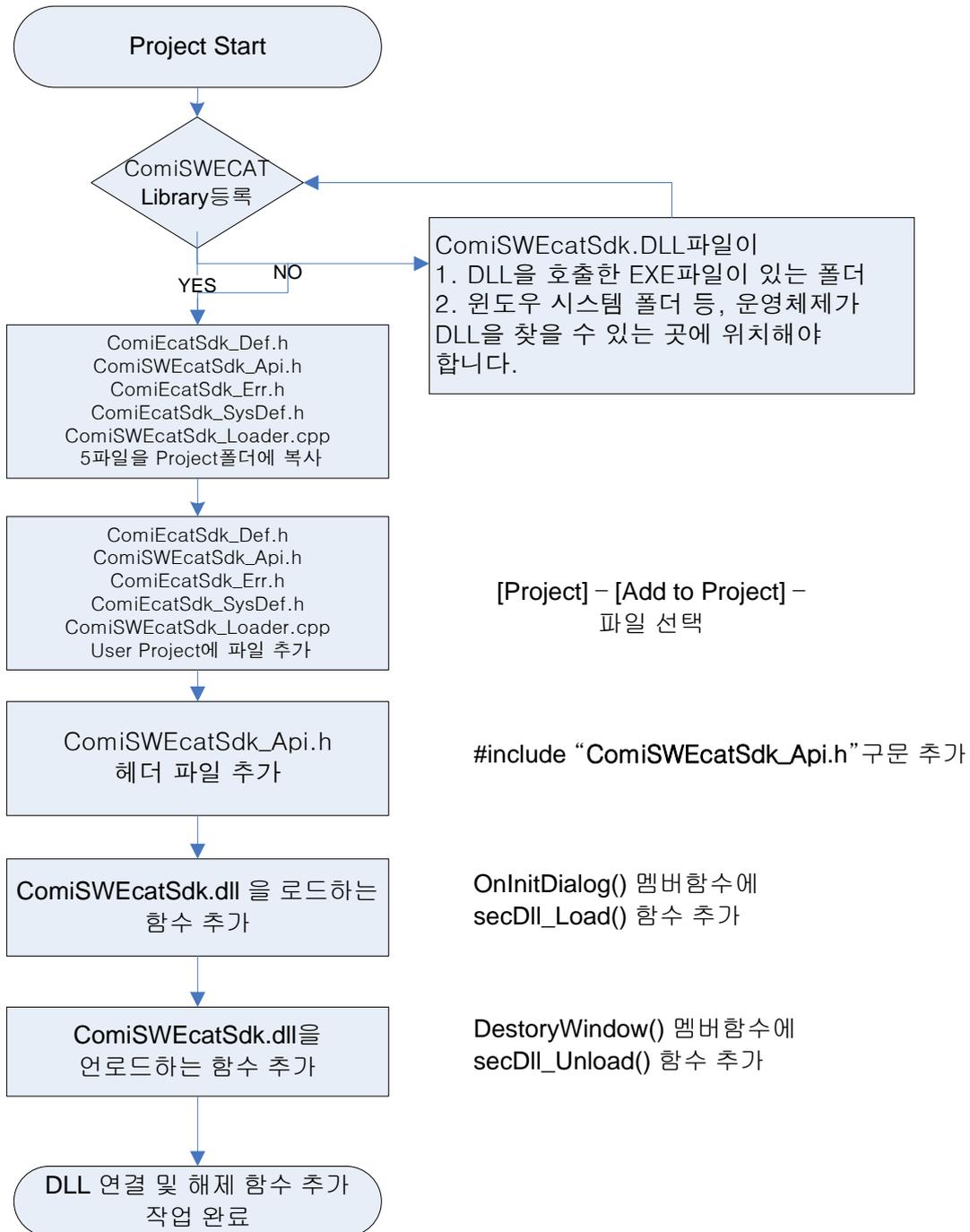


그림 3.3.2-1 Visual Studio 10.x 에서의 ComiSWECAT 사용 순서도

Microsoft 社의 Visual Studio 2010(이하 VS2010)를 실행 합니다.

메뉴에서 [파일]->[새로 만들기]->[프로젝트]를 선택하여 새로운 프로젝트를 시작 합니다.

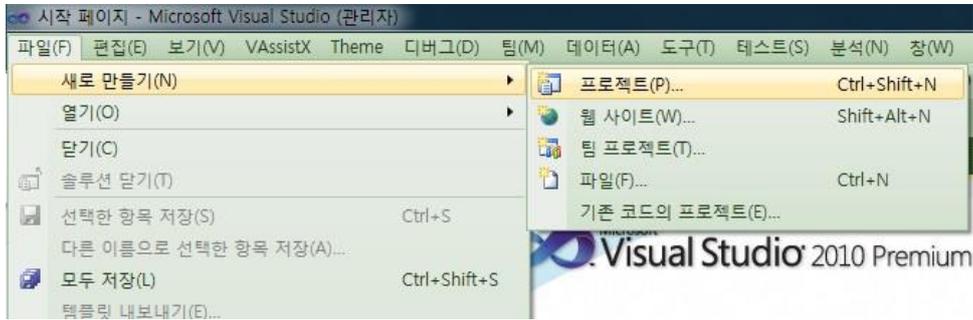


그림 3.3.2-2 새로운 프로젝트 생성 시작

[새 프로젝트]창이 화면에 나타나면, [설치된 템플릿]에서는 [Visual C++]을 선택하고, [Templates]에서 [MFC 응용 프로그램]을 선택합니다. 그리고 프로젝트를 생성할 위치와 프로젝트 이름을 입력한 후 [OK]버튼을 클릭 합니다.

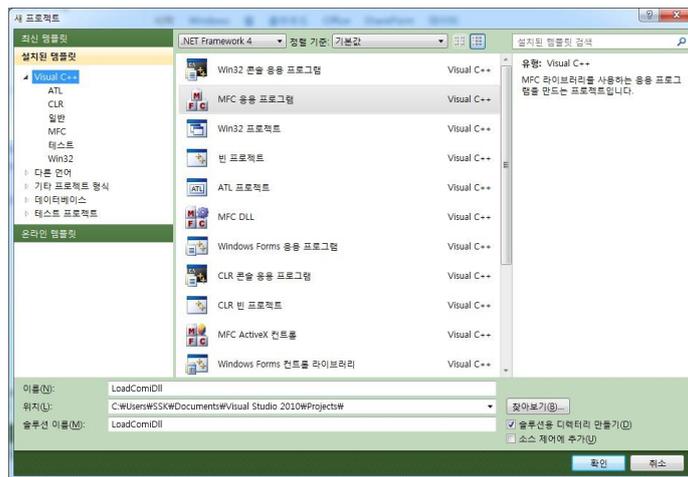


그림 3.3.2-3 새 프로젝트 옵션 선택화면

[MFC 응용 프로그램 마법사] 창이 화면에 나타나면, [다음]을 클릭합니다.

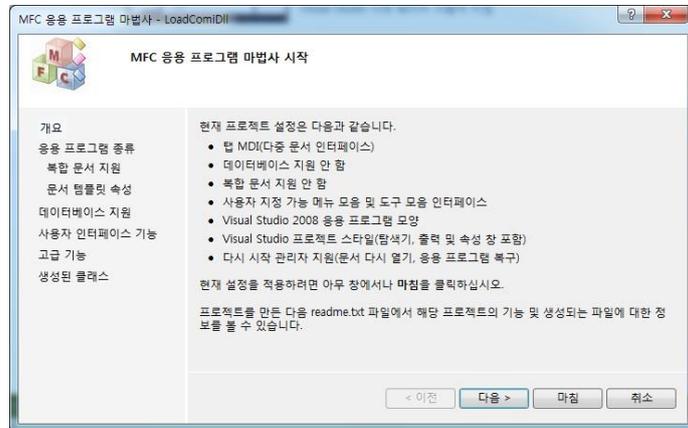


그림 3.3.2-4 MFC Application Wizard 의 Overview 화면

[응용 프로그램 종류]에서 [대화 상자 기반]을 선택하고, [유니코드 라이브러리 사용]를 해제(Uncheck) 한 다음 [마침]을 클릭합니다.



그림 3.3.2-5 MFC Application Wizard 의 Application Type 화면

VC++ 용 인터페이스 정의 파일인 ComiSWECATSdk_Api.h, ComiECATSdk_Def.h, ComiECATSdk_Err.h, ComiECATSdk_SysDef.h, ComiSWECATSdk_Loader.cpp 파일을 신규로 생성한 프로젝트 폴더로 복사합니다.

메뉴에서 [프로젝트]->[기존 항목 추가]을 선택합니다.

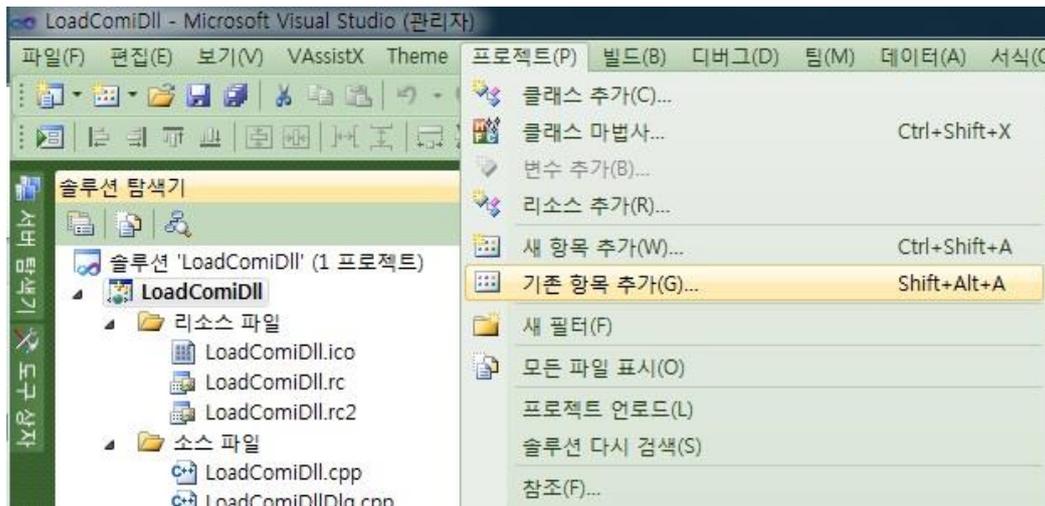


그림 3.3.2-6 메뉴에서 기존 항목 추가 선택 화면

추가될 파일을 선택한 후 [OK]버튼을 클릭하여 다섯 개의 파일을 프로젝트에 추가 합니다.

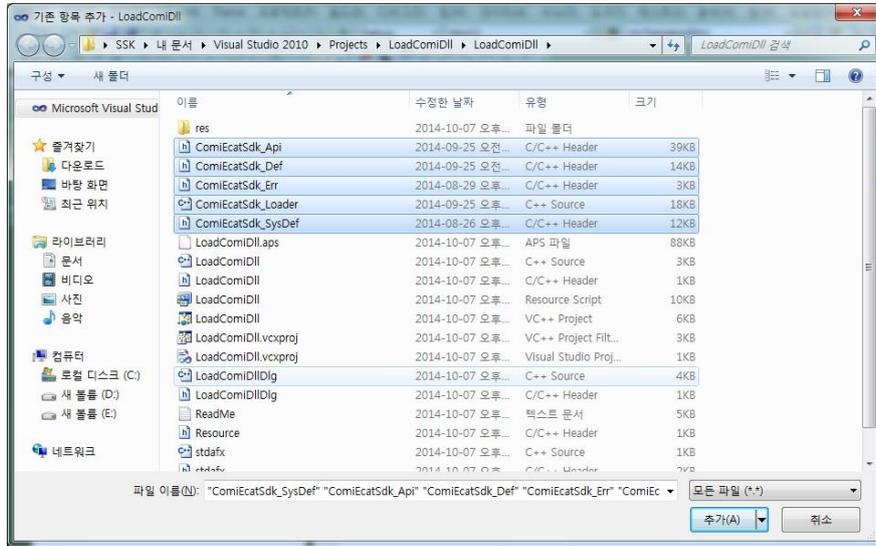


그림 3.3.2-7 새로 추가할 파일들 선택 화면

Workspace 창의 FileView 탭에서 ([생성한 프로젝트 이름]+Dlg.cpp) 파일을 선택합니다. 파일의 가장 위쪽에 인터페이스 파일을 추가합니다.

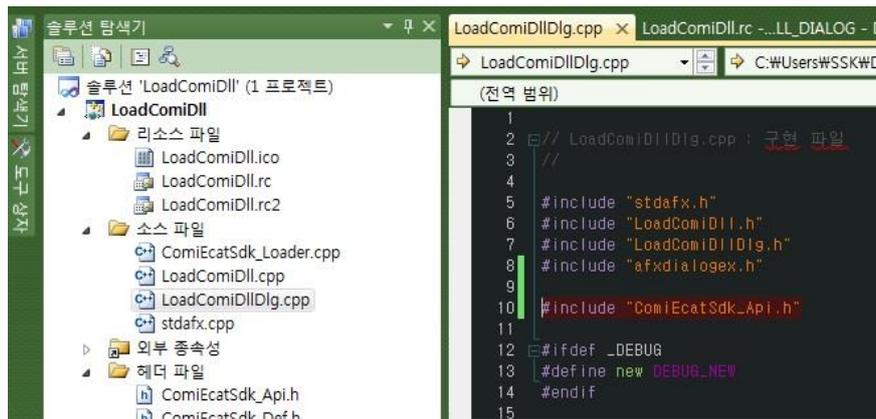


그림 3.3.2-8 사용자 생성 CPP 파일에 헤더 추가 화면

(생성한 프로젝트 이름)+Dlg.cpp 파일의 OnInitDialog() 함수 내부의 “TODO”아래에 “secDll_Load();”를 추가 합니다.

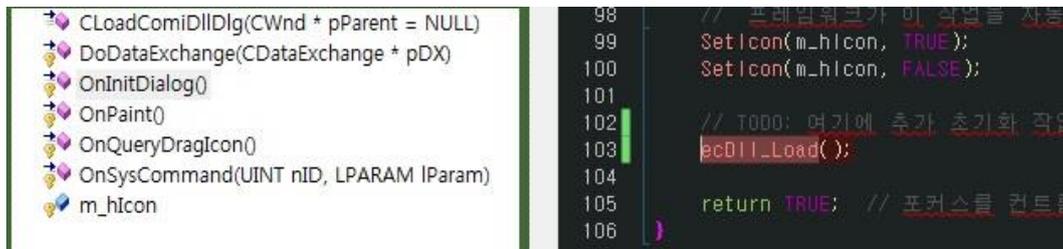


그림 3.3.2-9 LoadDll 추가

사용자 작성 프로그램이 종료되면 DLL 을 Unload 시켜야 합니다. DLL 의 Unload 는 사용자 작성 프로그램의 종료시 이루어져야 하며 secDll_Unload()이라는 함수를 통해서 이루어 집니다. secDll_Unload()을 추가 하는 방법은 다음과 같습니다.

Class View 창에서 ([생성한 프로젝트 이름]+Dlg) 클래스를 선택합니다.

