

기술 지원

보다 빠른 기술지원을 위해 문의 전 아래의 내용을 참고하십시오

- 문의 전 **Online Manual – InTouch Reference Guide** 를 참조하십시오
- 현재 사용하는 InTouch 의 버전
- 사용중인 **OS** 의 버전. 예를들어 **Windows 2003 server** 또는 **Microsoft Windows XP Version (sp2)이상**
- 정확한 시스템 에러 메시지.
- **System Management console** 상에 나타난 Log 메시지.
- 문제를 해결하기 위해 취한 조치와 결과.
- 같은 문제를 다시금 접하기 위한 방법

삼미정보시스템(주) 제조사업팀

Web – fa.sammicomputer.co.kr

Tel – 02-790-5507

Fax – 02-790-5509

Address – 서울시 용산구 갈월동 103-15 서조빌딩 7 층 (우) 140-807



Module 1

소개

Section 1 – 코스 소개

Section 2 – System Requirements, Licensing and Support

Section 3 – Stand - Alone InTouch Application 소개

Section 4 – IDE - Managed InTouch Application 소개

Section 1 – 코스 소개

Module 1 – 소개

Section 1 – Course Introduction

Section 2 – System Requirements, Licensing and Support

Section 3 – Introduction to Stand-Alone InTouch Applications

: 전통적인 InTouch 개발방법과 Application Manager, WindowMaker, Window Viewer Components 사용법

Section 4 – Introduction to IDE-Managed InTouch Applications

: InTouch 와 Archestra IDE 의 관계에 관한 기본 개념소개 및 InTouch managed/ Integrated Application 을 생성하기위해 사용되는 Galaxy 와 \$InTouchViewApp template 생성

Module 2 – Development Environment

Section 1 – 소개

: Window Maker 의 개발 환경기능

Section 2 – WindowMaker 사용

Lap 1 – New Application 생성 및 그래픽 작업

Module 3 – Tagname Dictionary

Section 1 – Tag 및 Tag 의 기능

: Tag 의 생성, 정의 , 수정 및 Tagname Dictionary 사용

Lap 2 – Tag 만들기

Module 4 – 애니메이션 링크**Section 1 - 애니메이션 링크**

: 이 장에서는 **Tagname dot** 필드 와 **Tag** 브라우저에 접속하는 방법과 기본적인 애니메이션링크 기능을 정의 하고, 확인합니다.

Section 2 – Animation Objects

: 다양한 애니메이션 설정 및 옵션을 설명합니다. **Window** 위의 객체에 애니메이션을 설정 해봅니다.

Section 3 – Animation Links – Quick Reference

: 모든 애니메이션링크의 기본적인 기능을 표로 설명합니다.

Lab 3 – 애니메이션 링크 설정**Section 4 – Using SmartSymbols**

: **SmartSymbol** 객체기반의 기술로써, **InTouch** 의 그래픽을 재 사용할 수 있는 **template** 을 만들 수 있습니다. **template** 은 **Application** 에서 뿐만이 아니라 **Network** 로 연결된 **PC** 로 객체를 파생시켜 **Application** 개발이 용이하게 합니다.

그렇기 때문에 개발자는 **HMI Application** 을 생성하고, 수정하고, 확인하기 위한 시간 을 단축시킬 수 있습니다.

Lab 4 – SmartSymbol 사용하기**Section 5 - ArchestrA Symbol 소개****Lab 5 – ArchestrA Symbol 사용하기****Module 5 – InTouch QuickScripts****Section 1 – QuickScript Types**

: 이 장에서는 **InTouch QuickScript** 종류에 대해 설명합니다.

Section 2 – Script 작성하기

: **QuickScript** 편집기의 구성과 기능에 대해 설명합니다.

Section 3 – QuickScripts – Quick Reference

: 각 Script 종류별로 기본적이 기능을 요약 설명합니다.

Lab 6 – QuickScript 사용하기

Module 6 – Alarms**Section 1 – Alarm and Events**

: 이 장에서는 알람과 이벤트의 종류를 설명하고 Application에서 알람을 설정하는 방법을 살펴 봅니다.

Section 2 – 분산알람객체

: 분산 알람 객체에 로컬 혹은 원격알람을 표시합니다. 알람디스플레이 객체의 속성 및 기능을 살펴봅니다.

Lab 7- 알람과 이벤트 생성하기

Module 7 – Real- time and Historical Trending**Section 1 – Real –time Trends**

: InTouch는 실시간 Trend (Real-time), 이력트랜드 (Historical)의 두 가지 Trend type을 제공합니다. 각각의 Trend 객체는 여러 개의 Tag를 Display하도록 구성할 수 있습니다.

Lab 8 – 실시간 Trend 구성하기

Section 2- Historical Trends

Lab 9 – 이력 Trend 구성하기

Module 8 – I/O Communication**Section 1 – I/O Communication Protocols**

: I/O 통신 프로토콜을 설명하고, InTouch Access name 을 어떻게 구성하는지 살펴봅니다.

Section 2 – Wonderware I/O Servers

: Wonderware I/O Server (Modbus) 의 구성을 살펴봅니다.

Lab 10 – I/O 통신구성하기

Section 3 – Troubleshooting I/O

: 이 장은 I/O Server 와 InTouch 또는 I/O Server 와 PLC 사이의 통신 문제 해결 방법을 설명합니다.

Module 9 – Advanced Tag Functionality**Section 1 – I/O 관련 Function**

이 장은 IOSetName, IOSetItem 함수 및 Dynamic Reference Addressing 와 같은 고급 I/O function 에 대해 살펴봅니다.

Section 2 – Indirect Tag

Indirect tag 의 개념과 사용에 대해 설명합니다.

Lab 11 – Indirect tag 생성 및 사용

Section 3 – InTouch SuperTags

SuperTag 템플린구조의 사용과 정의를 살펴봅니다.

Lab 12 – SuperTag 정의

Section 4 – DBDump and DBLoad

DBDump 및 DBLoad Utility 의 사용에 대해 살펴봅니다.

Section 5 – Tagname Dictionary 의 활용

Tag Converting , Deleting, substituting 등 Tag 를 편집하는 방법과 Cross Reference Utility 사용에 대해 살펴봅니다.

Lab 13 – Tag 삭제**Module 10 – Security****Section 1 – 보안 모델**

InTouch 의 3 가지 보안 모델에 대해 살펴봅니다.

Lab 14 – InTouch Model 보안설정 하기**Section 2 – Application 보안**

InTouch Runtime 과 개발 환경에서 **Application** 을 보호하기 위한 방법을 살펴보고, 또한 **Application** 을 자동으로 시작하도록 설정하는 방법을

살펴봅니다.

Module 11 – Application Backup**Section 1 – Application Backup**

이장에서는 분산개발 환경을 활용하고 **Application** 을 어떻게 저장하는지 설명합니다.

Section 2 – System Requirements, Licensing and Support

이 단락에서는 InTouch 10.0 을 설치 하기위한 Hardware 및 software 의 사양을 나타냅니다. 이 정보의 Update 사항을 확인을 위해서는 Wonderware Technical Support website 를 참조하시기 바랍니다. www.wonderware.com/support

System Requirements for InTouch HMI

Hardware Recommendations

다음 목록은 InTouch 10.0 을 설치하기 위한 하드웨어 사양입니다.

InTouch HMI Stand Alond Hardware

- | 1.2GHz 이상의 PC
- | 최소 512 MB 의 RAM, 1GB 이상 권장
- | 최소 4GB 의 하드디스크 용량
- | 최소 1024 X 768 이상의 지원하는 VGA 와 모니터
- | CD-ROM 또는 DVD driver (프로그램 설치시)

InTouch HMI and the ArchestrA IDE Hardware

- | 2GHz 이상의 PC (Dual core Processor 권장)
- | 2GB RAM
- | 최소 4GB 의 하드디스크 용량
- | 최소 1024 X 768 이상의 지원하는 VGA 와 모니터
- | CD-ROM 또는 DVD driver (프로그램 설치시)

Note: Microsoft Windows Vista OS 는 InTouch 10.0 의 최소사양 이상을 필요로 합니다. InTouch 10.0 을 Windows Vista 에서 사용 할 경우, 아래의 Web site 에서 사양을 확인하십시오.

www.microsoft.com/windows/products/windowsvista/editions/systemrequirements.mspx

Software Requirements for InTouch HMI v10.0

이 단락은 InTouch 10.0 의 설치를 위한 Operating system 과 software 요구 사양을 보여줍니다.

Operating Systems

다음의 테이블 목록은 InTouch Server 와 Client Components 를 실행 시키기 위해 설치될 수 있는 Operating System 지원테이블입니다.

	InTouch HMI Components				
Operating System	Window Maker	Window View	ArchestrA IDE	ArchestrA Run Time	Galaxy Repository
Windows Vista Enterprise	○	○	○	○	
Windows Vista Business	○	○	○	○	
Windows Vista Ultimate	○	○	○	○	
Windows Server 2003 Standard Sp2	○	○	○	○	○
Windows Server 2003 Enterprise Sp2	○	○	○	○	○
Windows Server 2003 Standard R2 Sp2	○	○	○	○	○
Windows Server 2003 Enterprise R2 Sp2	○	○	○	○	○
Windows XP Professional SP2	○	○	○	○	
Windows XP Tablet 2005	○	○		○	

Note: InTouch HMI version 10.0 은 Microsoft Windows 2000 을 지원하지 않습니다. 또한, Microsoft Windows Server 2003 SP2 는 InTouch Server Component 를 실행 하기 위한 OS 로써, Widows XP Pro SP2 는 InTouch Client Components 를 실행 하기 위한 OS 로써 권장 합니다.

Section 3 – Introduction to Stand-Alone InTouch Application

이 장에서는 전통적인 InTouch 사용방법과 Application Manager, WindowMaker, 그리고 WindowViewer Component의 사용방법을 자세히 살펴봅니다.

소개

Wonderware 의 InTouch 는 Microsoft Windows 2000, Windows XP, Windows Vista Operating System 에서 빠르고 쉽게 HMI Application 을 개발 할 수 있도록 합니다.

또한 ActiveX Controls, OLE, 그래픽, 네트워크를 포함한 Microsoft Windows 의 핵심 기능들을 사용하여 Application 을 개발 할 수 있습니다

InTouch 는 전 세계에 식품 가공, 반도체, 정유 및 천연가스, 자동차, 화공약품, 제약, 펄프 및 제지, 교통 및 유트리티 등의 다양한 분야에서 사용되고 있습니다.

InTouch Components

InTouch 는 세가지 주요 Component 로 구성되어 있습니다 :

- | Application Manager, WindowMaker and WindowViewer.

Note: InTouch HMI version 10.0 은 Microsoft Windows 2000 을 지원하지 않습니다.



Application Manager : 생성된 Application 을 구성합니다. Local node 에 존재하는 Application 의 목록을 보여줍니다. 또한 Name, Path, Resolution, Version, Application Type, Application Version , Data Modified, Description 등의 Application 정보를 나타내줍니다.

Application Manager 에서는 WindowViewer 를 NT Service 로 구성할 수 있으며, Servier – Client 구조의 Network Application Development (NAD) 및 Dynamic Resolution Conversion (DRC)을 구성 할 수 있습니다.

또한 DBDump , DBLoad Utility 가 포함 되어있습니다.



WindowMaker : InTouch 개발 환경으로써 객체 기반의 그래픽은 애니메이션기능 , display Window 들을 개발하는데 사용됩니다. 이 Display Window 는 Plant I/O System , Window 기반의 Application 들에 연결될 수 있습니다.



WindowViewer : WindowMaker에서 개발된 Application의 실행환경입니다. WindowViewer는 InTouch QuickScripts, 이력데이터 저장 및 레포트, 프로세스 알람 저장, I/O 통신을 수행합니다.

Application Manager Tools

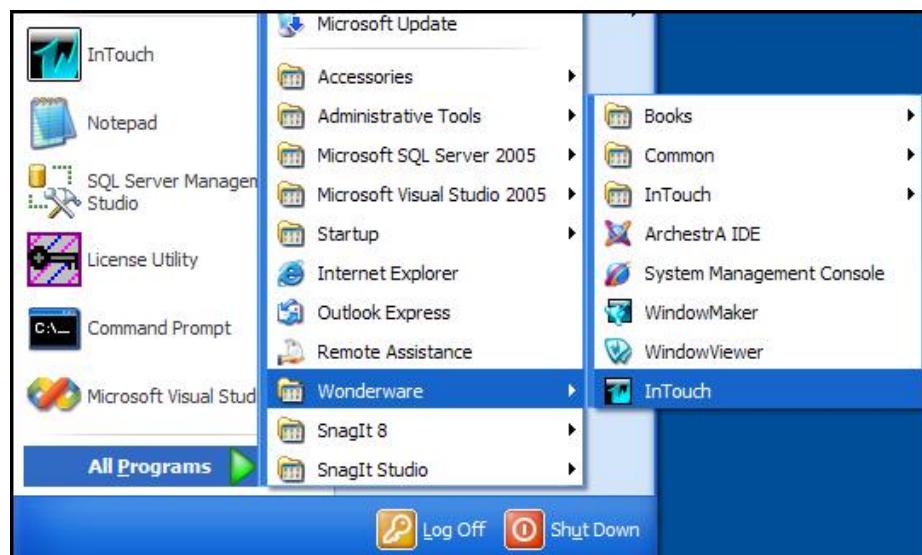
Button	Description
	새로운 Application을 생성하기 위해 File 메뉴에 New 명령을 실행합니다.
	선택된 Application을 WindowMaker로 실행시키기 위해 File 메뉴에 Window Maker 명령을 실행합니다.
	선택된 Application을 WindowViewer로 실행시키기 위해 File 메뉴에 Window Viewer 명령을 실행합니다.
	DBLoad Utility를 실행하기 위해 File 메뉴의 DBLoad를 실행합니다. 이 Utility는 file 안에 Tagname Dictionary를 Load하기 위해 사용됩니다.
	DBLoad Utilit를 실행하기 위해 File 메뉴의 DBDump를 실행합니다. 이 Utility는 Application의 Tagname Dictionary를 csv 파일로 빼내기 위해 사용합니다.