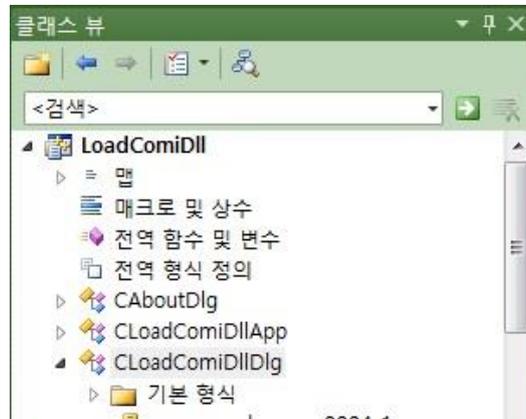


그림 3.3.2-10 사용자 생성 Dialog Class 선택

(생성한 프로젝트 이름)+Dlg 클래스가 선택된 상태에서 Properties 창의 ‘Overrides’를 선택합니다.



그림 3.3.2-11 Overrides 선택

‘DestroyWindow’항목 옆의 콤보 박스를 클릭하여 ‘<Add>DestroyWindow’를 선택합니다. (생성한 프로젝트 이름)+Dlg 클래스에 ‘DestroyWindow’라는 이름의 오버라이드된 함수가 추가 됩니다.

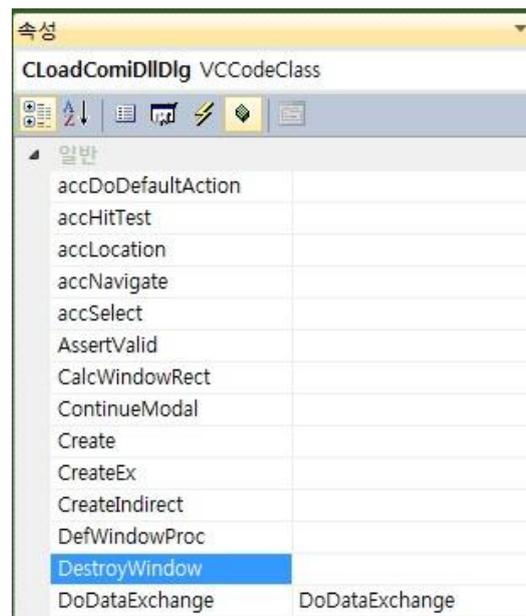


그림 3.3.2-12 Destroy Window 함수 추가

([생성한 프로젝트 이름]+Dlg) 클래스의 멤버함수인 DestroyWindow()에 'secDll_Unload();'을 추가 합니다.
'secDll_Unload()'함수를 추가 하면 윈도우가 종료 될 때 자동으로 DLL도 해제 됩니다.



그림 3.3.2-13 UnloadDll 함수 추가

3.3.3 Borland C++ Builder 개발자를 위한 안내

Borland C++ Builder 는 해당 개발 환경 버전인 BCB 5, BCB 6 및 BDS 2006 버전에서 ComiSWECAT 의 인터페이스 연결 방법이 매우 유사하기 때문에 공통적인 부분으로서 안내를 해드립니다.

전체 버전(Version)의 Borland C++ Builder 에서 ComiSWECAT 을 사용하시려면 다음의 절차를 통해 안내 받으시기 바랍니다.

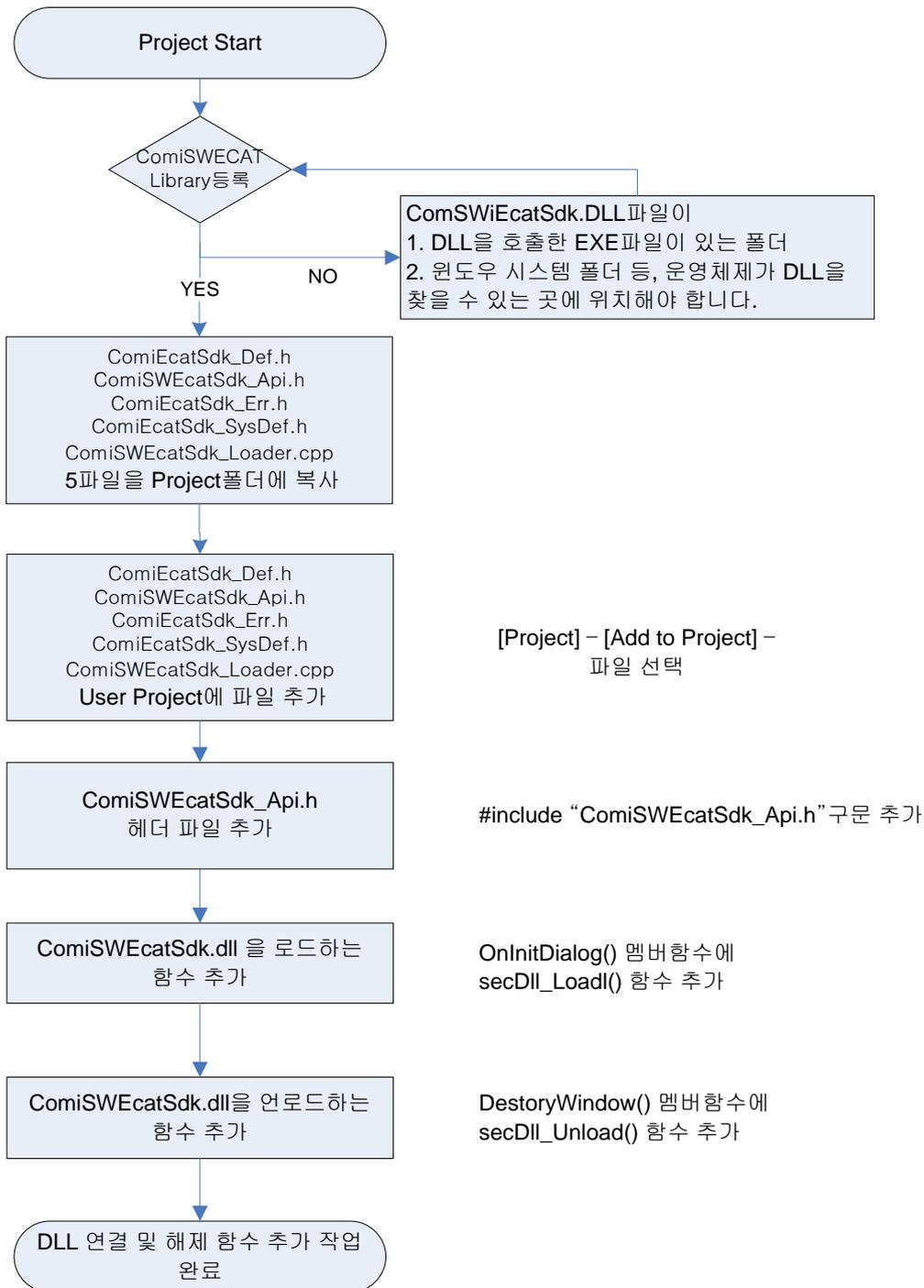


그림 3.3.3-1 Borland C++ Builder 에서 ComiSWECAT 사용 순서도

본 개발자를 위한 실제 안내에서는 다양한 버전의 Borland C++ Builder 의 화면을 통해 안내 헤드리도록 하였습니다.

Borland C++ Builder 를 실행합니다. 메뉴에서 [File]->[New]->[Application]을 선택하여 새로운 프로젝트를 시작합니다.



그림 3.3.3-2 BCB 5 에서 새로운 프로젝트 생성

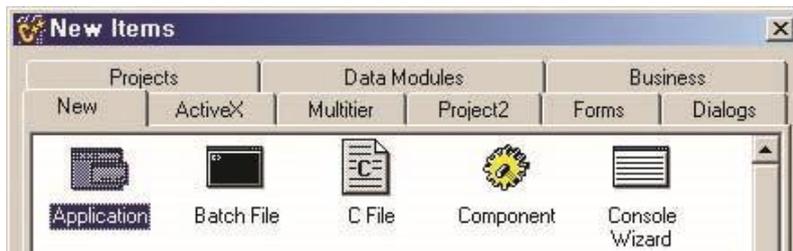


그림 3.3.3-3 BCB 5 에서 새로운 프로젝트 생성



그림 3.3.3-4 BCB 6 에서 새로운 프로젝트 생성

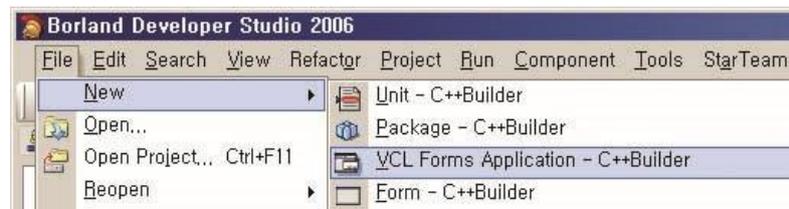


그림 3.3.3-5 BDS 2006 에서 새로운 프로젝트 생성

Borland C++ 및 VC++ 용 공용 인터페이스 정의 파일인 ComiSWECATSdk_Api.h, ComiECATSdk_Def.h, ComiECATSdk_Err.h, ComiECATSdk_SysDef.h, ComiSWECATSdk_Loader.cpp 파일을 신규로 생성한 프로젝트 폴더로 복사 합니다.

이름	수정한 날짜	유형
ComiEcatSdk_Api	2014-09-25 오전...	C/C++ Header
ComiEcatSdk_Def	2014-09-25 오전...	C/C++ Header
ComiEcatSdk_Err	2014-08-29 오후...	C/C++ Header
ComiEcatSdk_Loader	2014-09-25 오후...	C++ Source
ComiEcatSdk_SysDef	2014-08-26 오후...	C/C++ Header

그림 3.3.3-6 ComiSWECAT 사용시 공통으로 사용되는 파일

 <p>안내</p>	<p>참고 사항</p> <p>Borland 社의 C++ Builder 에서는 [File]-[Save] or [Save All]을 해주어야 프로젝트 폴더 및 프로젝트 관련 파일들을 생성합니다.</p>
---	---

그림과 같이 C++ Builder 에서 추가할 인터페이스 파일을 실제 사용자 프로젝트에 추가합니다. Project 의 메뉴의 Add to Project 를 사용하시면 됩니다.

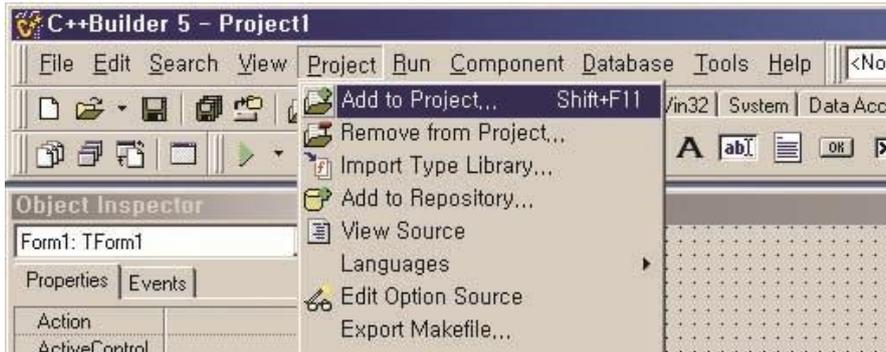


그림 3.3.3-7 C++ Builder 에서 프로젝트에 파일 추가 단계 1

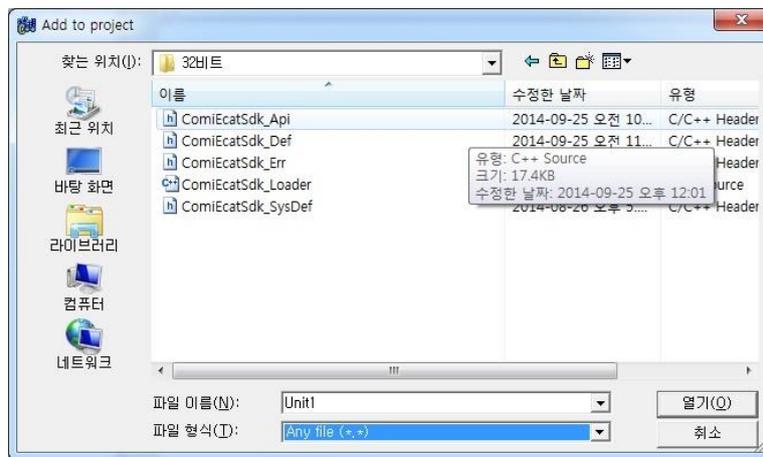


그림 3.3.3-8 C++ Builder 에서 프로젝트에 파일 추가

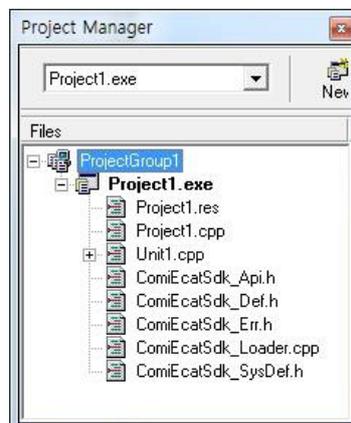


그림 3.3.3-9 Project Manager 에서 추가된 파일 확인(確認)

라이브러리 함수를 사용하고자 하는 대상 구현부 응용프로그램 파일에 인터페이스 파일을 선언합니다.

```
Unit1.cpp | Project1.cpp |
//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "ComiEcatsdk_Api.h"
//-----
```

그림 3.3.3-10 라이브러리 사용시 필요한 헤더 파일 선언

ComiSWECATSDk.dll 파일을 secDll_Load() 함수를 이용하여 로드할 수 있도록 FormCreate 함수 또는 응용프로그램 시작 부분에 추가합니다.

ComiSWECATSDk.dll 파일을 secDll_Load() 함수를 이용하여 로드합니다. secDll_Load()을 추가 하는 방법을 FormCreate 를 통해 할 수 있으며, 그 예를 소개해 드립니다.

[Object Inspector] - [Events] 탭의 OnCreate 에서 더블클릭합니다.

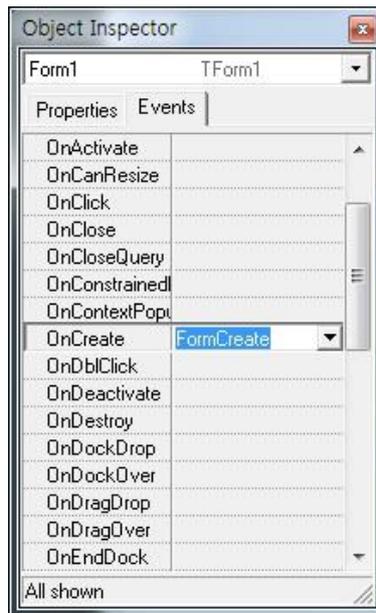


그림 3.3.3-11 OnCreate Event 추가하여 FormCreate 함수와 연결

추가된 FormCreate() 또는 응용프로그램 종료 함수 안에 secDll_Load() 함수를 추가합니다.

```
Unit1.cpp | Project1.cpp |

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    secDll_Load();
}
//-----
```

그림 3.3.3-12 secDll_Load 함수 추가

고객(顧客)님이 작성하신 프로그램이 종료되면 DLL 을 명시적으로 Unload 시켜야 합니다. DLL 의 Unload 시점은 고객(顧客)님께서 작성하신 응용프로그램이 종료되는 시점에 반드시 이루어져야 하며 secDll_Unload() 라는 함수를 통해서 이루어 집니다. secDll_Unload() 을 추가 하는 방법은 다음과 같습니다.

[Object Inspector] - [Events] 탭의 OnDestroy 에서 더블클릭합니다.