	선택된 Application 을 Application Manager 에서 삭제합니다.
	View 메뉴의 내용을 drop-down 리스트메뉴로 보여줍니다.
	Node Properties 창을 열수 있습니다. 이 곳에서 Network Application Development (NAD), Dynamic Resolution Conversion (DRC) 설정을 할 수 있습니다.
	System 안에 InTouch Application 을 찾아 Application Manager 에 등록 할 수 있도록 Tool 메뉴의 Find Application 명령을 실행 시킵니다.
	Galaxy 에 접속하기 위해 ArchestrA IDE 를 실행 시킵니다.

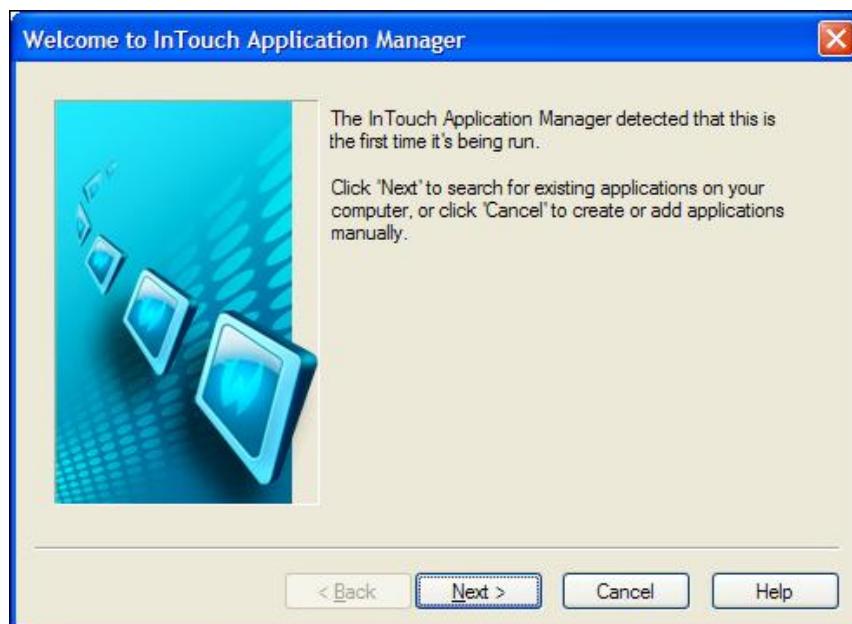
Runing InTouch for the First Time

InTouch.exe 를 최초 실행시키면, InTOUCH.INI file 이 자동으로 생성됩니다. 이 파일은 Application 폴더 안에 위치하며, Application 의 기본적이 구성설정이 포함되어있습니다. Application 을 구성하기 위한 설정은 INTOUCH.INI 파일이 쓰여집니다.

- a. 시작/프로그램/Wonderware/InTouch 를 선택합니다.



Welcome to InTouch Application Manager 창이 나타납니다.



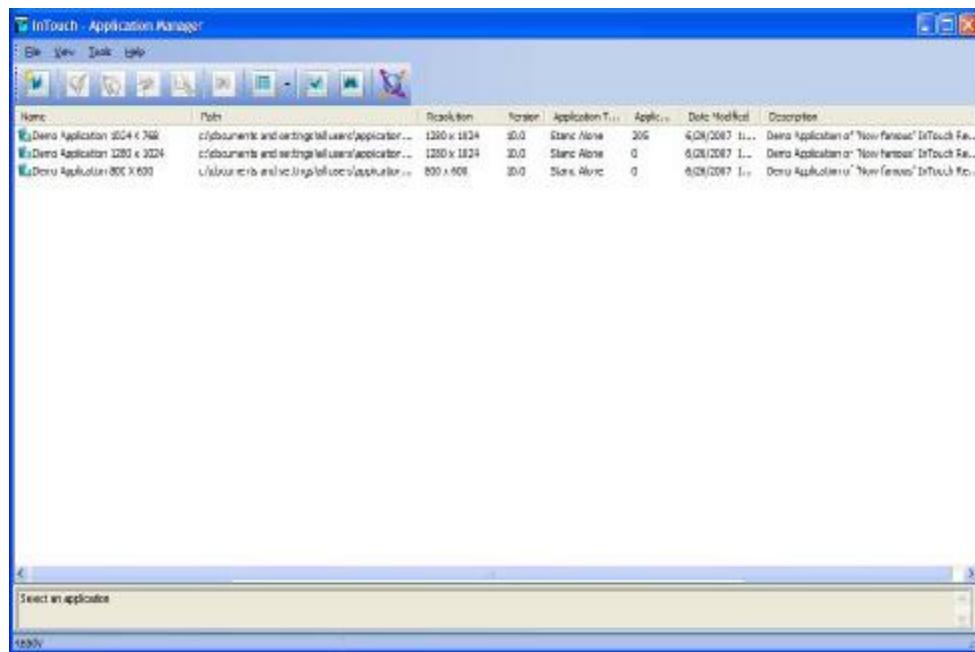
- b. Next 를 클릭합니다.
- c. Application 을 만들 디렉토리를 지정하기 위해 입력 창에 경로를 입력합니다.
Browse 를 클릭합니다.



- d. Finish 를 클릭합니다.

Application Manager 는 local PC 에 존재하는 InTouch Application 을 찾습니다.
Application 링크는 Application name 을 포함한 아이콘으로 표시되며, Application 정보도 표시 됩니다.

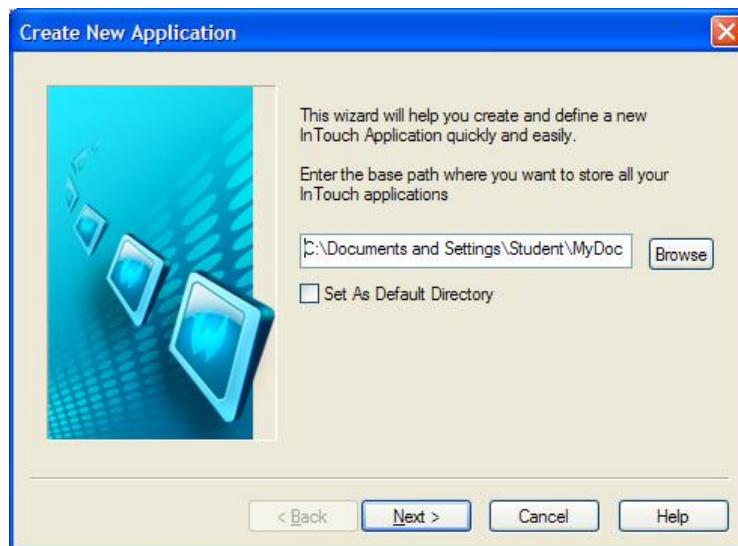
예를 들면 다음 그림과 같습니다 :



Creating a New Application

To create a new application

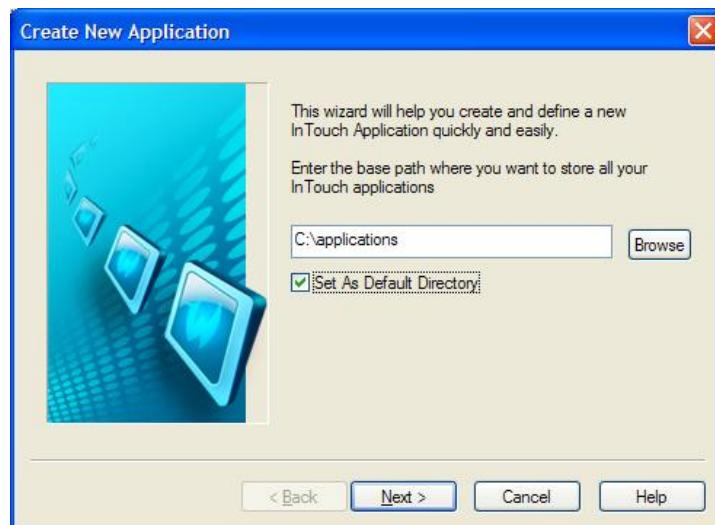
- a. Application Manager 도구에서 New 버튼 을 클릭합니다.
Create New Application 창이 나타납니다.



기본적으로, 시스템은 C:\Documents and Settings\Username\My Documents\My InTouch Application\ Path 를 나타냅니다. Application 을 새로 만들 디렉토리 경로를 입력합니다.

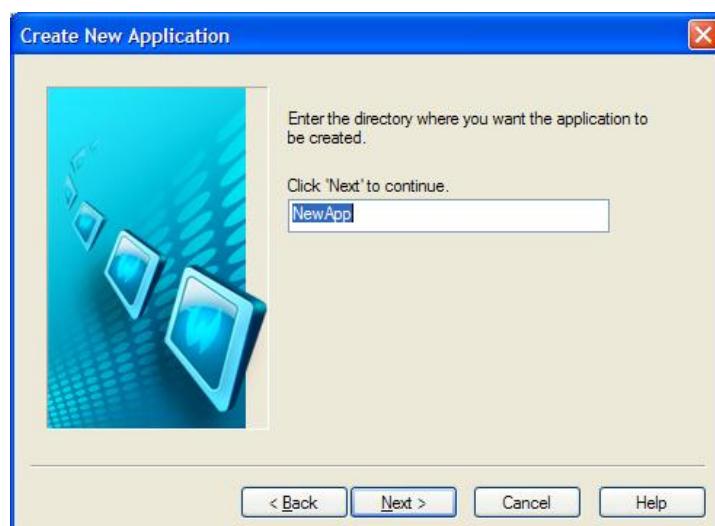
디렉토리의 위치를 **Browse** 에서 클릭합니다.

- b. 여기에서는 C:\application 디렉토리를 사용합니다.
- c. Set As Default Directory checkbox 를 클릭합니다. 모든 Application 은 지정된 directory 안에 생성됩니다.

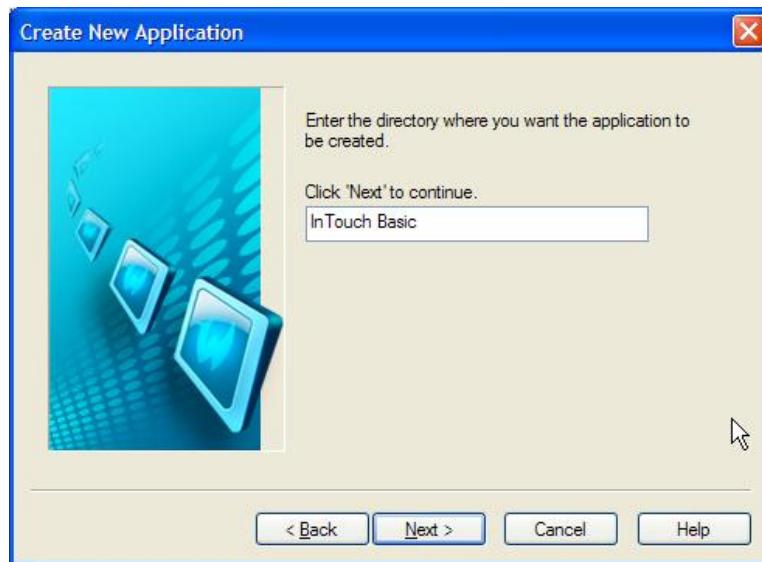


- d. Next 를 클릭합니다.

Create New Application 창이 나타납니다. 이 창은 Applicaton 이 생성될 폴더를 정의할 수 있습니다.

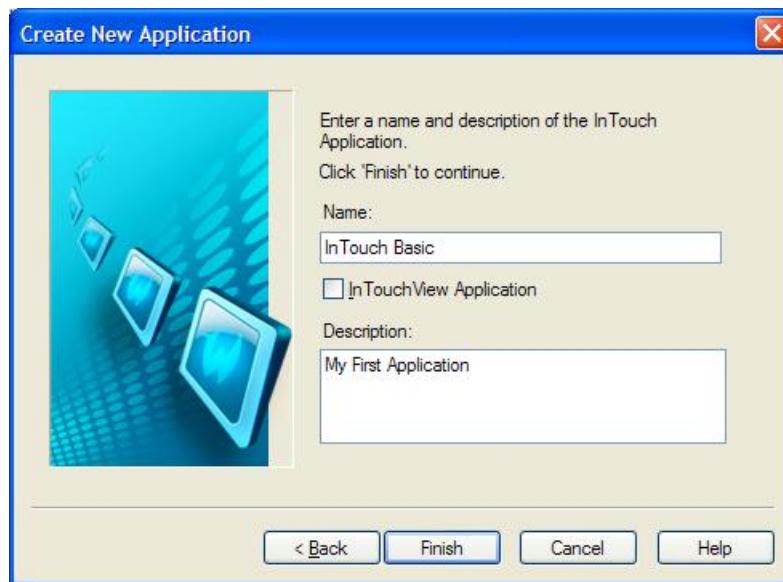


e. 생성 할 폴더 이름을 입력합니다.



f. Next 를 클릭합니다.

g. 새로운 Application 의 이름을 입력합니다. 이 이름은 Application Manager에 리스트 될 때 보여집니다.

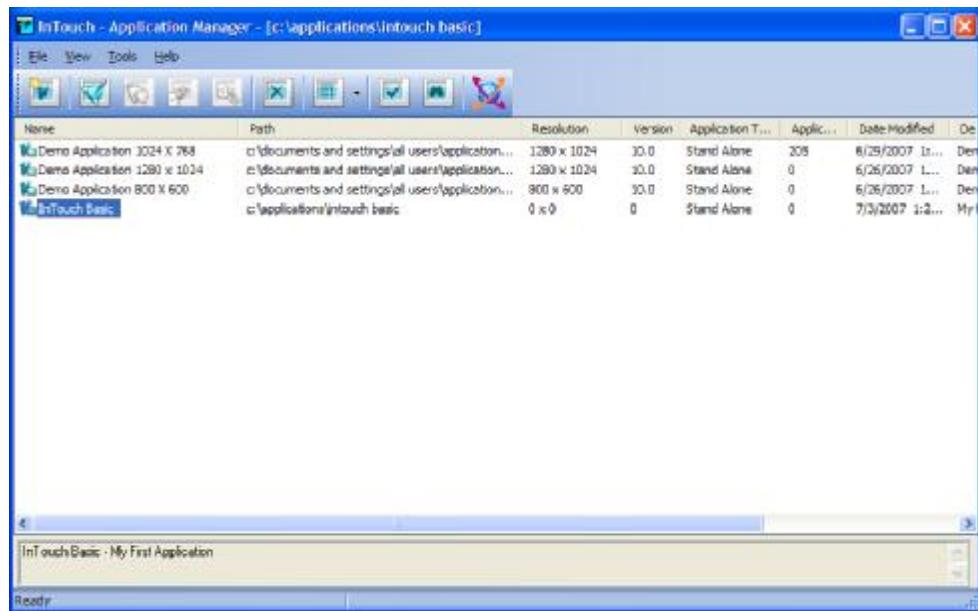


- h. Description box 에 Application 의 설명을 입력합니다.

Note: InTouchView Application 옵션은 선택하지 않습니다. 이것은 ArchestrA 환경에서 Industrial Application Server 와 함께 사용됩니다.

- i. Finish 를 클릭합니다.

InTouch Application Manager 에 새로 만든 Application 의 icon, path, version, Mode, Description 이 나타납니다.



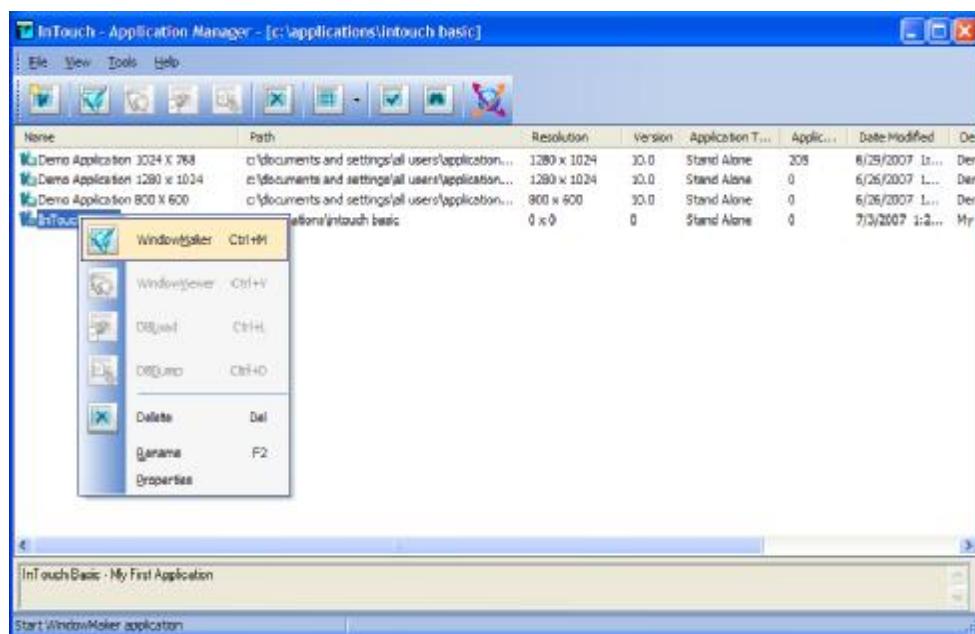
Note: Application 의 Application Manager 에 처음 디스플레이 될 때, 해상도와 버전은 0 으로 표시됩니다. 해상도와 버전은 Application 이 열린 후 update 됩니다.

Using the InTouch Application Manager

InTouch Application Manager에서는 Application 을 생성, open, rename, delete 또는 DBDump, DBLoad Utility 를 사용 할 수 있습니다.**To open an application**

- a. InTouch Application 을 열기 위해, 아래의 방법 중에 선택하여 사용합니다.

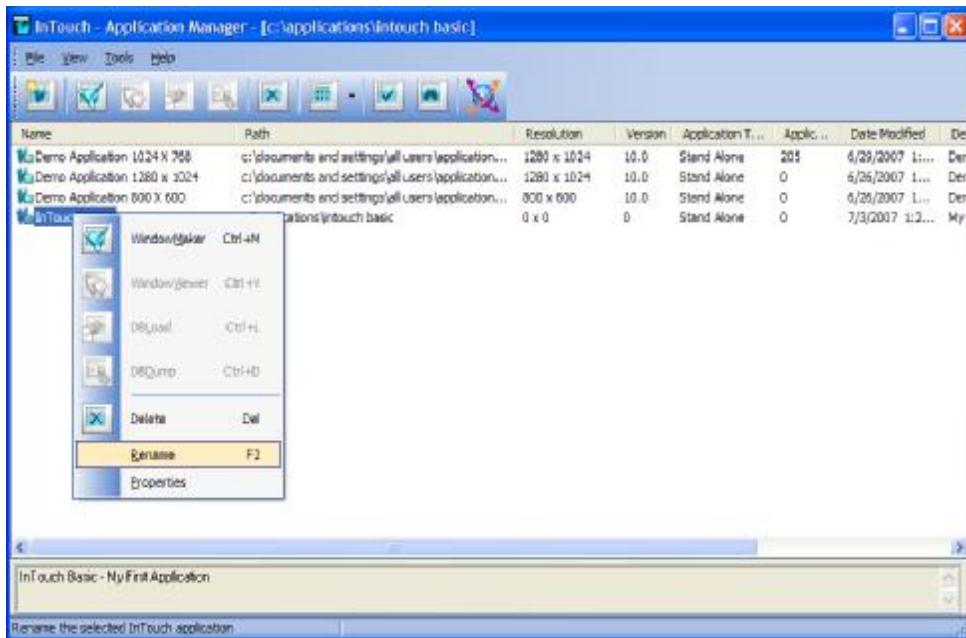
- | Application 을 선택한 후 , tool bar 에서 WindowMaker 아이콘  을 선택합니다.
- | Application 을 선택한 후, 더블클릭 합니다.
- | Application 을 선택한 후, Enter 를 누릅니다.
- | Application 에서 오른쪽 마우스 클릭 후 WindowMaker 를 클릭합니다.
(최초 생성된 Application 은 WindowViewer 가 활성화 되지 않습니다.)



WindowMaker ↳ WindowViewer 를 열며, Application Manager 는 닫힙니다.

To rename an Application

- a. 리스트에서 Application 을 선택하고, 오른쪽 마우스 클릭 후 Rename 을 선택합니다.

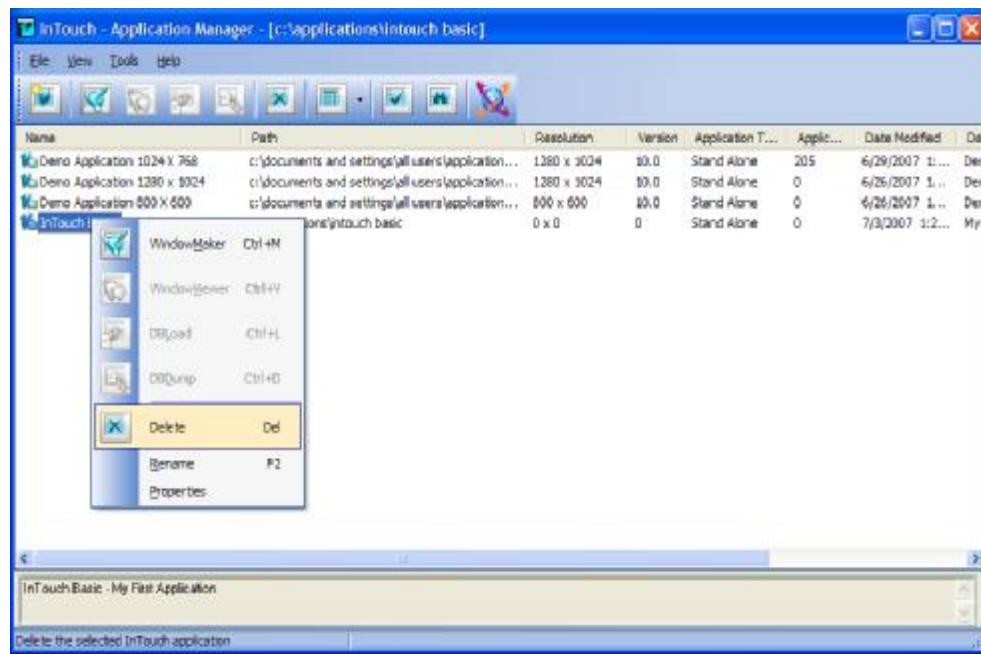


- b. 새로운 name 을 입력한 후, Enter 키를 누릅니다.

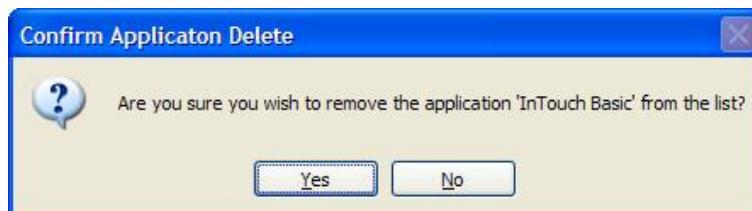
To delete an application link

Application Manage 에는 Local system 에 위치해 있는 Application link 가 표시됩니다. 리스트로부터 Application link 를 지운다 해도, 실제 InTouch Application file 은 디렉토리에서 삭제되지 않습니다.

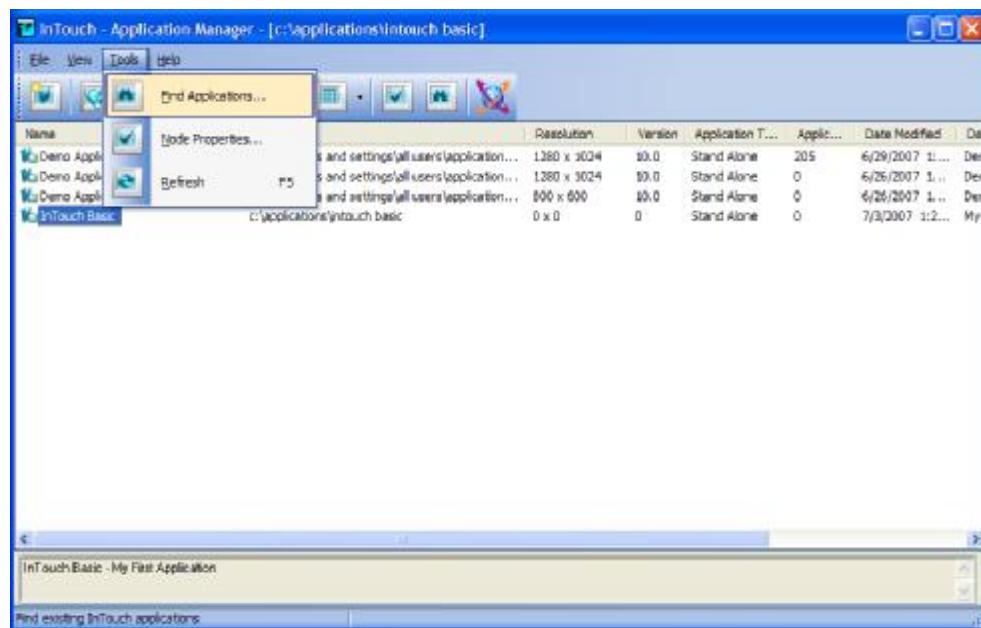
- a. 삭제 할 Application 을 선택한 후, 오른쪽 마우스 클릭을 합니다.



- b. 삭제 확인 창이 나타납니다. Yes 또는 No 를 클릭합니다.



- c. Application Manager 리스트로부터 삭제된 Application 을 다시 링크하기 위해서는 Tools/ Find Application 을 클릭합니다.



Browse for Folder 창이 나타납니다.