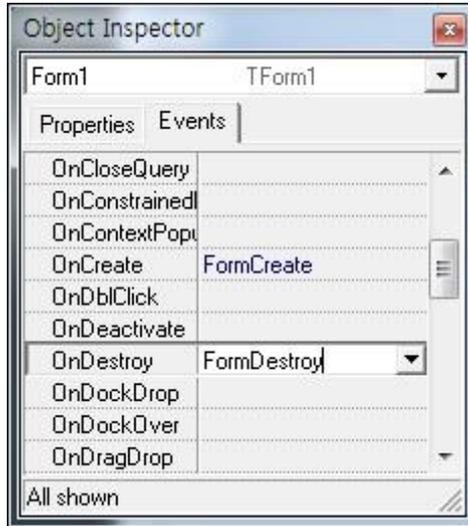


그림 3.3.3-13 응용프로그램의 종료시 DLL 이 명시적으로 UnLoad 될 수 있도록 OnDestroy Event 와 함수의 연결

추가된 FormDestroy() 또는 응용프로그램 종료 함수에 secDll_Unload() 함수를 추가합니다.

```

Unit1.cpp | Project1.cpp |
//-----
void __fastcall TForm1::FormDestroy(TObject *Sender)
{
    secDll_Unload();
}
//-----
    
```

그림 3.3.3-14 FormDestroy 함수를 통하여 명시적인 secDll_Unload 함수 구현

3.3.4 Borland Delphi 개발자를 위한 안내

Borland Delphi 는 해당 개발 환경 버전인 Delphi 5, Delphi 6 및 Delphi 7, BDS 2006 버전에서 ComiSWECAT 의 인터페이스 연결 방법이 매우 유사하기 때문에 공통적인 부분으로서 안내를 해드립니다.

전체 버전(Version)의 Delphi 에서 ComiSWECAT 을 사용하시려면 다음의 절차를 통해 안내 받으시기 바랍니다.

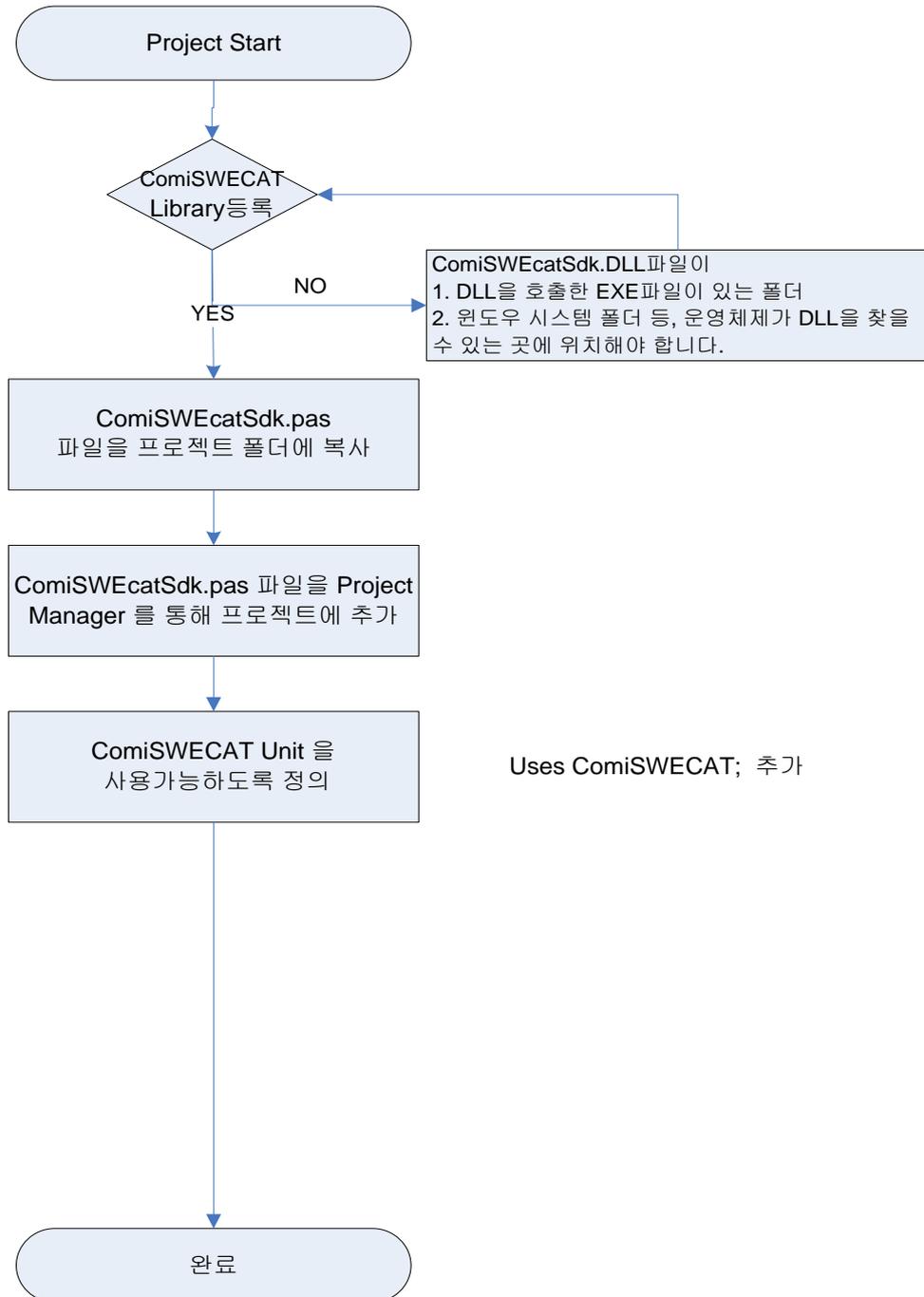


그림 3.3.4-1 Borland Delphi 에서 ComiSWECAT 사용 순서도

Delphi 7 을 기준으로 설명드리겠습니다. 만약 안내해드리는 도중에 다른 개발 환경과 구분이 되어야 할 내용은 별도로 설명 드리겠습니다. Borland Delphi 를 실행합니다.



그림 3.3.4-2 Borland Delphi 7 화면

프로젝트 시작 전에 ㈜커미조아 ComiSWECAT 의 Delphi 용 공용 인터페이스 파일을 프로젝트 폴더에 복사합니다. 이 파일은 ㈜커미조아 ComiSWECAT 의 DLL(Dynamic Link Library) 와 고객님의 응용 프로그램과의 인터페이스를 정의하여 놓은 파일입니다.

델파이(Delphi)는 명시적으로 프로젝트파일 간의 상호 변환이 필요 없습니다. 따라서 Delphi 5, Delphi 6, Delphi 7 각각의 버전에서 몇 가지 기본적인 컴포넌트에 기반한 내용을 제외하고는 그대로 사용할 수 있습니다. ㈜커미조아 ComiSWECAT 에서는 Delphi 에 대한 풍부한 예제를 제공하고 있습니다.

새로운 프로젝트를 시작하기 위해서, 메뉴의 [File]-[New] 명령을 통해, 새로운 응용 프로그램 개발을 시작합니다.

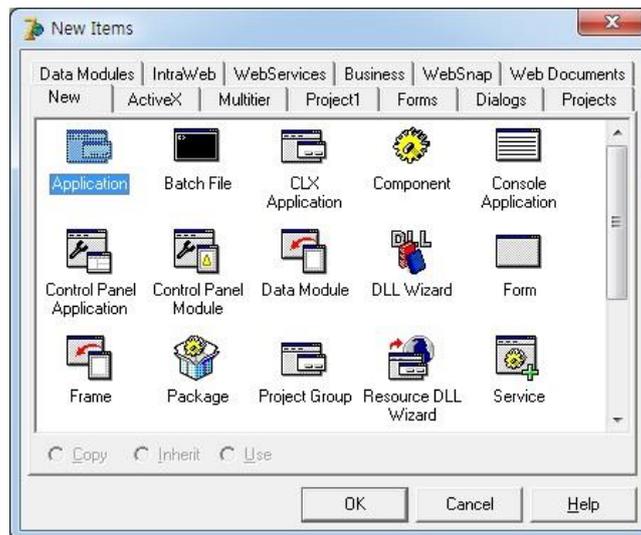


그림 3.3.4-3 Delphi 7 의 프로젝트 시작

프로젝트가 시작되면 화면상에 'Form1' 혹은 Delphi IDE 의 Project1 이 나타납니다.
 인터페이스 파일을 추가하기 위한 작업으로서 [Project] 메뉴의 [Add to Project] 를 선택합니다.

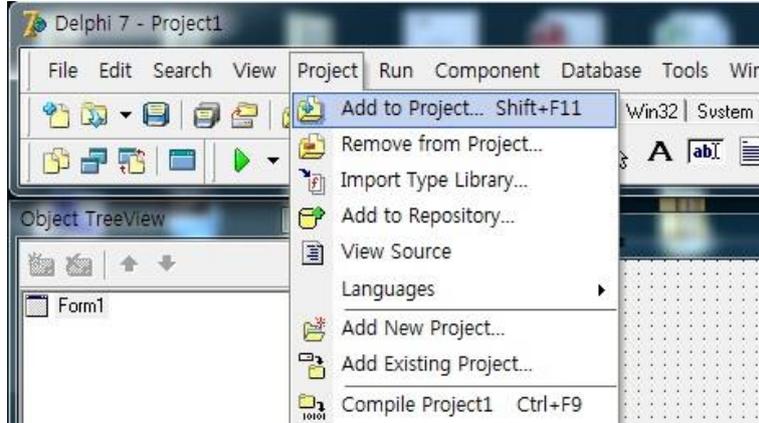


그림 3.3.4.4 Delphi 용 인터페이스 파일 추가

(주) 커미조아 ComiSWECAT 의 공용 인터페이스 정의 파일인 ComiSWECATsDk.PAS 파일을 프로젝트에 추가합니다.

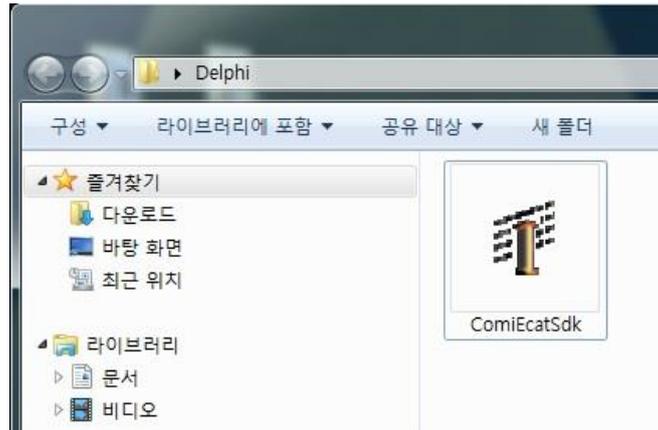


그림 3.3.4.5 Delphi 용 인터페이스 파일

Project Manager 를 통해 확인해 보면 ComiSWECATsDk.PAS 파일이 프로젝트에 등록된 것을 확인할 수 있습니다.

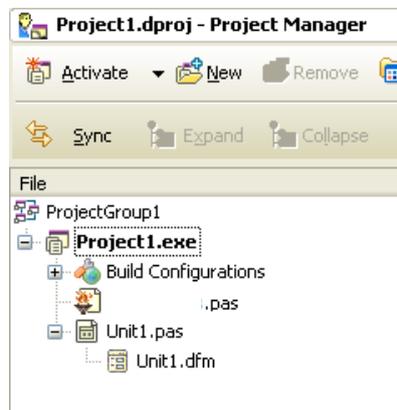


그림 3.3.4.6 Delphi 용 인터페이스 파일을 프로젝트 매니저에 추가

	<p style="text-align: center;">Delphi 의 ComiSWECAT 인터페이스 파일의 사용에 대한 부가 안내</p> <p>저희 (주)커미조아의 ComiSWECAT 의 ComiSWECATSdk.pas 파일은 다른 개발환경(VC++, C++ Builder) 와 달리 DLL 의 묵시적인 로드와 언로드가 자동으로 이루어집니다. 이것은 델파이가 가지고 있는 Initialization 과 Finalization 을 이용한 것으로 프로젝트에 별다른 LoadDll 함수 호출 없이 자동으로 DLL 이 로드 됩니다.</p> <p>또한 응용프로그램 종료시에 UnloadDll 이 명시적으로 호출되지 않아도, 자동적으로 응용 프로그램이 종료시에 UnloadDll 이 실행됩니다.</p>
---	---

실제 구현 부의 코드를 에디터를 통해 확인합니다.

```

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
  //////////////////////////////////////
  //// COMIZOA SDK LIBRARY 를 위한 인터페이스 파일을 사용합니다.
  uses ComiEcatSdk;
  //////////////////////////////////////

  {$R *.dfm}

end.

```

그림 3.3.4.7 uses 구문을 통해 ComiSWECATSdk Unit 사용

위와 같이 implementation 부에 uses 를 통해 라이브러리 Unit 을 사용할 수 있도록 반드시 지정해 주십시오. (상단의 uses 에 선언하여도 무방합니다) 이후, 델파이 에서는 다른 개발과 동일하게 DLL 라이브러리를 사용하실 수 있습니다.

3.3.5 Visual Basic 개발자를 위한 안내

Visual Basic 6.0 은 마이크로소프트의 컴포넌트 기반 응용프로그램 개발을 위해 태어난 뛰어난 개발 환경입니다. ComiSWECAT 는 Visual Basic 6.0 를 완벽히 지원하며, 인터페이스 파일을 제공하고 있습니다. Visual Basic 고객(顧客)님들께서도 응용프로그램 개발에 편의성을 드리기위해 저희 (주)커미조아는 언제나 노력하고 있습니다.

실제 Visual Basic 6.0 의 프로젝트 시작 전에, 프로젝트 디렉토리에 (주)커미조아 ComiSWECAT 인터페이스 파일인 ComiSWECATSdk.BAS 파일과 ComiSWECAT 파일 ‘ComiSWECATSdk.dll’ 파일을 반드시 프로젝트 디렉토리에 복사해주시기 바랍니다.

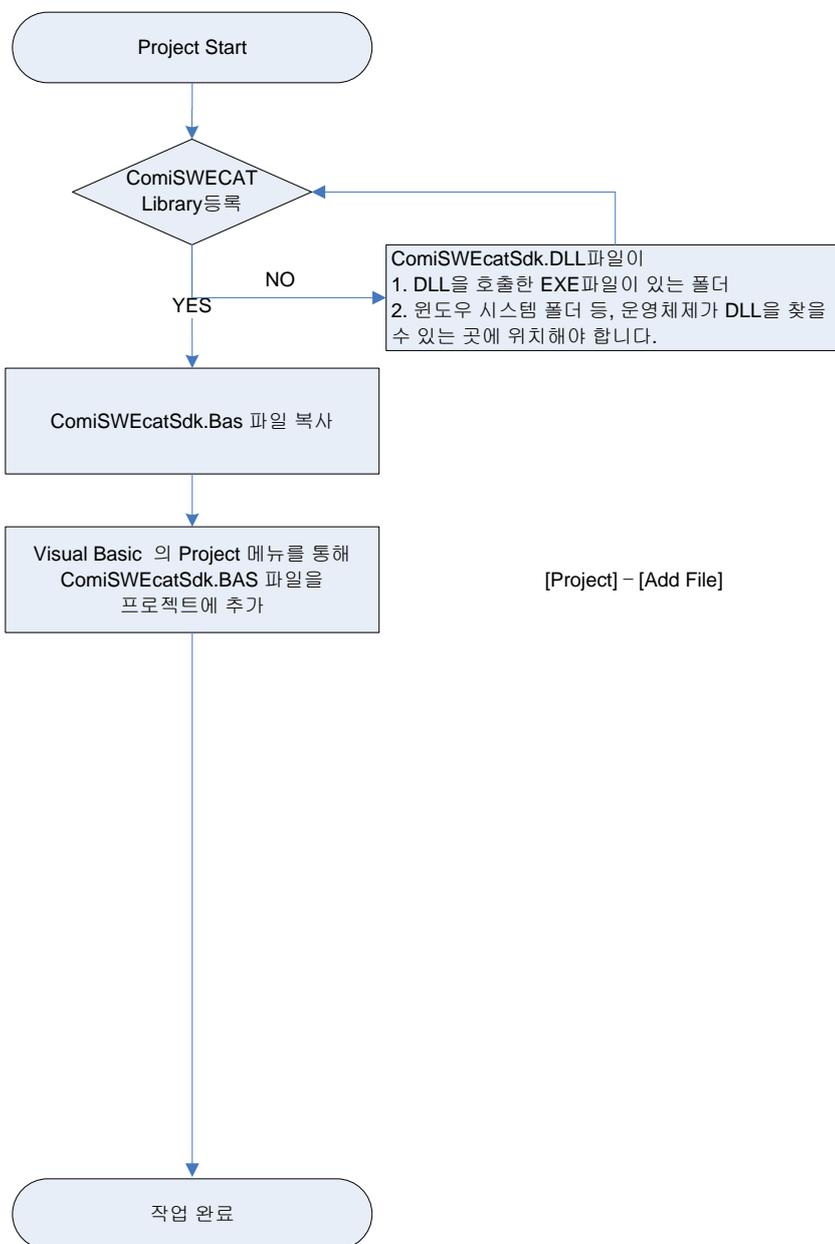


그림 3.3.5-1 Visual Basic 에서 ComiSWECAT 사용 순서도

Visual Basic 을 실행합니다. Visual Basic 이 시작되면 다음과 같은 신규 프로젝트 화면이 표시됩니다.



그림 3.3.5-2 새로운 프로젝트 생성

	<p>Visual Basic 에 대한 ComiSWECAT 지원 사항은 어떠합니까? ComiSWECAT 은 .NET 환경의 Visual Basic 까지 지원을 하고 있습니다.</p>
	<p>특히 미리 구성된 ComiSWECAT 인터페이스 파일은 간단히 프로젝트에 추가만 하시면, 바로 ComiSWECAT 를 사용하실 수 있도록 도움을 드리고 있습니다.</p>
	<p>만약, Visual Basic 사용 고객(顧客)님께서 ComiSWECAT 사용에 어려움이 있으시다면, 언제든지 저희 (주)커미조아 고객(顧客) 지원팀으로 연락주시기 바랍니다. 최선을 다해 지원해 드릴 것을 약속드립니다.</p>

신규프로젝트 창에서 'Standard EXE' 를 통해 표준 응용프로그램 개발을 시작합니다. '열기' 버튼을 누릅니다. 만약 이 화면이 나타나지 않으면, 아래의 화면과 같이 'File' 메뉴의 'New Project' 항목을 통해 신규 프로젝트를 시작합니다.

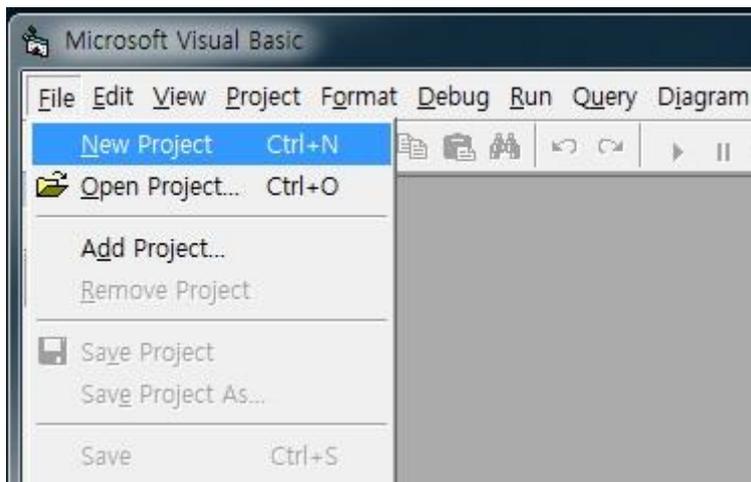


그림 3.3.5-3 신규 프로젝트의 시작

시작된 표준 EXE 응용프로그램 개발 메뉴에서 아래 그림과 같이 Project 메뉴를 통해 ‘Add File...’ 을 선택합니다.

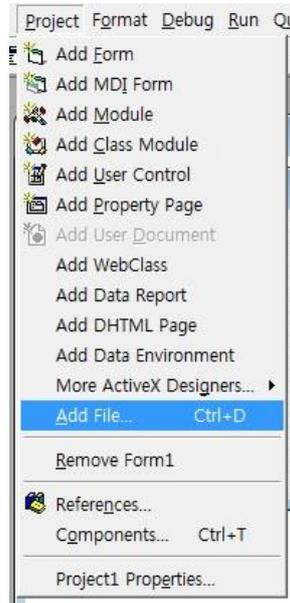


그림 3.3.5-4 프로젝트에 인터페이스 파일을 추가하기 위한 과정

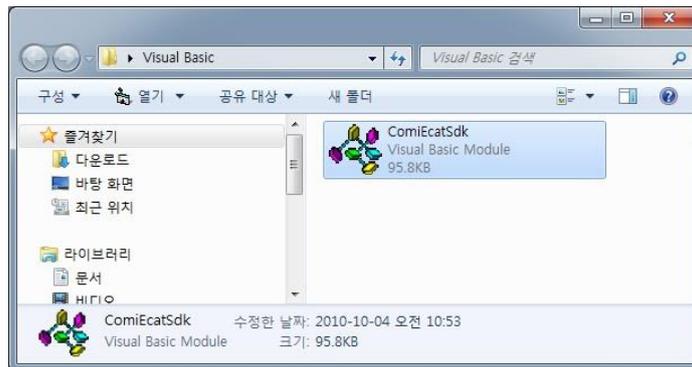


그림 3.3.5-5 프로젝트에 추가 대상이 되는 ComiSWECATSdk.BAS 파일

ComiSWECATSdk.BAS 파일을 프로젝트에 추가해 주시면, 명시적인 ComiSWECAT 로드가 이루어지게 되며, Visual Basic 의 프로젝트에서 함께 사용하실 수 있습니다.

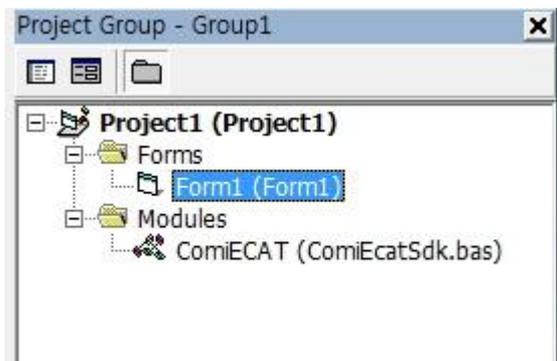


그림 3.3.5-6 프로젝트에 모듈로 추가된 ComiSWECATSdk.BAS 파일

추가된 인터페이스 파일을 통해 다음과 같이 실제 응용프로그램 구현이 가능합니다.

```

Project1 - Form1 (Code)
(General) (Declarations)

Call HomeGetConfig(GetActiveChannel, Home.ListIndex, 0, 1000, 0)
' 홈복귀 설정 값 반환

Call HomeSetSpeedPattern(GetActiveChannel, ecHome_s, 10000, 2000,
' 홈복귀 속도 설정

Call HomeMove(GetActiveChannel, GetDirection, GetIsBlocking)
' 홈복귀
    
```

그림 3.3.5-7 ComiSWECAT 을 통한 응용프로그램 구현

	Visual Basic 의 명시적인 ComiSWECAT 인터페이스에 대한 부가 안내
	ComiSWECAT 에 포함된 Visual Basic 용 인터페이스 파일은 별도의 DLL 라이브러리(ComiSWECAT)의 로드(Load) 및 언로드(Unload) 가 필요 없습니다. 따라서, 고객(顧客)님의 프로젝트의 Form1 에 추가된 인터페이스 파일을 통하여, 응용프로그램 구현을 바로 시작하실 수 있습니다.

ComiSWECAT Introduction

다년간의 보다 강력하고 편리한 라이브러리 기술개발을 통해 자신있게 제공하여 드리는 ComiSWECAT 은 편리한 함수명의 규칙(Rule)을 통해 사용자 편의성을 극대화 하였습니다. ㈜커미조아 ComiSWECAT 의 최신 기능과 숙련된 모션 제어 기술은 결코 흉내낼 수 없는 커미조아의 기술입니다. 지금 확인(確認)하십시오.

본 장에서는 ComiSWECAT 이 제공하는 라이브러리 인터페이스에 대한 자세한 설명(說明)을 안내합니다. ComiSWECAT 은 라이브러리 기능을 보다 강력하고 효율적으로 지원할 수 있는 다양한 런타임(Run-time) 인터페이스와 라이브러리의 다양한 기능을 직관적(直觀的)으로 제공합니다. 본 매뉴얼에서는 ComiSWECAT 에서 제공하는 라이브러리 함수에 대한 설명(說明)을 기능에 따라 그룹별로 수록하였습니다.



4 ComiSWECAT 소개

4.1 함수의 명명 규칙

ComiSWECAT 에서 제공하는 모든 함수는 다른 API 함수와 이름이 중복되는 것을 피하기 위하여 아래의 예와 같이 “sec”이라는 접두어가 붙습니다.

secDll_Load(), secDll_Unload(), secGn_LoadDevices(), secGn_UnloadDevices (),...

그리고 “sec” 접두어 바로 다음에는 해당 함수가 속하는 기능의 그룹을 대표하는 접두어가 이어집니다. 이렇게 한 이유는 동일한 기능 그룹에 속한 함수들을 쉽게 구분할 수 있고, 함수가 리스트될 때에 일반적으로 알파벳순으로 정렬되므로 동일 기능 그룹에 속한 함수들이 함께 리스트될 수 있도록 하기 위함입니다. 아래는 몇 가지 기능 그룹을 대표하는 접두어가 적용된 함수들의 예입니다.

- . General Functions (Gn): secGn_LoadDevices(), secGn_GetNumDevices(), ...
- . 환경설정 함수들 (Cfg): secCfg_Start(), secCfg_End(), ...
- . 원점복귀 관련 함수들 (Home): secmHomeCfg_SetMode(), secmHomeCfg_GetMode(), ...
- . 단축구동 관련 함수들 (Sx): secmSxMot_MoveStart(),ecmSxMot_Stop(), ...
- . 다축구동 관련 함수들 (Mx): secmMxMot_MoveStart(),ecmMxMot_Stop(), ...
- . 보간구동 관련 함수들 (Ix): secmIxMot_LineStart(),ecmIxCfg_MapAxes(), ...

4.2 데이터형 표기

당사의 ComiSWECAT 인터페이스는 매뉴얼에서 명시한 윈도우 표준 Dynamic Link Library 를 지원하는 어떠한 개발 환경에서도 사용 가능합니다. 하지만 데이터형에 대한 이름은 개발환경에 따라서 서로 다릅니다. 따라서 본 매뉴얼에서는 데이터의 형 표기를 표 6 과 같이 통일하여 표기합니다. 이에 대한 각 컴파일러의 대응되는 데이터형 표기는 표 6 을 참조하여 사용하시기 바랍니다.

그리고 본 매뉴얼에서는 “[in]”과 “[out]” 표기를 사용하여 매개변수가 함수에 전달되는 것인지 아니면 전달받는 것인지를 명시하였습니다. “[in]”은 함수에 값을 전달함을 의미하고, “[out]”은 함수로부터 값을 전달받는다는 것을 의미합니다. 단, 이 표기는 본 매뉴얼에서만 사용되는 것이며, 실제 헤더파일에는 표기되어 있지 않습니다.

Data type	Description	C/C++
t_char	1 바이트 ASCII 문자 코드	char
t_byte	부호가 없는 양의 정수	unsigned char
t_uchar	부호가 없는 양의 정수	unsigned char
t_bool	True 나 False 형태의 논리형	bool
t_i8	1 바이트 ASCII 문자 코드	char
t_ui8	부호가 없는 양의 정수	unsigned char
t_i16	부호가 있는 2 바이트 정수	short
t_word	부호가 없는 2 바이트 정수	unsigned short
t_ui16	부호가 없는 2 바이트 정수	unsigned short
t_i32	부호가 있는 4 바이트 정수	int

t_ui32	부호가 없는 4 바이트 정수	unsigned int
t_dword	부호가 없는 4 바이트 정수	unsigned int
t_f32	부호가 있는 4 바이트 실수	float
t_f64	부호가 있는 8 바이트 실수	double
t_success	True 나 False 형태의 논리형	bool
t_cmdidx	부호가 없는 4 바이트 정수	unsigned int

표 6 언어독립적 데이터 표기 및 각 언어별 대응 데이터 형

DLL functions

쥬커미조아는 ComiSWECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiSWECAT 라이브러리의 즐거움을 이제 함께 하십시오.

본 단원에서는 라이브러리 로드/언로드 와 관련된 가장 일반적인 함수들을 소개합니다. 고객(顧客) 여러분들께서는 라이브러리의 초기화와 사용을 위해서는 반드시 본 장을 읽어주시기 바랍니다. 구성되는 함수에는 ComiSWECAT 의 로드 및 언로드 에 대한 내용들로 구성되어 있습니다.



5 DLL Functions

5.1 함수 요약

Summary of Functions
□ t_success secDll_Load() 라이브러리를 사용자(使用者) 응용프로그램의 사용을 위해 로드(Load) 합니다.
□ t_success secDll_Unload() 라이브러리를 사용자(使用者) 응용프로그램의 사용 종료(終了)를 위해 언로드(Unload) 합니다.
□ BOOL secDll_IsLoaded () 라이브러리가 로드 되었는지 알 수 있는 역할을 합니다.

5.2 함수 설명

NAME	INFORMATION
secDll_Load - 라이브러리(Library) 로드	DLL Function VC++/BCB/.NET Level 1 위험 요소 없음
SYNOPSIS	
□ t_success secDll_Load()	

DESCRIPTION

ComiSWECAT 을 고객(顧客)님의 응용프로그램의 메모리 공간으로 호출합니다. 이 의미는 이 함수가 호출되는 순간 라이브러리는 고객(顧客)님의 프로그램 내부 함수처럼 호출할 수 있게 됩니다. 이 함수는 고객(顧客)님의 전체 프로그램에서 ComiSWECAT 을 사용하기 위한 수순으로서는 가장 먼저 호출되어야 합니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	DLL 을 로드하는데 실패하였음을 의미합니다.
1	DLL 을 <u>성공적으로</u> 로드하였음을 의미합니다.