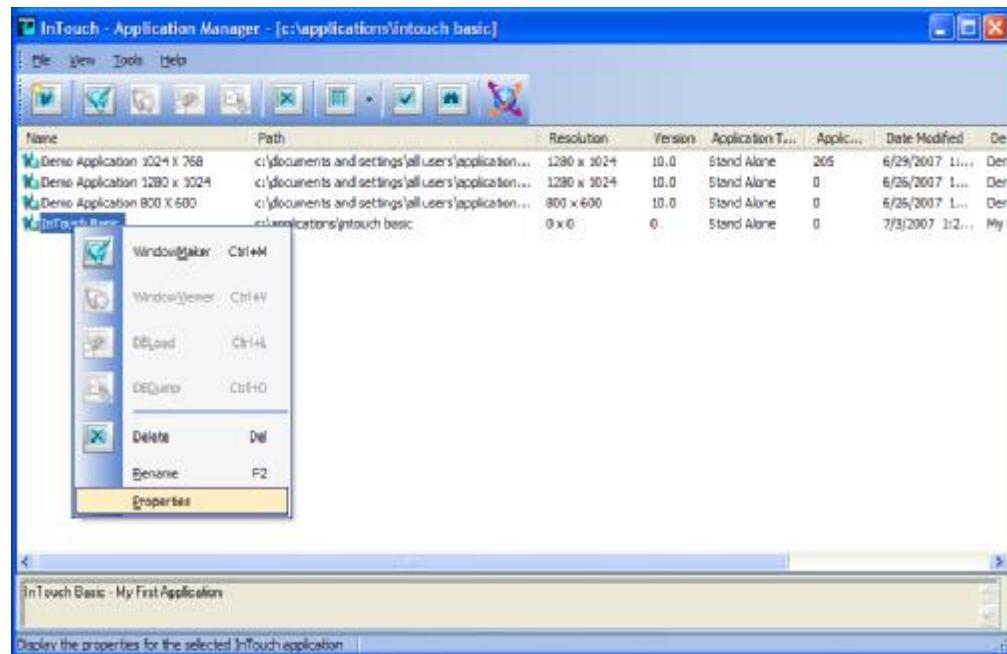


d. Application 위치를 찾아서 선택하고 OK를 클릭합니다.

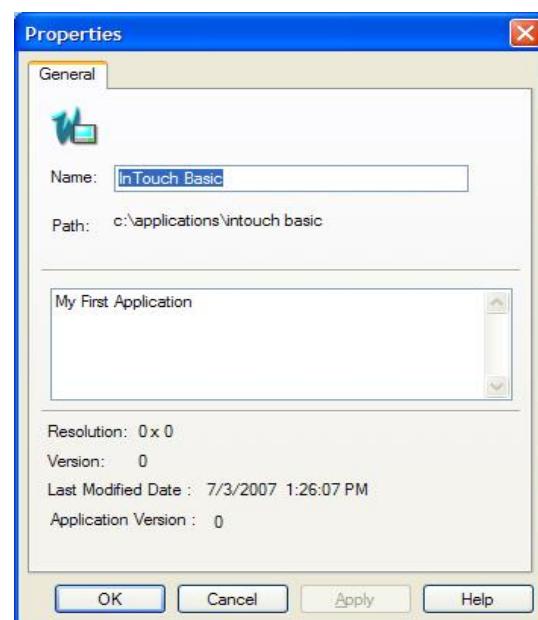
선택된 디렉토리 안에 있는 모든 Application이 Application Manager에 표시될 것입니다.

View an Application's Properties

- a. 리스트 안에 Application 을 선택합니다.
- b. 오른쪽 마우스 클릭 후, Properties 를 선택합니다.



Properties 창이 나타납니다.



Section 4 – Introduction to IDE-Managed InTouch Applications

이 장에서는, ArchestrA IDE 의 기본 개념과 InTouch Application 을 관리하고 통합하는 Galaxies 와 \$InTouchViewApp template 의 생성을 포함한, InTouch 와의 관계를 알아봅니다.

IDE 로 관리되는 InTouch Application

Wonderware System Platform 는 공장자동화와 정보관리에 필요한 SCADA, HMI.설비 및 생산관리를 하나의 플랫폼으로 제공합니다. **System Platform** 은 가상적인 플랫폼 위에 존재하는 모든 시스템에게 공통적이고 전략적인 산업 Application Service 들을 제공합니다. 또한 이는 산업표준을 기본으로 하는 **ArchestrA real-time SOA** 기술을 기반으로 합니다.

이 장에서는 **ArchestrA IDE** 에 어떻게 접속하는지, Application 에 정의한 플랫폼, engines, 템플릿, instance, 속성값들을 저장하기 위한 **Galaxy database** 를 생성하는 방법에 대해 자세히 살펴봅니다. 또한 새로운 InTouch Application 만들고, ArchestrA IDE 로부터 InTouch 를 실행하는 방법 및 InTouchViewApp template 을 사용하는 방법을 단계별로 제공합니다.

소개

개발자는 InTouch Application Manager 또는 ArchestrA IDE 를 사용하여, InTouch Application 을 관리할 수 있습니다.

- | Application 을 InTouch Application Manager 에서 개발하는 것을 Stand Alone InTouch InTouch Application 방식이라 합니다.
- | Application 을 ArchestrA IDE 에서 개발하는 것을 Managed InTouch Application 방식이라 합니다.

ArchestrA Integrated Development Environment (IDE)를 사용하여 개발자는 InTouch application 을 관리할 수 있습니다.

개발자는 하나의 Galaxy database 에 접속하여 Application 을 개발하고, 그것을 하나이상의 node 에 배치하여 실행시킬 수 있습니다.

개발자는 ArchestrA IDE 의 System platform 기능을 다음 사항에서 사용할 수 있습니다.

- | InTouch Application 이 어느 node 에서 실행되고 있는지 볼 수 있습니다.
- | InTouch Application 을 하나의 Database(repository) 에서 사용할 수 있습니다.
- | 원격 node 에서 실행되고 있는 WindowView 의 변경사항을 배포할 수 있습니다.
- | InTouch Application 에 ArchestrA Symbol 을 사용할 수 있습니다.

Stand Alone 과 IDE – Managed InTouch Application 의 비교

	Stand Alone InTouch Applications	Managed InTouch Applications
Create Application	Application Manager	ArchestrA IDE - 새로운 Application - Stand alone Application import - SmartSymbol import
Edit Application	Application Manager에서 WindowMaker 실행	IDE에서 WindowMaker 실행
Delete Application	Application Manager로부터 삭제한 후 폴더 삭제	InTouchViewApp template 삭제
ArchestrA Symbols 지원	No	Yes
DBDump, DBLoad 지원	Yes (Application Manager)	No
해상도 변환	Yes	No
분산 Application 관리	Network Application Development (NAD)	ArchestrA IDE
새로운 Application 버전을 구성하는 방법	Application Manager에서 구성 (NAD)	WindowMaker에서 구성
Test Application Fast-Switch 사용	Yes	Yes

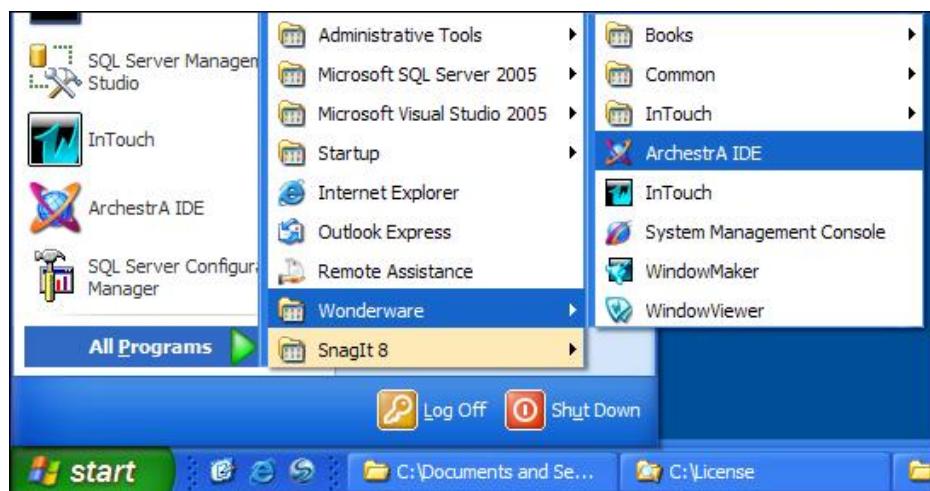
Tag 값과 Parameter 의 retentive 기능 사용	Yes	Yes
---------------------------------------	-----	-----

InTouch Managed Application 생성하기

ArchesrA IDE 실행하기

ArchesrA IDE 안에서 Managed InTouch Application 을 생성할 수 있습니다.
ArchesrA IDE 를 실행하기 위해서

- 시작/ 모든프로그램/ Wonderware/ ArchesrA IDE 를 선택합니다.



Note: 개발자는 바탕화면에 ArchesrA IDE 아이콘 을 더블클릭 함으로써 IDE를 실행할 수 있습니다.

Connect to Galaxy 창이 나타납니다.



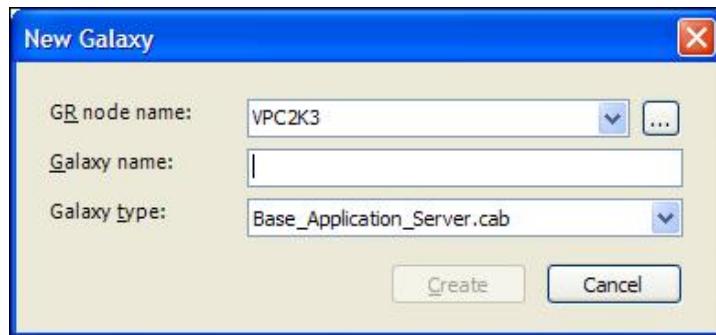
Note: Galaxy 는 Application 에 정의 되어있는 Platforms, engines, template, instance, 속성 등을 저장 합니다.

Galaxy 새로 만들기

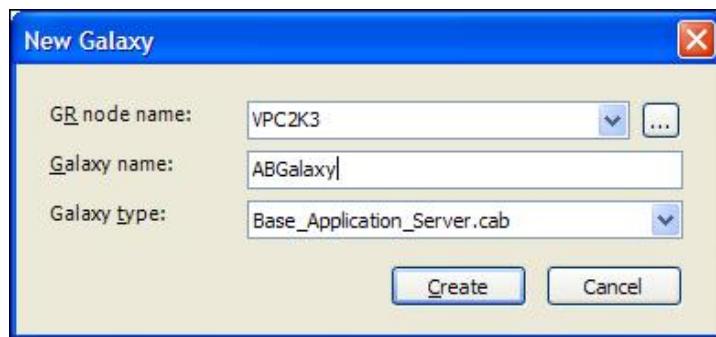
- New Galaxy 버튼을 클릭합니다.



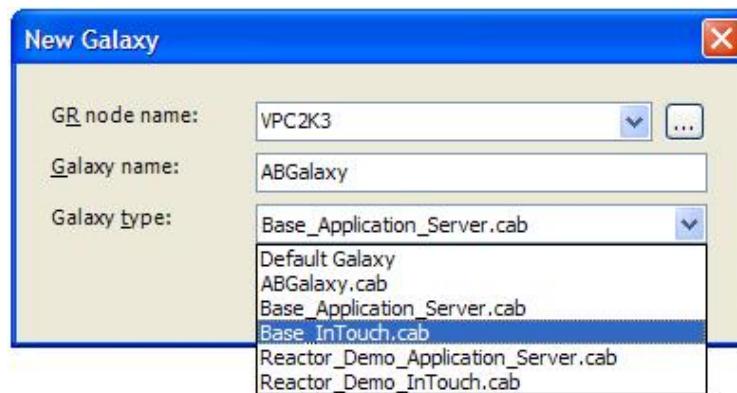
New Galaxy 창이 나타납니다.



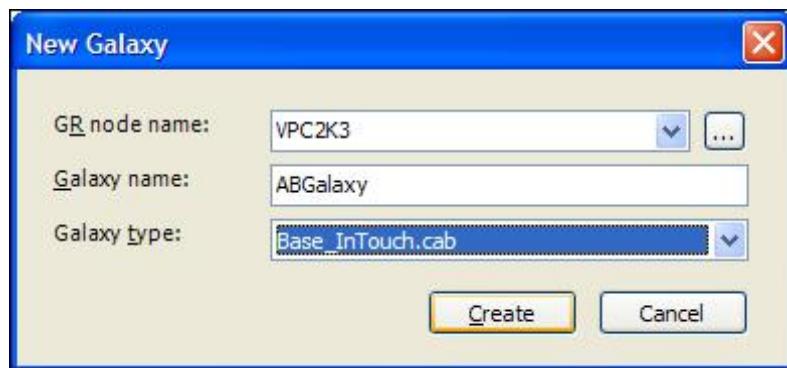
- b. Galaxy name에 생성할 Galaxy 이름을 입력합니다.



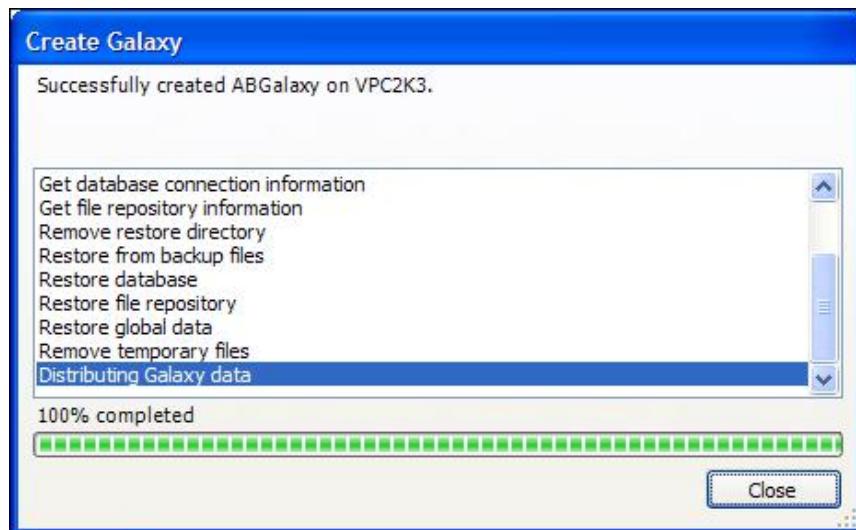
- c. Galaxy Type에서 Base_InTouch.cab 옵션을 선택합니다.



Create 를 클릭합니다.