SEE ALSO

* secDll_Unload

EXAMPLE

```
C/C++
#include "ComiSWECATSdk_Api.h"

void StartProgram(void) //만드시 함수 처음 부분에 작성
{
    // 이 함수의 반환값은 DLL 의 로드의 성공여부를 반환합니다.
    t_success nIsLoaded = secDll_Load ();
}
```

NAME

secDll_Unload

- 라이브러리(Library) 로드 해제

INFORMATION

 DLL Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```
□ t_success secDll_Unload ()
```

DESCRIPTION

ComiSWECAT 을 고객(顧客)님의 응용프로그램의 메모리 공간에서 해제합니다. 이 의미는 이 함수가 호출되는 순간 (주)커미조아의 ComiSWECAT 은 고객(顧客)님의 응용프로그램에서 제거됩니다. 이 함수는 고객(顧客)님의 전체 프로그램에서 ComiSWECAT 를 사용을 종료 하기 위한 수순으로서는 가장 나중에 호출되어야 합니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	DLL 을 언로드하는데 <u>실패</u> 하였음을 의미합니다.
1	DLL 을 <u>성공적</u> 으로 언로드하였음을 의미합니다.

SEE ALSO

* secDll_Load

EXAMPLE

```

C/C++
#include "ComiSWECATSdk_Api.h"

void EndProgram(void) //반드시 모든 함수 해제 후 DLL 해제
{
    // 이 함수의 반환값은 없습니다. 따라서 다른 에러처리는 필요하지 않습니다.
    secDll_Unload ();
}

```

NAME

secDll_IsLoaded

- 라이브러리(Library) 로드 상태 확인

INFORMATION DLL Function VC++/BCB/.NET Level 1 위험 요소 없음**SYNOPSIS**

□ BOOL secDll_IsLoaded ()

DESCRIPTION

ComiSWECAT 이 고객(顧客)님의 응용프로그램의 메모리 공간으로 호출 되었는지를 확인합니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	DLL 을 로드하는데 <u>실패</u> 하였음을 의미합니다.
1	DLL 을 <u>성공적으로</u> 로드하였음을 의미합니다.

EXAMPLE

C/C++

#include "ComiSWECATSdk_Api.h"

```
void Dll_IsLoaded(void)
{
    if(secDll_IsLoaded()) //로드 성공
    else // 로드 실패
}
```

General functions

㈜커미조아는 ComiSWECAT를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서 저희 ㈜커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiSWECAT 라이브러리의 즐거움을 이제 함께 하십시오.

이

편

단원에서는 장치의 로드/언로드 등 장치와 관련된 가장 일반적인 함수들을 소개합니다. 구성되는 함수에는 장치의 로드 및 언로드, 시스템에 연결된 장치 개수 등에 대한 내용을 다룹니다.



6 General Functions

6.1 함수 요약

Summary of Functions	
<input type="checkbox"/> t_success secGn_LoadDevices ([out] t_i32 *ErrCode)	라이브러리가 로드된 상태에서 장치를 로드(Load) 하는 역할을 합니다.
<input type="checkbox"/> t_success secGn_UnloadDevices ([out] t_i32 *ErrCode)	라이브러리가 로드된 상태에서 장치를 언로드(Unload) 하는 역할을 합니다.

6.2 함수 설명

NAME	INFORMATION
secGn_LoadDevices - 장치 로드	 General Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음
SYNOPSIS	
<pre>□ t_success secGn_LoadDevices([out] t_i32 *ErrCode)</pre>	

DESCRIPTION

시스템에 설치된 하드웨어 장치를 로드합니다. 이 함수는 ComiSWECAT의 다른 함수가 호출되기 전에 반드시 한번은 수행되어야 합니다. 일반적으로 프로그램의 시작부분에서 수행해주면 됩니다.

PARAMETER

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type)을 가지고 있습니다.

Value	Meaning
0	장치를 로드하는데 <u>실패</u> 하였음을 의미합니다.
1	장치를 <u>성공</u> 적으로 로드하였음을 의미합니다.

SEE ALSO

* secGn_UnloadDevices

EXAMPLE

C/C++

```
#include "ComiSWECATSdk_Api.h"

t_32 Error_Num = 0;          // 에러 코드 저장 변수

void Device_Load(void)
{
    if(!secGn_LoadDevices(&Error_Num)){ //장치 로드
        //로드 실패시 예외 처리
    }
} /* Device_Load (void)함수의 끝 */

void Device_UnLoad(void)
{
    secGn_UnloadDevices(&Error_Num); //디바이스 언로드
}

```

<h2>NAME</h2> <p>secGn_UnloadDevices</p> <p>- 장치 언로드</p>	INFORMATION
	 General Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

□ t_success secGn_UnloadDevices([out] t_i32 *ErrCode)

DESCRIPTION

시스템에 설치된 하드웨어 장치를 언로드합니다. 이 함수는 ComiSWECAT 장치 사용을 종료하기 위해 명시적으로 호출되어야 합니다. 일반적으로 프로그램의 종료부분에서 수행해주면 됩니다.

PARAMETER

- ▶ ErrCode: 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	장치를 로드하는데 실패하였음을 의미합니다.
1	장치를 성공적으로 로드하였음을 의미합니다.

SEE ALSO

* secGn_LoadDevices

EXAMPLE

* secGn_LoadDevices EXMPLE 참고

Net Interface functions

쥬커미조아는 ComiSWECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiSWECAT 라이브러리의 즐거움을 이제 함께 하십시오.

본 단원에서는 라이브러리 로드/언로드 와 관련된 가장 일반적인 함수들을 소개합니다. 고객(顧客) 여러분들께서는 라이브러리의 초기화와 사용을 위해서는 반드시 본 장을 읽어주시기 바랍니다. 구성되는 함수에는 ComiSWECAT 의 로드 및 언로드 에 대한 내용들로 구성되어 있습니다.



7 Net Interface Functions

7.1 함수 요약

Summary of Functions

□ t_ui32 secNet_ScanSlaves([in] t_i32 NetID, [out] t_i32 *ErrCode)
장치에 연결되어 있는 슬레이브를 검색합니다.

7.2 함수 설명

NAME	INFORMATION
<p>secNet_ScanSlave</p> <p>- 슬레이브 검색</p>	<p> Net Interface Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>
SYNOPSIS	
<pre>□ t_ui32 secNet_ScanSlave([in] t_i32 NetID, [out] t_i32 *ErrCode)</pre>	

DESCRIPTION

이 함수는 사용자가 사용 할 네트워크에 연결되어 있는 모든 슬레이브를 검색하여 그 개수를 반환 합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 슬레이브의 개수를 반환합니다.

EXAMPLE

C/C++

```
#include "ComiSWECATSdk_Api.h"

void Device_Load(void)
{
    t_i32 NetID = 0;           // 네트워크 ID(0 base)
    t_32 Error_Num = 0;      // 에러 코드 저장 변수
    t_ui32 SlaveNum = 0;     // NetID 에 연결된 Slave 개수

    SlaveNum = secNet_ScanSlaves(NetID, &Error_Num);
}
```

Slave I/f functions

췁커미조아는 ComiSWECAT를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 췁커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiSWECAT 라이브러리의 즐거움을 이제 함께 하십시오.

본 단원에서는 라이브러리 로드/언로드 와 관련된 가장 일반적인 함수들을 소개합니다. 고객(顧客) 여러분들께서는 라이브러리의 초기화와 사용을 위해서는 반드시 본 장을 읽어주시기 바랍니다. 구성되는 함수에는 ComiSWECAT 의 로드 및 언로드 에 대한 내용들로 구성되어 있습니다.



8 Slave I/F Functions

8.1 함수 요약

Summary of Functions
<p>❑ t_success secSlv_GetProdInfo_A (t_i32 NetID, t_i32 SlaveIdx, TEcSlvProdInfoEx *pProdInfoBuf, t_i32 *ErrCode);</p> <p>장치에 연결된 슬레이브 정보를 얻어옵니다.</p>
<p>❑ t_success secSlv_GetState_A (t_i32 NetID, t_i32 SlaveIdx, TEcSlvState *pSlaveStateBuf, t_i32 *ErrCode);</p> <p>장치에 연결된 슬레이브 상태정보를 얻어옵니다.</p>

8.2 함수 설명

NAME	INFORMATION
secSlv_GetProdInfo_A - 슬레이브 Product 정보 확인	 Slave I/F Function  VC++/BCB/.NET  Level 1  위험 요소 없음
SYNOPSIS	
<pre> □ t_success secSlv_GetProdInfo_A([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] TEcSlvProdInfoEx *pProdInfoBuf, [out] t_i32 *ErrCode) </pre>	

DESCRIPTION

secSlv_GetProdInfo_A() 함수는 하나의 슬레이브에 대한 Product 정보를 반환 합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호(0)
- ▶ SlaveIndex : 슬레이브 번호(1 Base)
- ▶ pProdInfoBuf : Product 에 대한 정보를 포함한 구조체 변수.

Parameter	Type	Meaning
state	t_ui16	Slave 상태
ALstatuscode	t_ui16	AL 상태 코드
configadr	t_ui16	Configured address(Auto Configured)
aliasadr	t_ui16	Alias address(Not Used)
eep_man	t_ui32	Manufacturer/Vender ID
eep_id	t_ui32	Slave 를 구별하는 Product Code
eep_rev	t_ui32	Reserve Number.
Itype	t_ui16	Interface type(Future Use)
Dtype	t_ui16	Device type(Future Use)
islost	t_ui8	Slave 연결상태
name[63]	char	Slave Product Name

* 슬레이브의 state 코드는 다음과 같습니다.

Value	Meaning
0x0	Disconnect State
0x1	Init. State
0x2	Pre-operational State
0x3	Boot State
0x4	Safe-operational State
0x8	Operational State
0x10	Error or ACK Error State

* 슬레이브의 AL.statuscode 는 다음과 같습니다.

Value	Meaning
0x0000	No Error
0x0001	Unspecified Error
0x0002	No memory
0x0011	Invalide requested state change
0x0012	Unknown requested state
0x0013	Bootstrap not supported
0x0014	No valid firmware
0x0015	Invalid mailbox configuration
0x0016	Invalid mailbox configuration
0x0017	Invalid sync manager configuration
0x0018	No valid input available
0x0019	No valid output
0x001A	Synchronization error
0x001B	Sync manger watchdog
0x001C	Invalid sync manger types
0x001D	Invalid output configuration
0x001E	Invalid input configuration
0x001F	Invalid watchdog configuration
0x0020	Slave needs cold start
0x0021	Slave needs INIT
0x0022	Slave needs PREOP
0x0023	Slave needs SAFEOP
0x0024	Invalid input mapping
0x0025	Invalid ouput mapping
0x0026	Inconsistent settings
0x0027	Freerun not supported
0x0028	Synchronization not supported
0x0029	Freerun needs 3 buffer mode
0x002A	Background watchdog
0x002B	No valid inputs and outputs
0x002C	Fatal sync error
0x002D	No sync error
0x002E	Invalid input FMMU configuration
0x0030	Invalid DC SYNC configuration
0x0031	Invalid DC latch configuration
0x0032	PLL error

0x0033	DC sync IO error
0x0034	DC sync timeout error
0x0035	DC invalid sync cycle time
0x0042	MBX_EOE
0x0043	MBX_COE
0x0044	MBX_FOE
0x0045	MBX_SOE
0x004F	MBX_VOE
0x0050	EEPROM no access
0x0051	EEPROM error
0x0060	Slaver restarted locally
0x0061	Device identification valude updated
0x00F0	Application controller available
0xFFFF	Unknown

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	정보를 가져오는데 실패 했음을 의미합니다.
1	정보를 가져오는데 성공 했음을 의미합니다.

EXAMPLE

C/C++

```
#include "ComiSWECATSdk_Api.h"
```

```
void GetSlaveInfo(void)
{
    TEcSlvProdInfoEx ProdInfoBuf // Slave 대한 구조체 변수.
    t_32 Error_Num = 0;           //함수 별 에러 코드 저장 변수
    t_i32 NetID = 0;              // 네트워크 ID(0 base)
    t_i32 Slaveidx = 1;          // Slave 정보를 얻어오기 원하는 Slave Index(1 base)

    // 이 함수를 이용하여 슬레이브의 Product 정보 구조체를 얻어 옴
    if(!secSlv_GetProdInfo_A(NetID, SlaveIdx, &ProdInfoBuf, &Error_Num))
    {
        // 슬레이브 Product 정보 확인 실패 시 예외 처리
    }
}
```

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">secSlv_GetState_A</p> <p style="margin: 0;">- 슬레이브 State 확인 (슬레이브 EtherCAT Machine State 확인)</p>	<h3 style="margin: 0;">I N F O R M A T I O N</h3> <ul style="list-style-type: none"> Slave I/F Function VC++/BCB/.NET Level 1 위험 요소 없음
--	--

SYNOPSIS

□ t_success secSlv_GetState_A([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] TEcSlvState *pSlaveStateBuf, [out] t_i32 *ErrCode)

DESCRIPTION

secSlv_GetAlState_A() 함수는 슬레이브에 대한 State 값을 반환합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlaveIndex : 1-base 슬레이브 번호(슬레이브가 연결되어 있는 순서)
- ▶ pSlaveStateBuf : 슬레이브 상태정보를 포함한 구조체 변수

Parameter	Meaning
state	Slave 상태(secSlv_GetProdInfo_A 의 state 코드 참조)
ALstatuscode	AL 상태 코드(secSlv_GetProdInfo_A 의 ALstatuscode 참조)
islost	Slave 연결상태(0:정상, 1:disconnected)

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	정보를 가져오는데 실패 했음을 의미합니다.
1	정보를 가져오는데 성공 했음을 의미합니다.

EXAMPLE

C/C++

```
#include "ComiSWECATSdk_Api.h"
```

```
void GetSlaveState(void)
```

```
{
```

```
    TEcSlvState SlaveStateBuf    // Slaver 상태에 대한 구조체 변수.
```

```
    t_32 Error_Num = 0;          // 함수 별 에러 코드 저장 변수
```

```
    t_i32 NetID = 0;             // Network ID(0 base)
```

```
    t_i32 Slaveidx = 1;          // State 를 얻어오기 원하는 Slave Index(1 base)
```

```
    // 슬레이브 State 확인
```

```
    if(secSlv_GetState_A(NetIdx, SlaveIdx, &SlaveStateBuf, &Error_Num))
```

```
    {
```

```
        // 슬레이브 상태 정보 확인 실패 시 예외 처리
```

```
    }
```

```
}
```

Digital I/O functions

쥬커미조아는 ComiSWECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiSWECAT 라이브러리의 즐거움을 이제 함께 하십시오.

본 단원에서는 라이브러리 로드/언로드 와 관련된 가장 일반적인 함수들을 소개합니다. 고객(顧客) 여러분들께서는 라이브러리의 초기화와 사용을 위해서는 반드시 본 장을 읽어주시기 바랍니다. 구성되는 함수에는 ComiSWECAT 의 로드 및 언로드 에 대한 내용들로 구성되어 있습니다.



9 Digital I/O Functions

9.1 함수 요약

Summary of Functions
<p>❑ t_bool secdiGetOne([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode) 대상 디지털 입력(Digital Input) 전역채널에 대해 입력 상태를 확인합니다.</p>
<p>❑ t_dword secdiGetMulti([in] t_i32,NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode) 다중 디지털 입력(Digital Input) 전역채널에 대해 입력 상태를 확인합니다.</p>
<p>❑ t_i32 secdiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 Digital Input 채널 개수를 반환합니다.</p>
<p>❑ t_success secdoPutOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [in] t_bool OutState, [out] t_i32 *ErrCode) 대상 디지털 출력(Digital Output) 전역채널에 대해 출력을 발생합니다.</p>
<p>❑ t_success secdoPutMulti([in] t_i32,NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [in] t_dword dwOutStates, [out] t_i32 *ErrCode) 다중 디지털 출력(Digital Output) 전역채널에 대해 출력을 발생합니다.</p>
<p>❑ t_bool secdoGetOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode) 대상 디지털 출력(Digital Output) 전역채널에 대해 출력 상태를 확인합니다.</p>
<p>❑ t_dword secdoGetMulti([in] t_i32,NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode) 다중 디지털 출력(Digital Output) 전역채널에 대해 출력 상태를 확인합니다.</p>
<p>❑ t_i32 secdoGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 Digital Output 채널 개수를 반환합니다.</p>

9.2 함수 설명

NAME	INFORMATION
secdiGetOne - 단일 채널에 대한 입력 상태 반환	 Digital I/O Function  VC++/BCB/.NET  Level 1  위험 요소 없음
SYNOPSIS	
<pre>□ t_bool secdiGetOne([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode)</pre>	

DESCRIPTION

이 함수는 단일 채널에 대한 디지털 입력 상태를 반환 합니다.

PARAMETER

- ▶ NetID : Network 번호(0 base)
- ▶ DiChannel : 디지털 입력 전역채널 번호(0 base)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 채널의 디지털 입력 상태 값을 반환 합니다.

EXAMPLE

C/C++

```
#include "ComiSWECATSDk_Api.h"
```

```
void DI_GetOne(void)
{
    t_i32 NetID = 0;           // Network ID
    t_bool State = false;     // DI 결과 값 저장 변수
    t_32 Error_Num = 0;      // 함수 별 에러 코드 저장 변수
    t_ui32 DiChannel = 0;    // 입력상태를 알고자하는 채널번호

    State = secdiGetOne(NetID, DiChannel, &Error_Num);
}
```

```
if(State == 1)
{
    //DI 결과 값이 1 일때 처리
}
else
{
    //DI 결과 값이 0 일때 처리
}
} // DI_GetOne 함수 종료
```

<h1>NAME</h1> <p>secdiGetMulti</p> <p>- 다중 채널에 대한 입력 상태 반환</p>	INFORMATION
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```

❑ t_dword secdiGetMulti([in] t_i32 NetID, [in] t_ui32 IniChannel, [in] t_ui8
  NumChannels, [out] t_i32 *ErrCode)

```

DESCRIPTION

이 함수는 다중 채널에 대한 디지털 입력 상태를 반환 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ IniChannel : 시작 전역채널 번호
- ▶ NumChannels : 시작 전역채널로 부터 몇 개의 채널의 상태를 확인할 것인지에 대한 값 (최대 32 개 까지)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 다중 채널의 디지털 입력 상태 값이 채널개수(NumChannels) 만큼 비트순서로 저장되어 반환 합니다.

31 번 비트	...	1 번비트	0 번비트
(iniChannel+ 31)번 채널 상태	...	(IniChannel+ 1)번 채널 상태	IniChannel 번 채널 상태

EXAMPLE

```

C/C++
#include "ComiSWECATSdk_Api.h"
#define NUMCHNNELS 2 //사용자가 사용 할 채널 수를 입력

void DI_GetMulit(void)
{

```

```
t_i32 NetID = 0;           // Network ID
t_i32 NetID = 0;           // Network ID
t_32 Error_Num = 0;       //함수 별 에러 코드 저장 변수
t_i16 DI_InitCh = 0;      //시작 채널 번호 저장 변수
t_dword State = 0;        // DI_Multi 결과 값 저장 변수

State = secDiGetMulti (NetID, DI_InitCh, NUMCHNNELS, &Error_Num);

//*****
// State 에 결과 값에는 DI 결과 값이 비트순서대로 저장
//ex)0 번 채널부터 1 번까지 모든 채널이 ON 이라면 결과값은 0x11 즉 10 진수 3
//*****//

} // DI_GetMulit 함수 종료
```

NAME

secdiGetNumChannels
- 디지털 입력 채널 개수 반환

INFORMATION

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```
□ t_i32 secdiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)
```

DESCRIPTION

이 함수는 장치에 연결 되어 있는 디지털 입력 채널 개수 반환 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 디지털 입력 채널 개수를 반환 합니다.

EXAMPLE

C/C++

```
#include "ComiSWECATSdk_Api.h"

void DI_GetChCount(void)
{
    t_i32 NetID = 0;           // Network ID
    t_32 Error_Num = 0;       // 에러 코드 저장 변수
    t_i32 All_DiCount = 0;    // 마스터에 연결 된 모든 DI 채널 개수 저장 할 변수

    // 마스터에 연결 된 모든 DI 채널 개수를 로드
    All_DiCount = secdiGetNumChannels(NetID, &Error_Num);
}
```

<h2>NAME</h2> <p>secdoPutOne / secdoGetOne</p> <p>- 단일 채널에 대한 출력 발생 및 출력 상태 확인</p>	INFORMATION
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_success secdoPutOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [in] t_bool OutState, [out] t_i32 *ErrCode)
- t_bool secdoGetOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode)

DESCRIPTION

secdoPutOne 함수는 단일 채널에 대한 디지털 출력 상태를 발생시키며, secdoGetOne 함수는 단일 채널에 대한 디지털 출력 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ DoChannel : 디지털 출력 전역채널 번호(0 base)
- ▶ OutState : secdoPutOne() 함수의 인자이며, 단일 채널에 대한 디지털 출력상태를 발생합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* secdoPutOne : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	출력발생에 실패 했음을 의미합니다.
1	출력발생에 성공 했음을 의미합니다.

* secdoGetOne : 이 리턴값은 출력 상태 값을 반환 합니다.

Value	Meaning
0	DO 출력 OFF
1	DO 출력 ON

EXAMPLE

```

C/C++

#include "ComiSWECATSDk_Api.h"
#define DO_CHNUMBER 0 //사용자가 출력을 원하는 DO 채널 번호

void DO_PutOne(void)
{
    t_i32 NetID = 0;           // Network ID
    t_32 Error_Num = 0;      // 에러 코드 저장 변수
    t_bool State = 1;        // 출력 값

    if(!secdoPutOne (NetID, DO_CHNUMBER, State, &Error_Num))
    {
        //DO 실패 시 예외 처리
    }
} // DO_PutOne 함수 종료

void DO_GetOne(void)
{
    t_i32 NetID = 0;           // Network ID
    t_32 Error_Num = 0;      // 에러 코드 저장 변수
    t_bool State = 0;        // 출력상태 값

    State = secdoGetOne (NetID, DO_CHNUMBER, &Error_Num);
    if( State == true )
    {
        //DO 결과가 1 일 경우 처리
    }
    else
    {
        //DO 결과가 0 일 경우 처리
    }
} // DO_GetOne 함수 종료

```

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">secdoPutMulti / secdoGetMulti</p> <p style="margin: 0;">- 다중 채널에 대한 출력 발생 및 출력 상태 확인</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: left; padding: 2px;">INFORMATION</th> </tr> <tr> <td style="padding: 2px;"> Digital I/O Function</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> VC++/BCB/.NET</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> Level 1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> 위험 요소 없음</td> <td style="padding: 2px;"></td> </tr> </table>	INFORMATION		Digital I/O Function		VC++/BCB/.NET		Level 1		위험 요소 없음	
INFORMATION											
Digital I/O Function											
VC++/BCB/.NET											
Level 1											
위험 요소 없음											

SYNOPSIS

- t_success secdoPutMulti([in] t_i32 NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [in] t_dword dwOutStates, [out] t_i32 *ErrCode)
- t_dword secdoGetMulti([in] t_i32 NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)

DESCRIPTION

secdoPutMulti 함수는 다중 채널에 대한 디지털 출력 상태를 발생시키며, secdoGetMulti 함수는 다중 채널에 대한 디지털 출력 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ IniChannel : 시작 전역채널 번호(0 base)
- ▶ NumChannels : 시작 전역채널로 부터 몇 개의 채널의 상태를 확인할 것인지에 대한 값 (최대 32 채널)
- ▶ dwOutStates : secdoPutMulti() 함수의 인자이며, 다중 채널에 대한 디지털 출력상태를 발생시킵니다.

31 번 비트	...	1 번비트	0 번비트
(iniChannel+ 31) 번 채널 상태	...	(IniChannel+ 1)번 채널 상태	IniChannel 번 채널 상태

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* secdoPutMulti : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	출력발생에 실패 했음을 의미합니다.

1	출력발생에 성공 했음을 의미합니다.
---	---------------------

* secdoGetMulti : 이 리턴값은 출력 상태 값을 반환 합니다.

31 번 비트	...	1 번비트	0 번비트
(iniChannel+ 31) 번 채널 상태	...	(IniChannel+ 1)번 채널 상태	IniChannel 번 채널 상태

EXAMPLE

```
C/C++
```

```
#include "ComiSWECATSDk_Api.h"

#define NUMCHNNELS 2 //사용자가 확인 할 채널 수를 입력

void DO_PutMulti(void)
{
    t_i32 NetID = 0; // Network ID
    t_32 Error_Num = 0; // 에러 코드 저장 변수
    t_i16 DO_InitCh = 0; //시작 채널 번호 저장 변수
    t_dword dwOutStates = 0x3; // 0 번, 1 번 채널 ON

    if(!secdoPutMulti (NetID, DO_InitCh, NUMCHNNELS, dwOutStates, &Error_Num))
    {
        //DO 실패 시 예외 처리
    }
} // DO_PutMulti 함수 종료

void DO_GetMulti(void)
{
    t_i32 NetID = 0; // Network ID
    t_32 Error_Num = 0; // 에러 코드 저장 변수
    t_i16 DO_InitCh = 0; //시작 채널 번호 저장 변수
    t_dword dwOutStates = 0; // Output 상태 리턴

    dwOutStates = secdoGetMulti (NetID, DO_InitCh, NUMCHNNELS, &Error_Num);
} // DO_GetMulti 함수 종료
```

NAME

secdoGetNumChannels
- 디지털 출력 채널 개수 반환

INFORMATION

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```
□ t_i32 secdoGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)
```

DESCRIPTION

이 함수는 단일 채널에 대한 디지털 출력 상태를 반환 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 디지털 출력 채널 개수를 반환 합니다

EXAMPLE

C/C++

```
#include "ComiSWECATSdk_Api.h"

void DO_GetChCount(void)
{
    t_i32 NetID = 0;           // Network ID
    t_32 Error_Num = 0;       // 에러 코드 저장 변수
    t_i32 All_DOCount = 0;    //마스터에 연결 된 모든 DI 채널 개수 저장 할 변수

    All_DOCount = secdoGetNumChannels(NetID, &Error_Num);
    // 마스터에 연결 된 모든 DO 채널 개수를 로드
}
```

ComiSWECAT Manual / Appendix

부록편 내용 요약

A. ComiSWECAT Error Code

ComiSWECAT Error Code 는 고객(顧客) 여러분들께서 API 사용시 반환되는 Error Code 를 안내하고 있습니다. 이 내용은 전체적인 함수에 관련되어 있는 내용으로서 보다 정확하게 정리하였습니다. 필요한 사항들을 반드시 참조하시어, 모든 에러 상황에 대한 이해(理解)와 도움이 되시기를 바랍니다.

B. Frequently Asked Questions (FAQ)

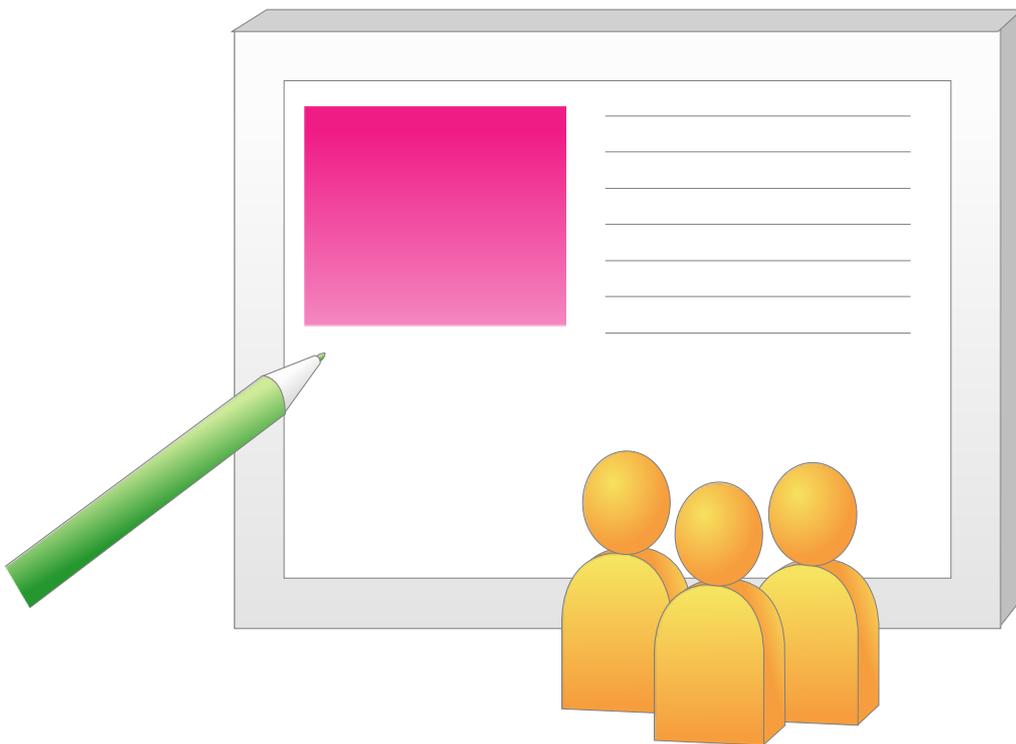
고객님들께서 자주 접하시게 되는 질문 사항이나 현상에 대해서, 보다 빠르고 정확하게 대응하실 수 있도록 빈번하게 질문되는 내용에 대해서 다루고 있습니다. 실제 모션 구동 상황에서 발생할 수 있는 사항을 비롯하여, 하드웨어, 소프트웨어에 대한 문의 사항들을 저희 (주)커미조아로 문의 하시기 전에 본 장을 통해서, 질문 사항에 대한 답변을 먼저 확인 하실 수 있도록 준비되어 있습니다.

C. Index of ComiSWECAT Functions

커미조아의 모션 라이브러리 매뉴얼에서 보다 쉽고 빠르게 함수를 확인할 수 있는 안내 페이지를 제공하고 있습니다. PDF화된 문서에서 보다 빠른 시간으로 함수를 찾을 수 있도록 하이퍼 링크(Hyper Link) 기능을 제공하고 있으며, 페이지와 함수명이 바로 연결되었으므로, 편리하게 원하는 함수의 설명을 확인 할 수 있습니다.

ComiSWECAT Error Code

ComiSWECAT의 모든 함수에는 ErrorCode를 반환하는 매개 변수가 있습니다. 이 매개변수를 통해 모션 동작시 나타날 수 있는 에러 상황들을 ErrorCode로 표현하여 매개 변수로 반환하게 됩니다. 이번 장에서는 이 ComiSWECAT에서 반환되는 모든 ErrorCode를 안내드리기 위해 준비하였습니다.