ArchestrA IDE 는 ABGalaxy database 를 생성하기 위해 필요로 되는 Component 들의 정보와 위치를 수집합니다.



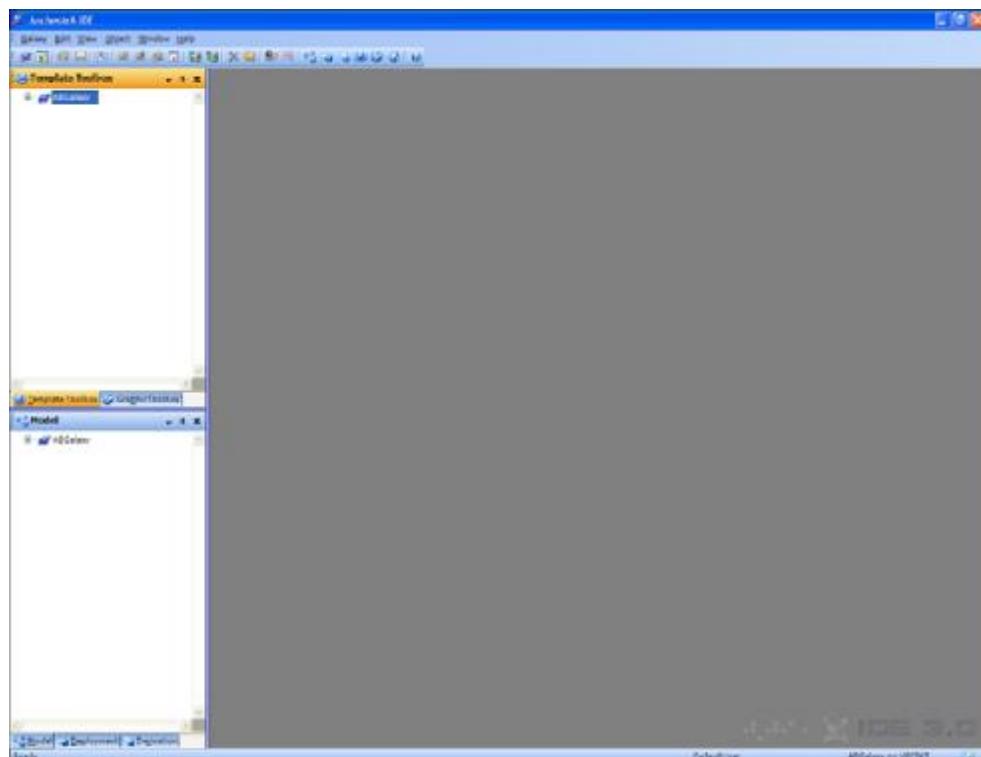
- d. Galaxy 생성이 완료되면 Close 를 클릭하고 Connect to Galaxy 창으로 돌아 옵니다.

Galaxy 연결하기

- Connect to Galaxy 윈도우에서 Connect 버튼을 클릭합니다.

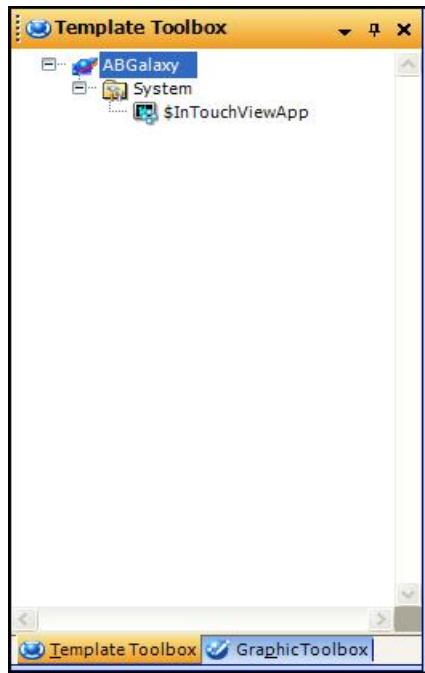


ArchesrA IDE 창이 열립니다. 새로 생성된 Galaxy 가 표시됩니다.

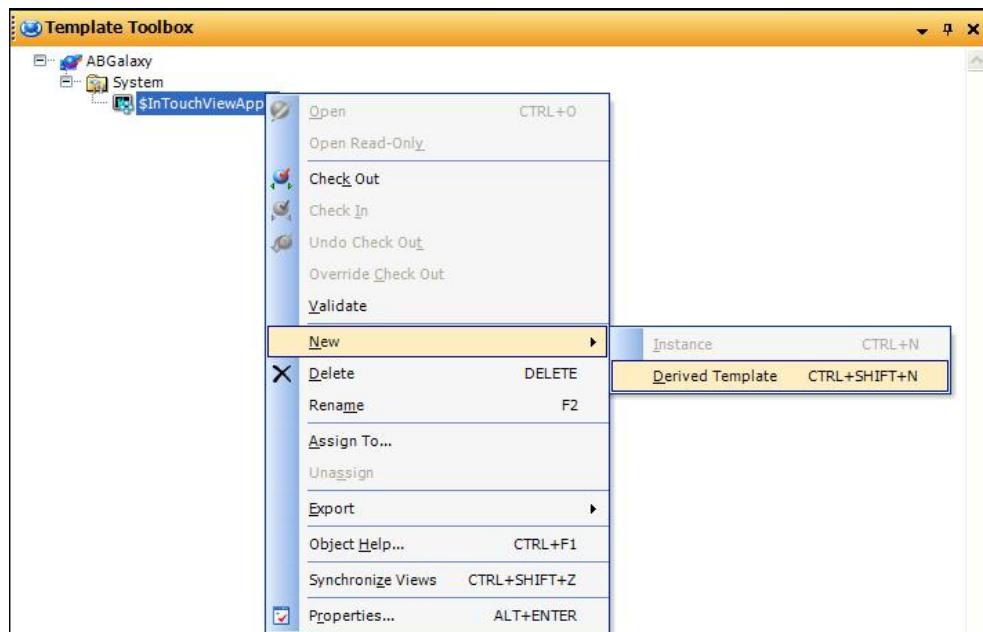


InTouch Derived Template 만들기

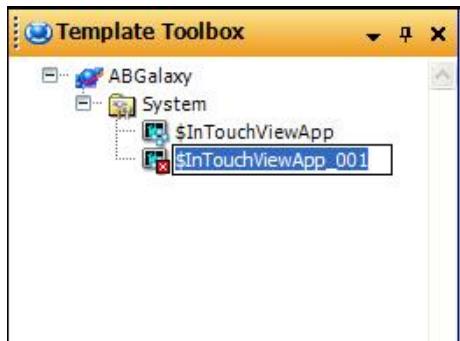
- a. Template Toolbox에서 System toolset을 확장합니다.



- b. \$InTouchViewApp 기본 템플릿에 오른쪽 마우스를 클릭하고 New/ DerivedTemplate 옵션을 선택합니다.



Template Toolbox 의 System folder 에 나타난, \$InTouchViewApp_001 에 이름을 입력합니다.



c. \$InTouchBasic 라고 명명합니다.



Note: InTouchView derived template 을 명명할 때 다음 사항을 고려합니다.

- Template 의 이름은 항상 \$로 시작합니다.
- Template 의 이름은 32 자를 넘을 수 없습니다.
- Template 의 이름은 Spaces 를 포함 할 수 없습니다.
- 이름의 두번째 문자는 \$를 사용할 수 없습니다.
- Template 의 이름은 적어도 한 개의 문자 이상이어야 합니다.

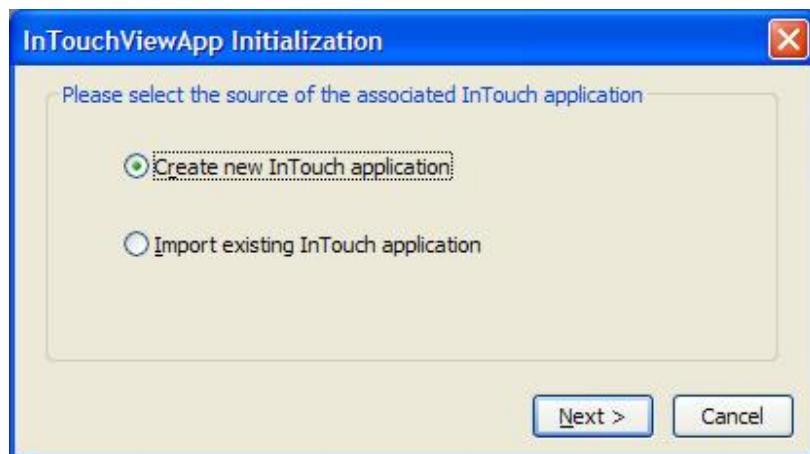
InTouch Managed Application 은 생성하기 위해서는 InTouchViewApp 객체를 사용합니다. 이 Application directory 공유 폴더로써 만들어집니다.

[\GRNodeName\GalaxyName-\\$InTouchViewAppObjectName](#)

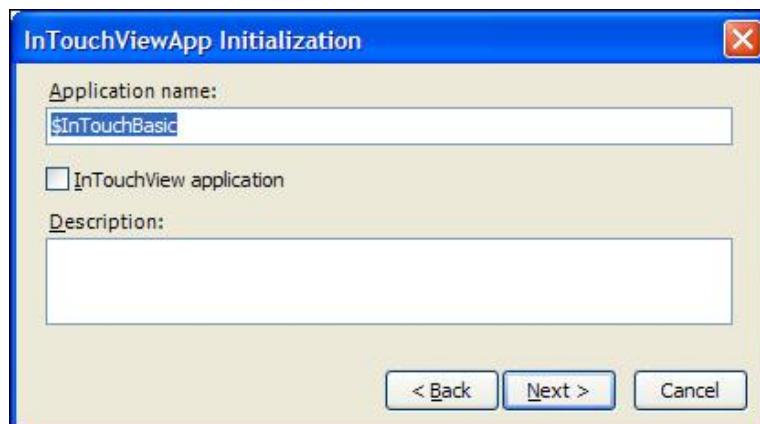
위의 Directory 는 InTouch Application Manager 가 아닌 IDE 에 의해 관리 됩니다.

새로운 Application 을 생성하고 WindowMaker 실행하기

- a. \$InTouchBasic 템플릿을 더블 클릭합니다.
InTouchViewApp Initialization 창이 나타납니다.
- b. Create New InTouch Application 을 선택한 후 , Next 를 클릭합니다.

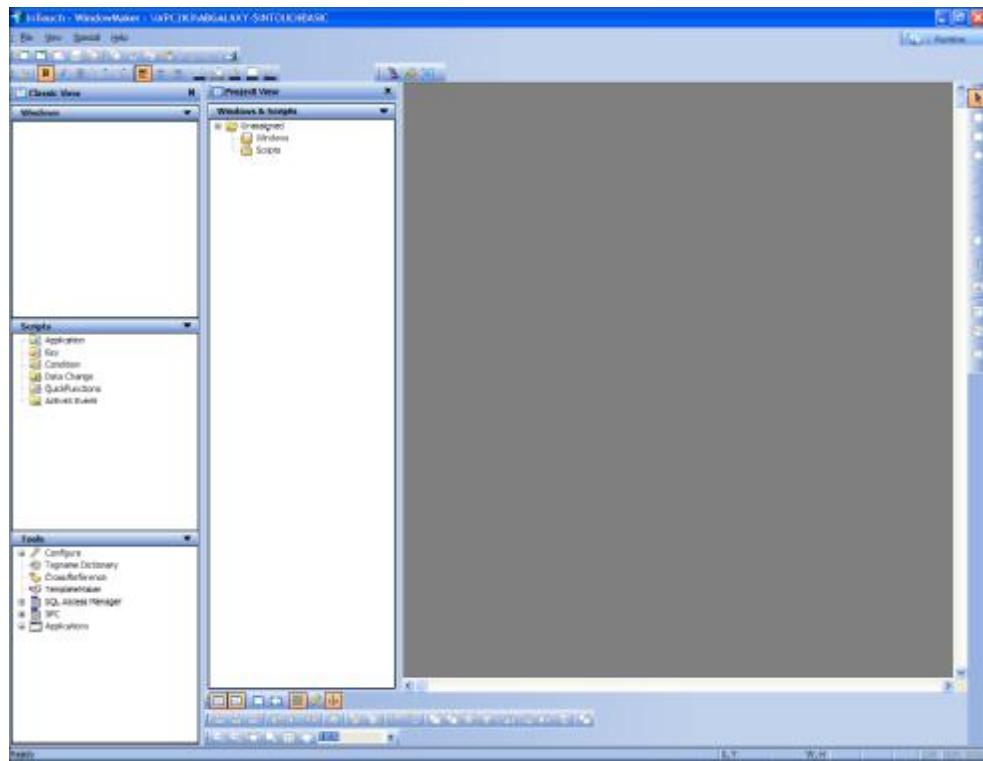


- c. InTouch Application 의 이름을 입력합니다.



Note: InTouchView Application 옵션은 선택하지 않습니다. 이것은 ArchestrA를 참조하여 외부 Data source 로 사용하기 위해 InTouch Application 을 만들 때 사용됩니다.

- d. Next 를 클릭하면 WindowMaker 가 시작되며, application 을 관리하기 위해 새롭게 생성된 \$InTouchBasic 객체가 열립니다.

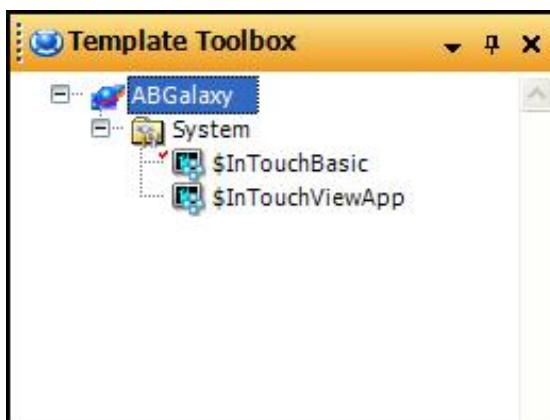


IDE - Managed Application 의 Check-out , Check – in 기능

IDE에서 InTouch Application 을 열면, 항상 자동으로 checked out 상태가 됩니다.

Application 이 Check-out 상태에서는 Application 은 read-only 상태이며, 기존의 개발자가 Check-in 상태로 만들기 전까지는 아무도 Check-out 시킬 수 없습니다.

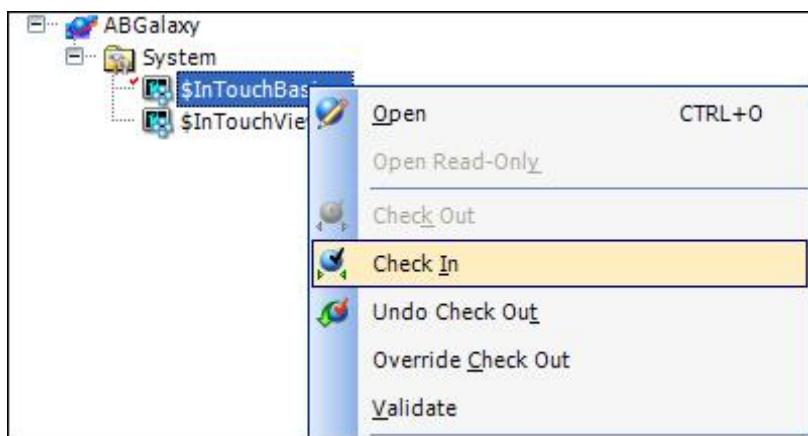
Check-out 상태의 Application 은 checkmark 표시로 상태를 알 수 있습니다.



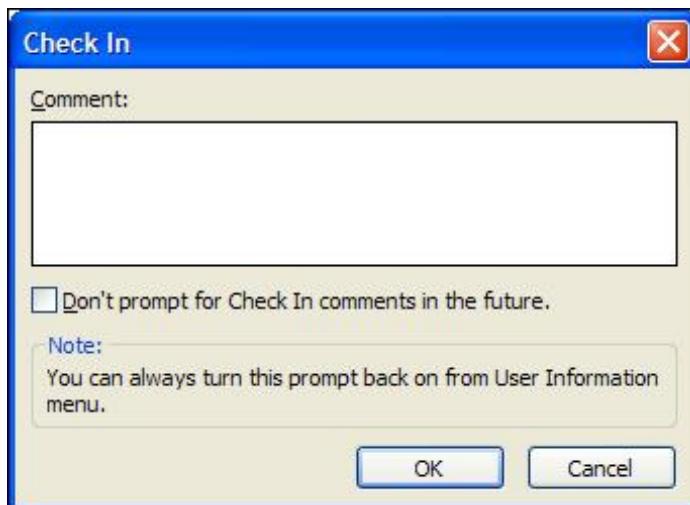
Note: InTouch Application 은 WindowMaker 를 실행하면서 자동으로 Check-out 상태가 됩니다. 만약에 Application 수정 없이 WindowMaker 를 닫으면, undo-checkout 이 자동으로 실행됩니다.

Application 을 check-in 상태로 만들기 위해서는 :

- ArchestrA IDE 를 실행합니다.

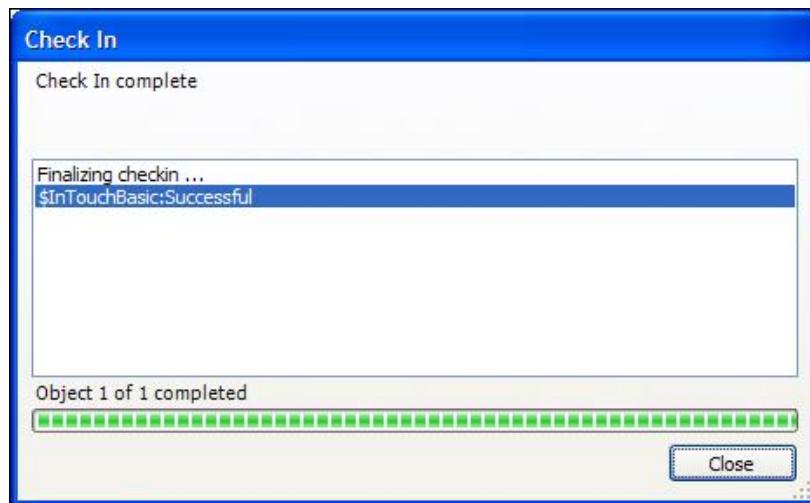


- Template Toolbox 의 Application 이름에서 오른쪽 마우스 클릭 후, Check In 옵션을 선택합니다.



c. Check In 창이 나타나면 OK를 클릭합니다.

Check In complete 창으로 Check in 상태를 확인할 수 있습니다.



d. Close 를 클릭합니다.



Application에 Checkmark 가 사라졌는지 확인합니다.



Module 2

소개

Section 1 – 소개

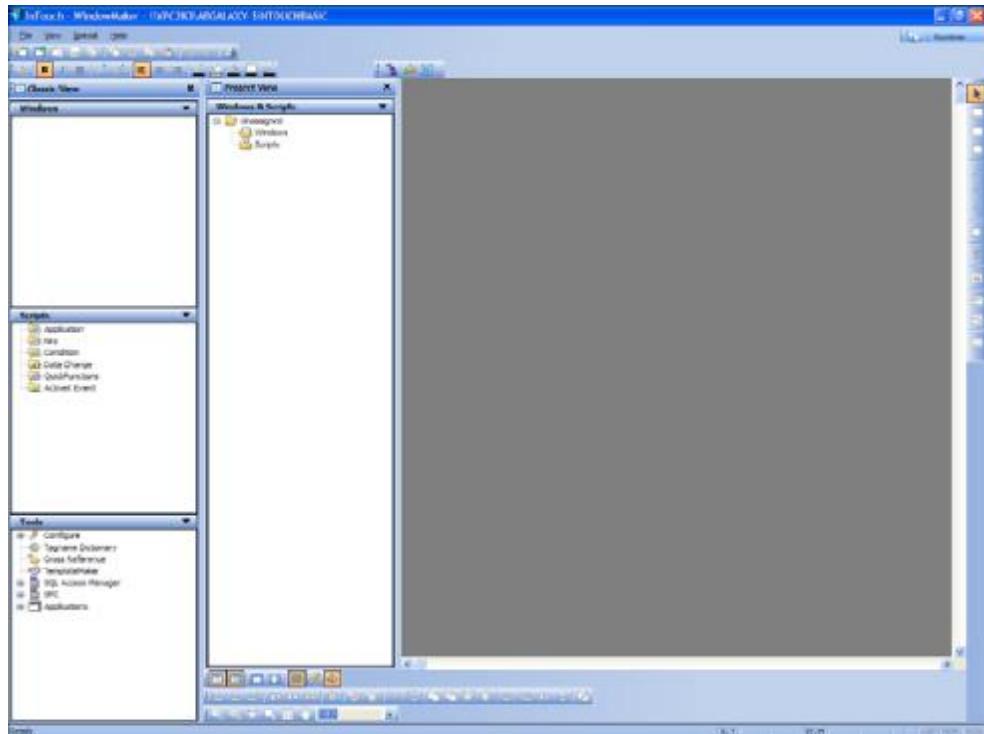
Section 2 – WindowMaker 사용

Lab 1 – 새로운 InTouch Application, 윈도우, 그래픽 생성

Section 1 - 소개

이 장에서는 tool bar, 버튼과 같은 WindowMaker의 Interface를 포함해 Application 개발 환경 및 WindowMaker의 기능들을 살펴봅니다.

The WindowMaker Interface



WindowMaker user interface는 Windows 2000과 Windows XP에 표준입니다. WindowMaker는 오른쪽 마우스 클릭 지원, tool bar의 위치이동 또는 고정, 풀다운 메뉴, 도움말 등의 기능을 포함하고 있습니다. 메뉴 오른쪽 마우스 클릭은 167,000,000 색상 지원을 제공하는 Custom color palette와 자주 사용되는 명령어에 빠르게 접근할 수 있도록 제공합니다.

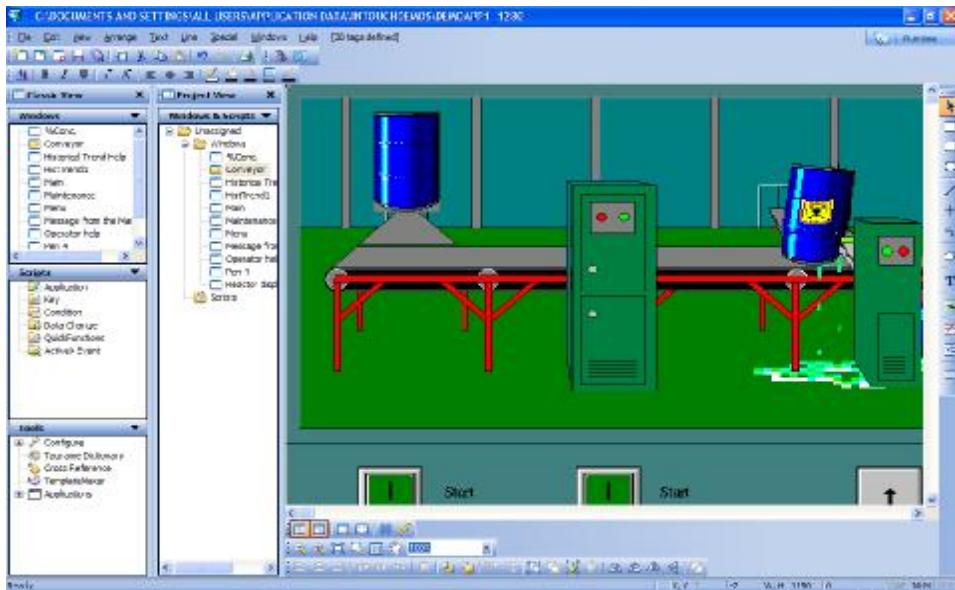
WindowMaker 개발 환경은 필요에 의해 설정할 수 있습니다. WindowMaker를 처음 열 때, Tool bar, Application Explorer, status bar를 포함되는 대부분의 요소들이 자동적으로 보여집니다.

이들 요소들은 필요에 따로 보여지게 하거나, 숨길 수 있으며, tool bar와 Application Explorer는

WindowMaker의 윈도우 안으로 위치를 이동시킬 수 있습니다.

선택적으로 ruler(자)를 보여줄 수 있고, grid visible을 On/Off 할 수 있습니다.

다음의 그림은 WindowMaker 개발 환경에서 ReactorDemo Application과 Conveyor 윈도우의 요소들을 보여줍니다. 새로운 Application을 만들고 WindowMaker를 실행할 때, Application 요소들은 자동적으로 아래와 같은 기본 설정을 보여줄 것입니다.



The Application Explorer

WindowMaker 의 Application Explorer 는 Application 요소들을 계층적인 그래픽 뷰로 제공합니다. 이 뷰들은 윈도우, 스크립트, 메뉴, wizard 와 Application 에 설정되는 다른 아이템들을 보여주고 그들에 쉽게 접근할 수 있도록 제공합니다. 뷰는 또한 공통적으로 사용되는 많은 WindowMaker 의 명령과 기능들에 빠르게 접근 할 수 있도록 합니다.

Application Explorer 는 SQL AccessManager, SPCPro, 그리고 Recipe Manager 와 같은 유ти리티 프로그램들을 보여주고, 사용자 정의 Application 을 등록할 수 있도록 합니다.

또한 Application Explorer 에서 윈도우나 다른 Application 요소들을 생성하고 설정할 수 있습니다.

Application Explorer 는 Classic View 와 Project View 두 가지 종류로 나뉘어 집니다. 이 뷰는 두 가지 타입 모두 Application 윈도우들과 이용 가능한 tool bar 를 보여주지만, 타입에 따라 다른 모습으로 요소들을 보여줍니다. Application Explorer view 또한 다른 tool bar 와 같이 화면에 보이게 하거나 숨길 수 있고, 고정시키거나 위치 이동을 할 수 있습니다.

Classic View 에 비해 Project View 의 장점은 Application 요소들의 더 나은 구조와 관리를 돋기 위한 폴더를 생성하는 것을 허용하는 것입니다.. 예로, 두 개의 다른 지역에 윈도우와 스크립트를 생성하면, 각 지역에 폴더를 생성할 수 있고, 생성된 폴더에 지역별로 윈도우와 스크립트를 이동할 수 있다는 것 입니다.

Note: Application Explorer 에 ArchestrA® 프로그램이나 윈도우 프로그램을 등록 할 수 있습니다. 이러한 기능은 HMI 과 I/O Server 또는 다른 control 들을 빠르게 전환하며 구성할 수 있도록 해 줍니다.

Warning: Application Explorer 에 WindowViewer 를 추가하지 않습니다. WindowViewer 를 추가하면 WindowMaker 안에 생성하는 새로운 윈도우들은 WindowViewer.의 윈도우와 동기화 되지 않을수 있습니다. WindowViewer 를 설치하는 적절한 방법은 File 메뉴의 WindowViewer 명령이나 WindowMaker 메뉴 바의 Runtime 클릭에 의한 것입니다.

Using Application Explorer

Application Explorer view는 WindowMaker 안으로부터 설정할 수 있습니다.