I ComiSWECAT Error Code 안내

Error Code	Define	Meaning
0	ecERR_NONE	에러 없음
-20	ecERR_INVALID_NETID	잘못된 Network ID 입력
-25	ecERR_INVALID_SLAVEID	잘못된 슬레이브 주소나 슬레이브 인덱스 입력
-30	ecERR_INVALID_CHANNEL	잘못된 축번호나 채널번호 입력
-60	ecERR_INVALID_FUNC_ARG	함수의 매개변수가 유효하지 않는 값이 입력
-70	ecERR_NULL_WDMNETCTXT	WDM 드라이버에서 제공하는 공유메모리 포인터가 NULL 인 경우
-90	ecERR_INVALID_LOG_MEM_ADDR	Logical Memory의 주소값이 올바르지 않은 경우
-100	ecERR_INVALID_IO_CHAN_MAP_DATA	I/O 채널 맵핑 데이터의 일부 멤버값이 올바르지 않은 경우
-50001	secERR_NULL_FILENAME	파일 메모리 포인터가 NULL 인 경우
-50002	secERR_NULL_ECACHECKPROC	EcatCheckProc 쓰레드 생성 실패한 경우
-50003	secERR_NONE_NETADAPTER	Network Adapter가 발견되지 않은 경우
-50004	secERR_NOT_OP_ALLSLAVE	일부 또는 전체 Slave가 OP 상태에 있지 않는 경우
-50005	secERR_NULL_BUFFER	버퍼가 NULL인 경우
-50006	secERR_EEPROM_READ	EEPROM Read 실패한 경우
-50007	secERR_EEPROM_WRITE	EEPROM Write 실패한 경우
-50008	secERR_EEPROM_FILE	EEPROM File 읽기가 실패한 경우
-50009	secERR_INVALID_EEPROM_RW_MODE	지원되지 않는 EEPROM Read/Write Mode 요청시
-50010	secERR_OVERSIZE_FIRMWARE	펌웨어 크기가 너무 큰 경우
-50011	secERR_FOE_WRITE	FOE Write 실패한 경우
-50012	secERR_FIRMWARE_FILE	Firmware 파일 읽기가 실패한 경우
-50013	secERR_TIMEOUT_BOOTMODE	BOOT 모드로 요청에 대해 Timeout된 경우
-50014	secERR_SDO_READ	SDO Read 실패시
-50015	secERR_SDO_WRITE	SDO Write 실패시
-50016	secERR_NOT_SUPPORTED_DEVICE	지원되지 않는 Device

Frequently Asked Questions

주로 자주 발생하는 질문이나, 고객(顧客) 기술 지원에 대한 내용은 고객(顧客)님의 제품 개발 및 모션 라이브러리 운용시에 많은 도움이 될 수 있습니다. 다양한 개발 환경에서 발생할 수 있는 여러가지 문제들을 세심하고 주의깊게 다루어, 고객(顧客)님들께 진정으로 도움이 될 수 있는 내용을 준비하였습니다.

FAQ

부록에서는 ㈜커미조아의 ComiSWECAT 사용에 있어, 문의 사항이나 궁금하신 점을 FAQ 로 안내하였습니다. 각 FAQ 는 ㈜커미조아의 웹 사이트(<http://www.comizoa.com>) 이나 별도로 등록된 고객(顧客)님의 정보를 통해 전달 될 수 있도록 노력하겠습니다. 본 FAQ 장을 고객(顧客) 문의를 하시기전에 미리 확인(確認)하시어, 좋은 참고가 되시기를 간절히 바라겠습니다.



II Frequently Asked Questions (FAQ)

II.I Visual Studio 2005

Q : Microsoft ® Visual Studio 2005 에서 MFC WIZARD 선택 단계에서 [Use Unicode Libraries] 항목이 확인(確認)된 상태에서 프로젝트를 시작한 다음, 빌드 하였을 경우 아래와 같은 에러가 발생하게 됩니다.

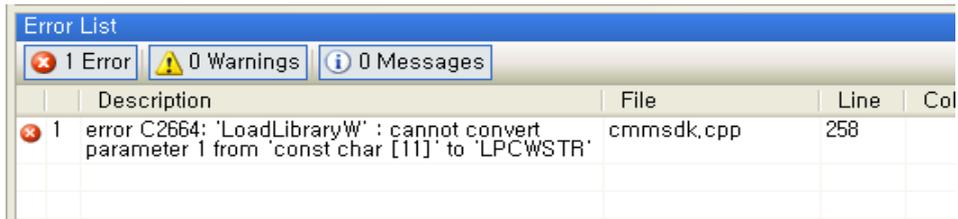


그림 9.2-1 Unicode 로 되어 있을 경우 발생하는 에러 화면

A : 에러 발생의 이유는 ComiSWECAT 은 표준 ANSI Libraries API 함수를 사용하는데, 현재 프로젝트의 설정이 Unicode 로 되어 있어 다음과 같은 에러가 발생할 수 있습니다. 해당 문제를 해결 하는 방법은 다음과 같습니다. MS VC++ 의 Solution Explorer 에서 현재 작업중인 프로젝트를 선택합니다.

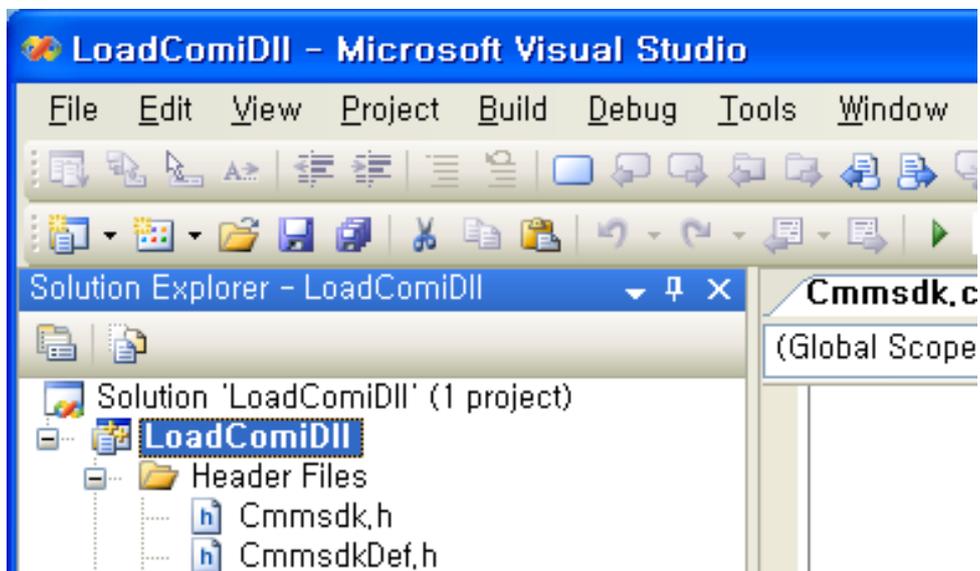


그림 9.2-2 사용자 생성 프로젝트 선택 화면

메뉴에서 [Project]->[Properties]를 선택하여 [Property Page]창을 엽니다.

II.II Visual Basic

Q : Visual Basic 6.0 에서 사용하던 COMIZOA SDK Library 를 Visual Studio .NET 에서 사용하면 오류가 발생한다.

A : Microsoft 사가 인정한 Visual Studio 오류로서 디자인 타임 라이선스가 존재하지 않을 경우에 발생합니다. Visual Basic 6.0 으로 작성한 프로젝트를 Visual Studio.NET 으로 업그레이드하여 사용하고자 할 때 발생할 수 있는 오류 사항입니다. COMIZOA SDK Library 에는 라이선스가 없으므로 문제가 발생한다면 프로젝트에 포함되어있는 다른 ActiveX Control 로 인하여 문제가 발생할 수 있는 여지가 있습니다.

<http://support.microsoft.com/default.aspx?scid=kb;ko;318597>

마이크로 소프트웨어에서 제공하는 위의 주소를 통해 자세한 내용 및 해결방안을 찾아보실 수 있습니다.

II.III Borland C++ Builder

Q : C++ Builder 5 버전과 6 버전에서 COMIZOA OLE Components (ComiSliderCtrl.ocx)가 Import 되어 있는데 보이지 않는다.

A : Visual Basic 으로 작성한 OCX 를 C++ Builder 에서 사용할 때 발생하는 문제입니다. ..\SDK\COMIZOA Components\Borland Package\C++ Builder\ 안에 각 버전별로 Component 관련 파일이 들어 있습니다. 이 파일들을 각각 위치에 맞게 복사해 주시면 됩니다.

새로운 환경에서 ComiSliderCtrl.ocx 사용하여 프로그램을 작성하시려면 다음과 같은 방법으로 등록하여 주시면 됩니다.

* Builder 5 에서 ComiSliderCtrl.ocx 사용방법.

1. 윈도우 시스템폴더에 OCX 를 복사한 후 regsvr32 명령으로 OCX 를 등록시킵니다.
(시작->실행 메뉴에서 : regsvr32 c:\windows\system32\ComiSliderCtrl.ocx)

2. ComiSliderCtrl.ocx 를 [..\CBuilder5\Bin\] 폴더로 복사합니다.
tibimp.exe 프로그램을 이용하여 Import Type Library File 을 생성합니다.

```
C:\Program Files\Borland\CBuilder5\BIN> tibimp.exe -Yu -Ya ComiSliderCtrl.ocx
ComiSliderCtrl_TLB.h
ComiSliderCtrl_TLB.cpp
ComiSliderCtrl_OCX.h
ComiSliderCtrl_OCX.cpp
ComiSliderCtrl_OCX.dcr
총 5 가지 파일이 생성됩니다.
```

3. 계속하여, 시작-> 실행 메뉴를 통해 'cmd' 를 입력하여, 명령프롬프트를 실행합니다.