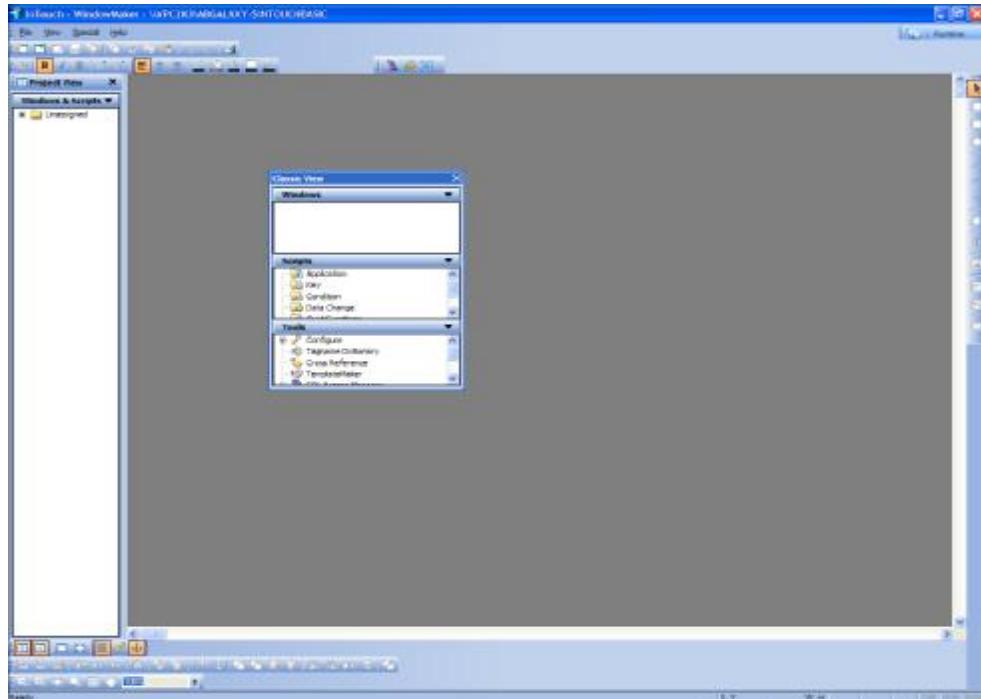
Application Explorer View 창 보이기와 숨기기

- | View 툴 바나 View / Classic View 명령의 Classic View 버튼  를 클릭합니다.
- | View 툴 바나 View / Project View 명령의 Project View 버튼  를 클릭합니다.

WindowMaker 안의 Application Explorer view는 함께 보여줄 수 있을 뿐만 아니라 WindowMaker의 원도우 가장자리에 고정시키거나, 원도우안의 어떤 곳이든 유동적으로 움직일 수 있습니다.

Application Explorer가 WindowMaker 원도우의 경계에 고정될 때, 필요에 따라 자동으로 크기가 조정되기도 하고, scroll bar가 보여지기도 합니다.

Application Explorer를 유동적으로 움직일 때, title bar가 보여지며, View의 크기와 위치를 변경할 수 있습니다.



Navigating in the Application Explorer

Application Explorer의 계층적인 그래픽 뷰 안에서 목록화된 그룹들은 펼쳐지거나 접혀질 수 있습니다. 예로, 그룹을 더블 클릭하면 펼쳐진 아이콘과 그룹의 멤버들이 보여집니다. 멤버를 더블 클릭하면 그룹은 멤버를 펼칠 것입니다. 멤버들을 포함하는 그룹은 플러스 표시 와 함께 상위에 위치합니다. 그룹을 확장하고 그것의 멤버를 보여주기 위해 클릭하거나 그룹들을 축소하고 그것의 멤버들을 숨기기 위해 클릭합니다.

Adding Applications to the Application Explorer

Application Explore 가 가지는 강력한 기능중의 하나는 WindowMaker.안에 다른 Archestra 프로그램이나, third-party Window Application 을 등록하여 사용할 수 있다는 것입니다..

예로, I/O Server 프로그램을 실행하고 설정할 수 있으며, 동시에 Application 을 개발할 수 있습니다. 또한 Windows Notepad, Wordpad, Microsoft Excel, Microsoft Word, Microsoft Paint 와 같은 자주 사용하는 third-party 윈도우 프로그램을 등록하여 사용할 수 있습니다. 프로그램 안에서 특정 문서나 spreadsheet 를 열도록 Application Explorer 에서 설정 할 수 있습니다.

Tip: InTouch add-on 프로그램, SQL Access, SPC Pro, 와 Recipe Manager 는 한번 설치되면 자동으로 Application Explorer 에 추가됩니다.

Application Explorer 에 Application 을 추가하기 위해

Note: Application Explorer view 는 Application 을 개발하기 위해 필요한 모든 요소들을 포함하고 있습니다. 필요에 따라 적절한 view 타입을 선택하여 사용할 수 있습니다. 여기에서는 Project View 만을 사용할 것입니다. Classic View 는 아이콘을 선택하여 숨겨 놓습니다.

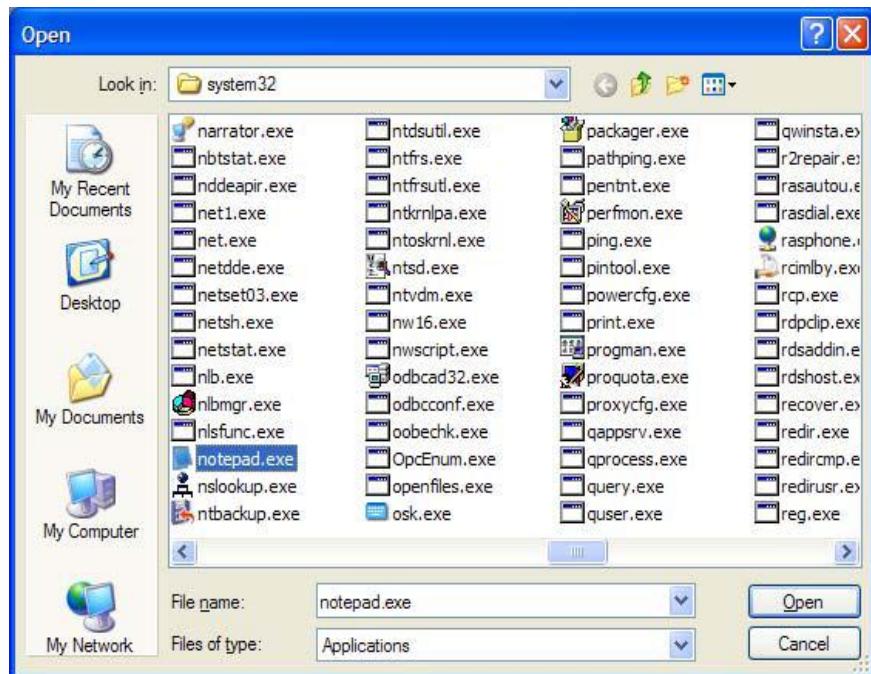
- Tools panel 안에 Application 을 오른쪽 클릭하고 New 를 선택합니다.



Application Properties 대화상자가 나타납니다.

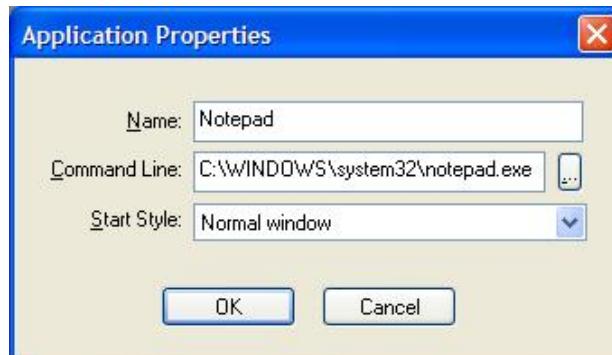


- b. Name 필드안에 Application Explorer에 보여주기 위해 이름을 입력합니다. 예로 Notepad를 입력합니다.
- c. Ellipsis 버튼 (...)를 클릭하고 Application을 선택합니다. Notepad 어플리케이션의 경로는 C:\WINDOWS\system32\notepad.exe

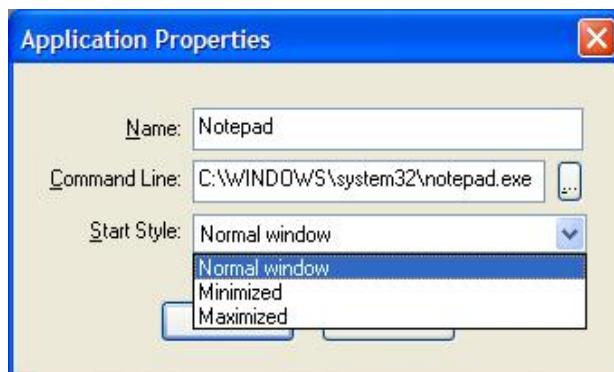


- d. Open를 클릭합니다.

Application Properties 대화상자가 다시 나타납니다.



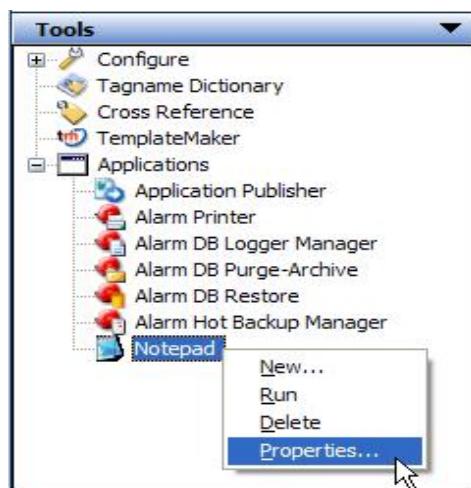
- e. WindowMaker에 등록되는 Application이 시작될 때, 어떻게 실행될지 선택하기 위해 Start Style 드롭다운 리스트를 클릭합니다. 예로, Normal window가 선택됩니다.



- f. OK를 클릭합니다.

등록한 Application은 Applications icon 아래에 추가됩니다. WindowMaker로부터 언제라도 Application을 실행할 수 있습니다.

- g. Application Properties 창을 열어 Application의 속성을 변경할 수 있으며, 그렇게 하기 위해, Application 이름에서 오른쪽 마우스를 클릭하여, Properties를 선택합니다.



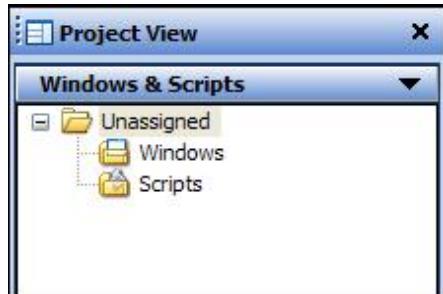
Creating a New Window

Note: Application Explorer의 두 가지 타입의 View 모두 Application 개발을 하기 위해 필요한 모든 요소들을 포함하고 있습니다. 이것들을 WindowMaker에서 숨기기 위해 **View/Hide** toggle 버튼을 사용합니다.

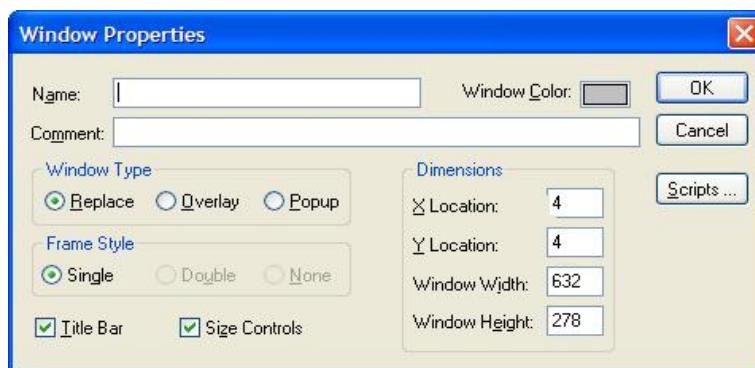
Application Explorer **Classic View**, **View**에 **Classic View** hide/show toggle 버튼  을 클릭합니다.

InTouch Application은 그래픽, 문자 객체, 애니메이션, Action Script 등을 포함한 윈도우로 구성됩니다. 윈도우를 생성할 때, 윈도우의 배경색, title bar, 위치 등을 속성에서 정의 할 수 있습니다.

- Application Explorer의 Windows & Scripts panel안에 Unassigned 폴더를 확장합니다. (Project View)



- Windows 폴더에서 오른쪽마우스 클릭을 하고 New를 선택합니다. Window Properties 대화상자가 나타납니다.



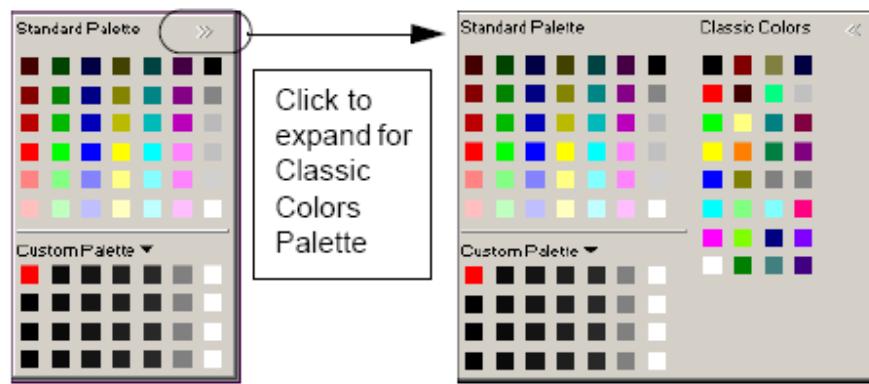
- Name 필드 안에 Scratch를 입력합니다. 이름은 32문자까지로 구성할 수 있습니다. 이름에는 공백, 마침표, 그리고 따옴표(")를 제외한 모든 문자를 사용할 수 있습니다.