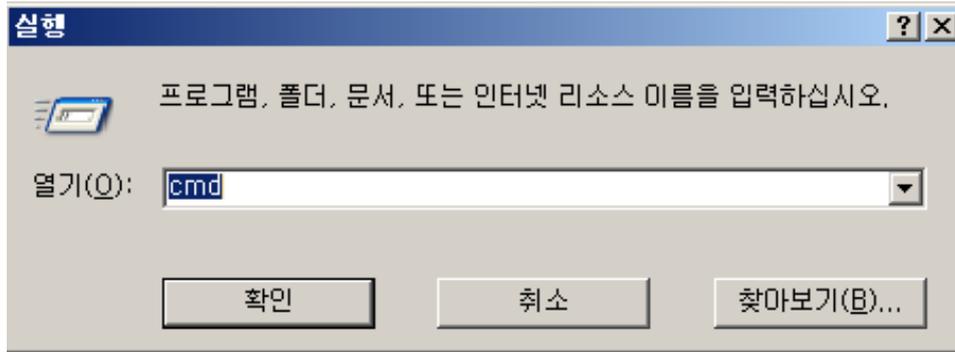


그림 9.2-5 [작업표시줄]-[시작]-[실행] 창에서 cmd 명령어 입력

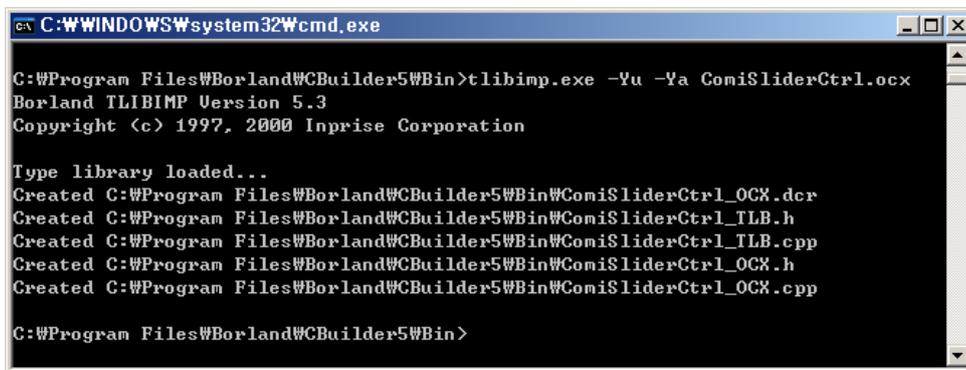


그림 9.2-6 Tlibimp.exe 파일을 이용 Import Type Library File 생성

4. Builder 5 에서 OCX 를 등록합니다.

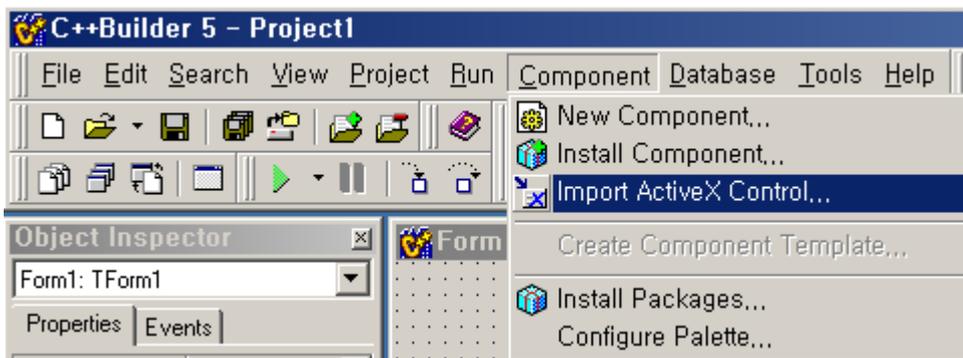


그림 9.2-7 C++ Builder 5 의 ActiveX Component 등록

5. Import ActiveX 윈도우가 화면에 나타나면, ComiSliderCtrl 을 선택하고 원하는 Palette Page 를 선택한뒤 Install 버튼을 눌러 설치를 시작합니다.

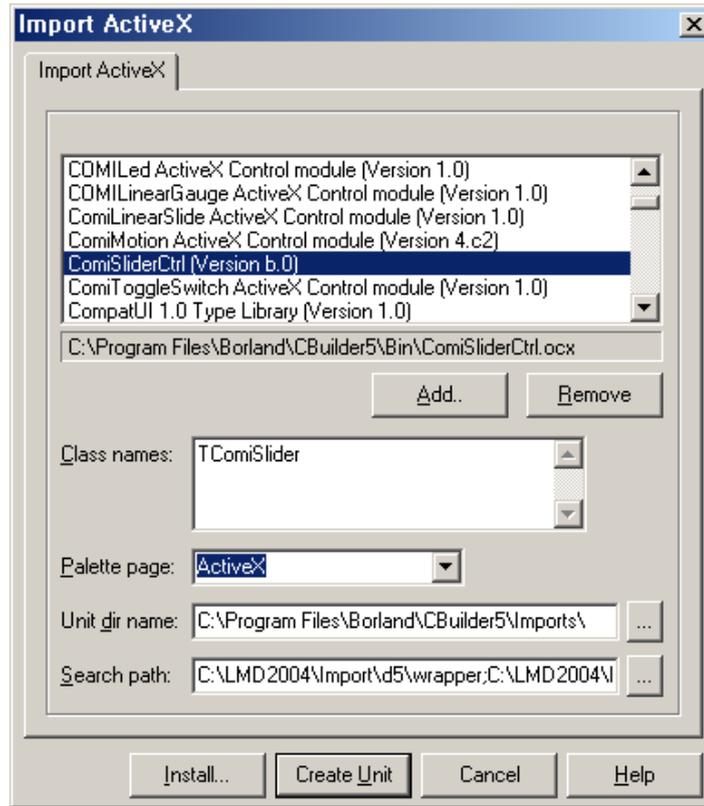


그림 9.2-8 C++ Builder 5의 ActiveX Component를 등록하기 위한 Import Active X 화면

6. Install 창이 화면에 나타나면 Into new package 탭에서 생성을 원하는 Package 파일 이름 및 설명을 입력하여 새로운 bpk를 생성합니다.

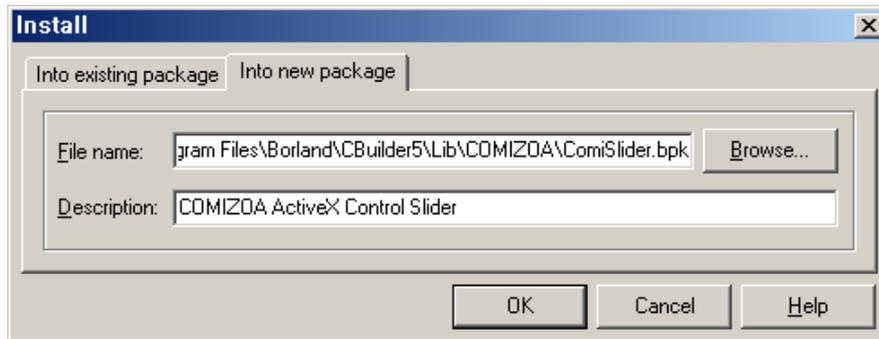


그림 9.2-9 C++ Builder 5의 ActiveX Component 설치 화면

7. Package를 Install 한다는 확인(確認) 창이 화면에 표시되면, “No”를 선택합니다.

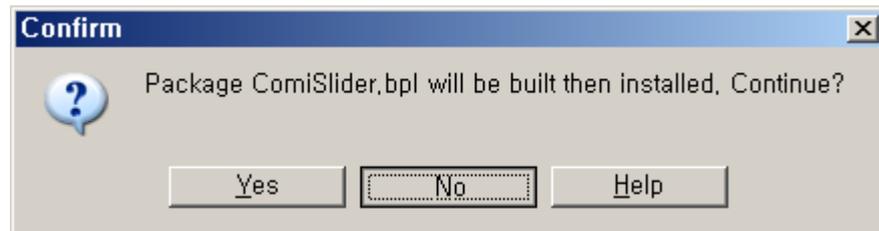


그림 9.2-10 C++ Builder 5에서 ActiveX Component 설치 화면

8. Tlibimp.exe 파일을 이용해 생성한 5 개의 파일을 ..\Borland\CBuilder5\Imports 에 덮어쓰기(Overwrite) 합니다

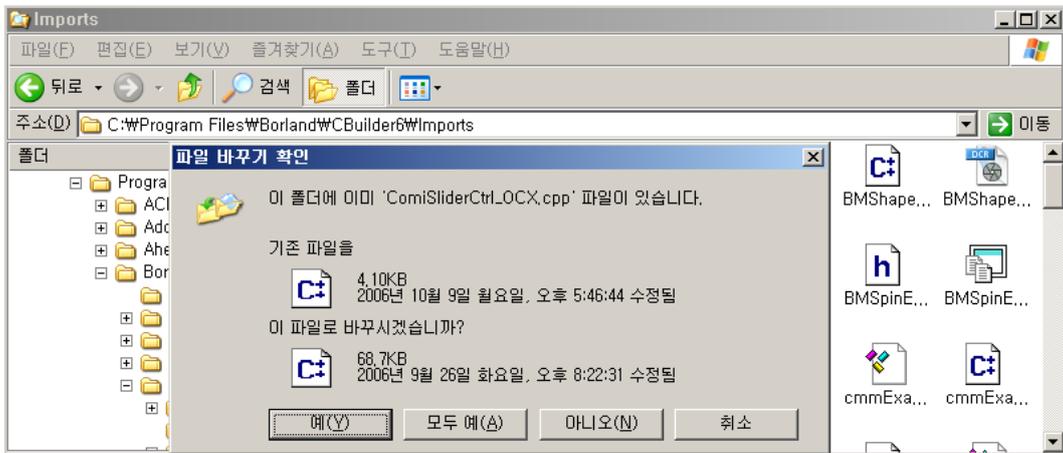


그림 9.2-11 Tlibimp.exe 가 생성한 파일을 통해 기존 파일을 대체하는 작업 화면

9. 덮어쓰기가 끝나면 생성한 Package 파일을 Install 합니다.

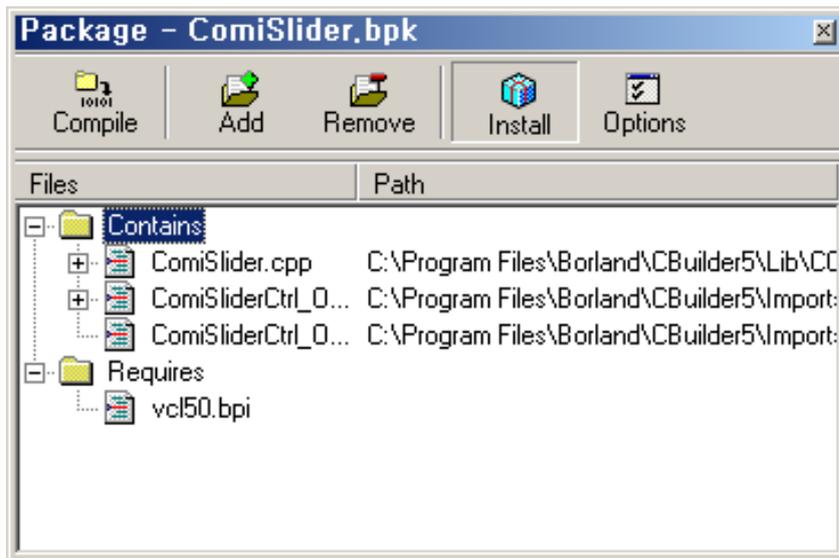


그림 9.2-12 C++ Builder 5 의 Package 표시 화면

10. Package 가 Install 되었다는 메시지와 함께 툴 팔레트에 OCX 가 정상적으로 등록된 것을 확인(確認)할 수 있습니다.



그림 9.2-13 C++ Builder 5 의 패키지 설치 완료 화면

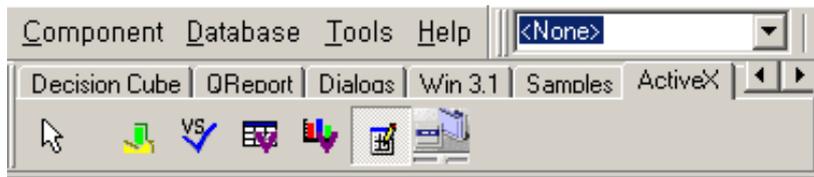


그림 9.2-14 C++ Builder 5 에서 팔레트(Palette) 에 등록된 ComiSlider Active X Control

* Builder 6 에서 ComiSliderCtrl.ocx 사용방법.

11. [Tool]-[Environment Options.]를 선택합니다..

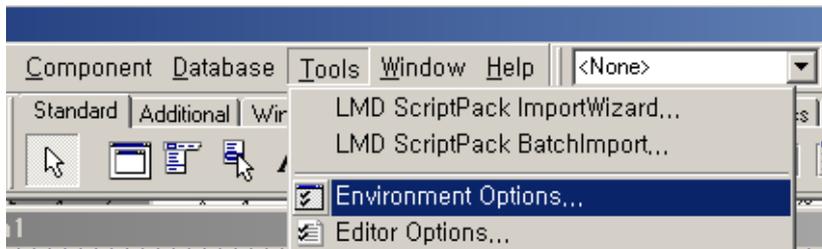


그림 9.2-15 C++ Builder 6 에서 ActiveX Component 등록 1

12. Enviroment Options 창의 Type Library 탭에서 “Ignore special CoClass Flags when importing”, “Can Create” - Check 한뒤 OK 버튼을 클릭합니다.

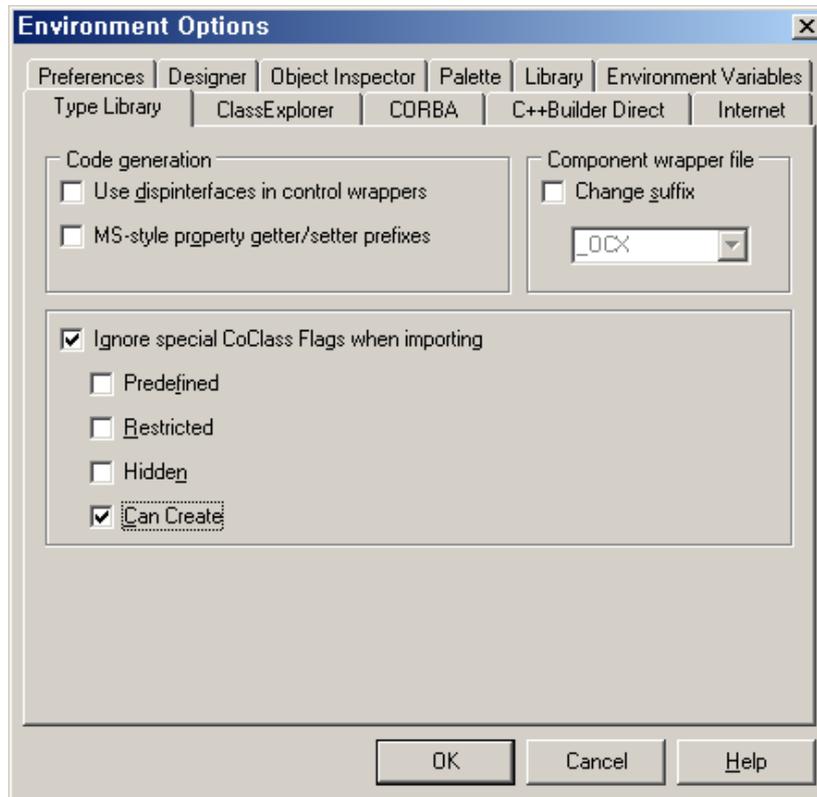


그림 9.2-16 C++ Builder 6 의 Environment Options 창의 화면

13. [Component]-[Import ActiveX Control..]를 선택한뒤 절차에 따라 OCX 를 등록하면 됩니다.

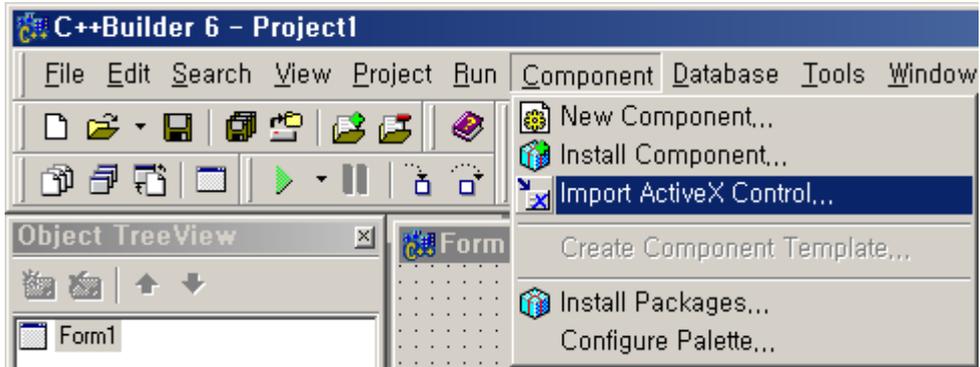
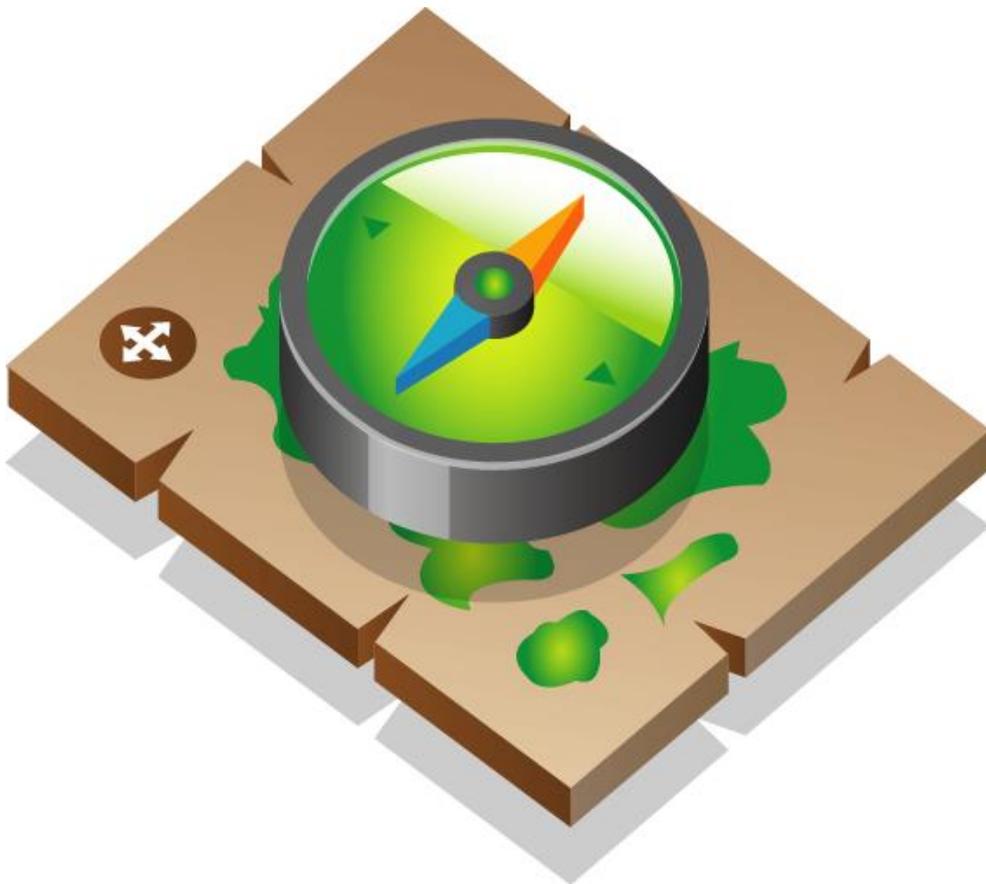


그림 9.2-17 C++ Builder 6 에서 ActiveX Component 등록 3

Index of ComisWECAT functions

필요한 함수(函數)를 가장 빠르고 쉽게 찾으십시오. ComisWECAT 매뉴얼에서는 고객(顧客)님들께서 원하시는 함수들을 일목요연하게 정리하였습니다. 필요한 함수는 온라인 문서에서 하이퍼 링크 기능으로 찾을 수 있도록 구성하였습니다.

하 이퍼 링크 기능(機能)을 통해 본장에서는 빠르고 정확하게 고객(顧客)님들께서 원하시는 함수(函數)를 찾으실 수 있도록 구성(構成)하였습니다. Adobe 社의 Acrobat Reader 와 같은 전자 문서 뷰어(Viewer) 를 통해 최단 시간내에 원하시는 함수(函數)를 찾을 수 있습니다.



III Quick Reference to ComiSWECAT Functions

<i>secDll_Load</i>	46
<i>secDll_Unload</i>	47
<i>secDll_IsLoaded</i>	48
<i>secGn_LoadDevices</i>	51
<i>secGn_UnloadDevices</i>	53
<i>secNet_ScanSlave</i>	56
<i>secSlv_GetProdInfo_A</i>	59
<i>secSlv_GetAIState_A</i>	62
<i>secdiGetOne</i>	66
<i>secdiGetMulti</i>	68
<i>secdiGetNumChannels</i>	70
<i>secdoPutOne / secdoGetOne</i>	71
<i>secdoPutMulti / secdoGetMulti</i>	73
<i>secdoGetNumChannels</i>	75

TEST & MEASUREMENT & AUTOMATION / COMIZOA

ComiSWECAT Manual

저작권자 : **㈜커미조아**

Copyright (c) by COMIZOA CO.,LTD. All right reserved.

2017년 01월 26일 1판 인쇄

매뉴얼 자료 번호 : 1.17.1.26



㈜커미조아

<http://www.comizoa.com>

Tel) 042 - 936 - 6500~6

Fax) 042 - 936 - 6507

이 사용자 설명서 상의 삽입된 삽화 및 예제 프로그램을 포함한 전체 내용은 대한민국 저작권법에 의해 보호되고 있습니다.
㈜커미조아의 사전 서면 동의 없이 사용자 설명서의 일부 또는 전체를 어떤 형태로든 복사, 전재할 수 없습니다.