Note: Window 속성창의 설정은 기본적으로 이전에 만들어 진 윈도우의 속성값 반영합니다. 만약에 어떤 윈도우 창이 열려 있을 때, 이 명령을 실행한다면, 기본 설정 값은 열려있는 윈도우의 값을 반영합니다. 또한 이 경우 윈도우 Script 가 포함되어있는 윈도우라면, Script를 새로운 윈도우에 반영할지 여부를 묻는 메시지 박스가 나타납니다.

- d. 속성창의 문자 입력란에서 오른쪽 마우스 클릭을 합니다. 입력란에 적용될 수 있는 명령들이 서브메뉴로 보일 것입니다.



- e. Comment 상자에는 윈도우와 관련된 설명을 입력합니다. 이 정보는 오직 문서 정보를 위한 것이며 Application에 사용되지 않습니다.
f. 윈도우에 배경색을 선택하기 위해 Window Color를 클릭합니다
Color palette가 나타납니다.



WindowMaker의 Color palette는 선, 사각형, 둥근 사각형, 타원, 다중선, 다각형 그리고 문자의 고정적인 색상이나, 애니메이션색상을 적용시키기 위해 사용되어집니다. 이것은 또한 윈도우의 배경 색이나 비트맵객체의 transparent 색을 선택 할 수 있도록 사용되어집니다. Color palette는 palette 상자 안의 색칠된 사각형을 클릭할 때나, 객체를 선택한 후 format tool bar의 line, fill, text 컬러를 적용할 때 사용됩니다.

Creating a Custom Color

WindowMaker에서 Custom palette에 원하는 색상을 정의하여 사용할 수 있으며, 다른 윈도우 Application에서 생성된 palette를 가져와서 표준 palette에 추가할 수 있습니다. 반대로 Custom palette를 다른 윈도우 Application으로 export 할 수도 있습니다.

- a. Color palette를 열고, Custom Palette안에 비어있는 사각형의 하나에서 오른쪽 마우스 클릭을 합니다.

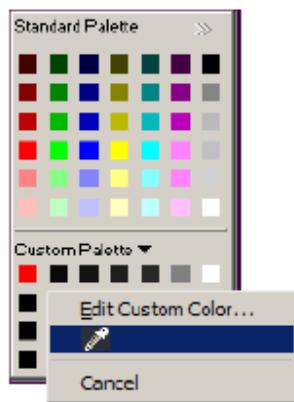


- b. Edit Custom Color 옵션을 클릭합니다.

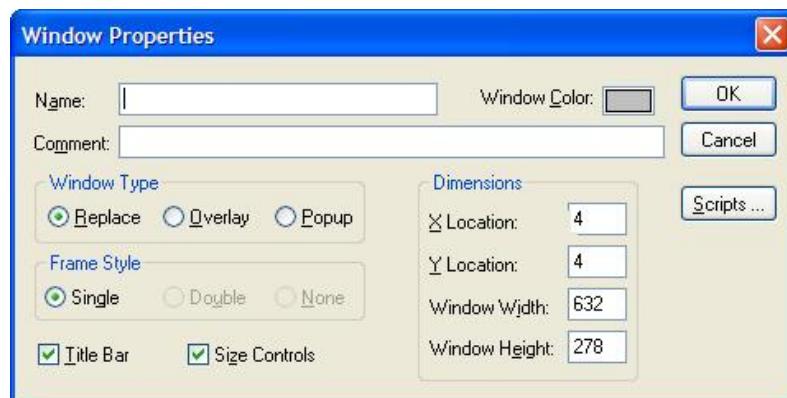
Add a Color 대화상자가 나타납니다:



- c. 메인 색 상자의 십자선이나 오른쪽 상자의 화살표를 사용하여 색을 선택합니다.
선택된 색은 ColorSolid 상자에 보여집니다. OK를 클릭하면, Custom Palette 부분에
지정한 색상이 추가됩니다.
- d. 또는 스포이드를 선택하여 추가하려는 색상을 클릭합니다. Color를 선택할 때,
WindowMaker 안의 어디든지 선택이 가능합니다.



Additional Window Properties



Window Type: 3가지 이용 가능한 옵션이 있습니다.

- | **Replace:** 원도우가 화면에 표시 되면 자신의 위치와 겹쳐지는 다른 윈도우들을 모두 닫는 윈도우 속성입니다.
- | **Overlay:** 원도우가 현재 화면에 있는 윈도우 위에 나타나며 겹쳐지는 위치의 윈도우를 상위에 나타날 수 있습니다. 이 **Overlay** 윈도우가 닫히게 되면 가려져 있던 윈도우가 다시 나타납니다. 이 윈도우 뒤에 가려져 있는 윈도우에 포커스를 주게 되면 그 윈도우가 앞으로 나오게 되고 이제 그 윈도우가 활성 윈도우가 됩니다.
- | **Popup:** 원도우는 **Overlay** 윈도우와 비슷하나 이것은 열린 모든 윈도우 상위에 항상 있게 됩니다.(뒤에 있는 다른 윈도우를 클릭하더라도) 이 **Popup** 윈도우는 사용자가 닫아야 닫혀집니다.

e. Scratch 윈도우에 기본 설정 (replace)이 사용됩니다.

Frame Style: 세가지 이용 가능한 유형이 있습니다.

- | **Single:** Title bar와 Size Control을 가질 수 있는 3D bordered window
- | **Double:** Title bar 와 Size Controls 없이 크기 조절을 할 수 없는 3D bordered window
- | **None:** Size Controls를 할 수 없고, border가 없는 윈도우.(Size Control을 가지면, 크기 조절이 가능한 3D bordered window)

f. Scratch 윈도우에 기본 설정 (Single)이 사용됩니다.

Title Bar: Title Bar는 클릭과 드래그에 의해 윈도우를 움직이는데 사용될 수 있습니다.

g. Scratch 윈도우에 기본설정 (checked)가 사용됩니다.

Size Controls: WindowMaker안에 윈도우크기를 재 조정 할 수 있습니다.

h. Scratch 윈도우에 기본설정 (checked)이 사용됩니다

Dimensions: 윈도우의 각 좌표에 픽셀의 위치를 입력합니다.

- | X Location: WindowMaker 디자인영역의 왼쪽 끝에서 정의된 윈도우의 왼쪽 끝까지의 픽셀 수를 나타냅니다.
- | Y Location: WindowMaker 디자인 영역의 가장 위부터 정의된 윈도우의 위쪽 끝까지의 픽셀 수를 나타냅니다.
- | Window Width: 필드에 윈도우의 넓이를 픽셀 수로 입력합니다

Note: 윈도우는 윈도우의 최소 넓이를 제한합니다. 예를 들어 표준 VGA에서는 102 가 최소 픽셀 크기입니다.

- | Window Width: 필드에 윈도우의 높이를 픽셀 수로 입력합니다.

Note: 기본적으로, 이를 필드들의 값들은 이전에 생성된 윈도우의 값으로 설정될 것입니다.. 또한 WindowMaker안에 윈도우의 크기가 수동으로 변경될 때 자동적으로 수정됩니다.

기본 설정을 Scratch 윈도우에 적용합니다.

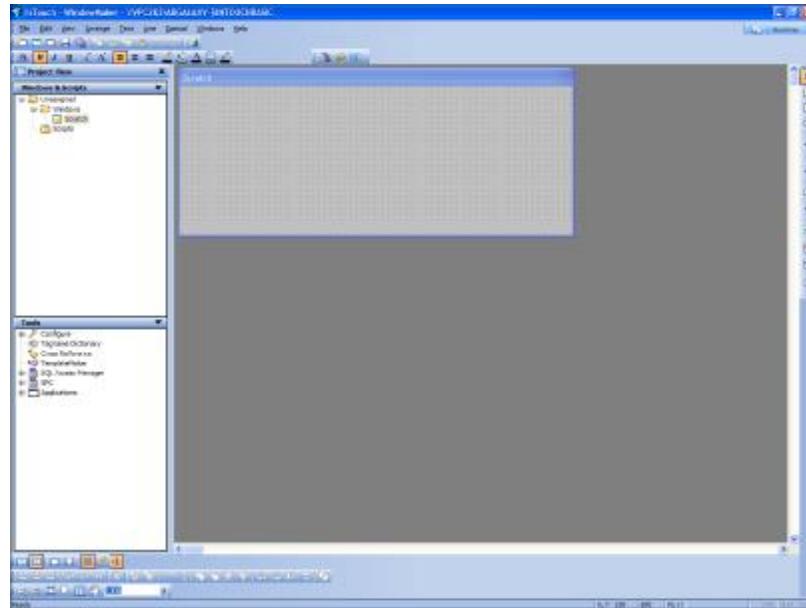
Scripts 버튼: Window Script 에디터를 엽니다. 스크립트의 3가지 유형을 적용할 수 있습니다.

- | On Show: 윈도우가 열릴 때 한 번만 실행
- | While Showing: 윈도우가 보이는 동안 Every Msec 필드에 있는 속도로, 매 Msec 마다 반복 실행
- | On Hide: 윈도우가 닫힐 때 한번 실행

Note: 만약에 활성화 되어있는 윈도우에 윈도우 Script가 정의되어 있다면, 새로운 윈도우를 생성할 때, 그 Script를 복사해서 사용할 수 있습니다.

- i. OK를 클릭합니다.

WindowMaker안에 새로운 윈도우가 나타나고, toolbar 아이콘이 활성화 됩니다.



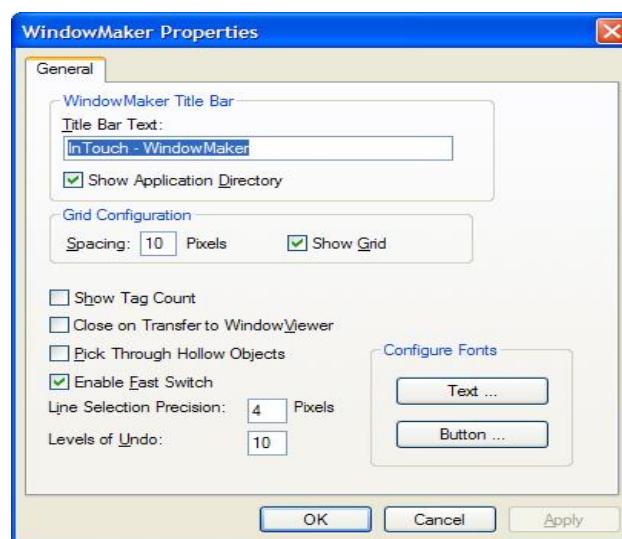
Customizing the Development Environment

WindowMaker는 사용자 정의된 어플리케이션을 구성하기 위해 많은 설정 옵션을 포함합니다.

예로, 회사 이름, grid의 픽셀 넓이 설정, 버튼이나 문자 객체에 폰트/스타일/크기 설정 등을 포함하여 Application의 title bar 바의 문자열까지 사용자가 정의할 수 있습니다.

Setting WindowMaker Properties

- Special / Configure / WindowMaker를 클릭합니다



Title Bar Text: Runtime 시 Application의 title bar 바에 보여주기 위해 제목을 입력합니다.

Show Application Directory: Title bar안에 Application 디렉토리의 경로를 포함시킵니다.

Spacing: Grid 간격의 픽셀의 숫자를 입력합니다.

Show Grid: WindowMaker의 Snap to grid가 작동될 때 윈도우에 grid가 보여집니다.



Grid를 on/off 하기 위해 View toolbar의 Snap to Grid를 클릭하거나 Arrange / Snap to Grid 명령을 클릭합니다.

Close on Transfer to WindowViewer: WindowViewer가 시작될 때 WindowMaker는 자동적으로 종료 되어집니다.

메모리가 문제되지 않고, WindowViewer와 WindowMaker를 자주 변환한다면 이 옵션은 사용하지 않는 것이 좋습니다.

Note: 이 옵션을 선택한다면, WindowViewer Properties General Properties에 있는 Close WindowViewer 옵션은 자동적으로 선택됩니다.

Pick Through Hollow Objects: 선택된 개체의 테두리 모양을 설정합니다..

Enable Fast Switch: 메뉴 바에 개발모드와 실행모드의 빠른 명령 전환을 보여줍니다.
(WindowMaker 와 WindowViewer)

WindowMaker에서 이 옵션이 선택되면 Fast switch는 화면의 오른쪽 위에 Runtime 문자이고, WindowViewer에서는 Development 문자입니다. 두 프로그램 사이에 빠른 전환을 하기 위해 Fast switch를 클릭합니다.

Note: Fast switch를 사용할 때, WindowMaker는 WindowViewer가 시작되기 전에 열려있는 윈도우의 변경사항들을 자동적으로 저장합니다.

Line Selection Precision: 커서가 선으로부터 떨어질 수 있고 그것을 선택할 수 있는 픽셀의 수를 입력합니다. 대부분의 경우 4로 기본설정이 되어있습니다..

Levels of Undo: Undo/redo 단계(25단계까지)의 수를 입력합니다.. undo/redo 기능을 사용하지 않기 위해서는 0를 입력합니다.

하나의 단계는 하나의 행동을 말합니다. Undo 와 redo 기억장치는 새로운 윈도우가 생성되거나 윈도우가 열릴 때, 모두 삭제됩니다. 또 한 두 기억장소 모두 윈도우가 저장될 때 비워지게 됩니다.

Note: 이를 설정이 변경된 후, 변경사항을 적용하기 위해서는 WindowMaker를 재 시작해야 합니다.

Configure Fonts area:

Text: 텍스트 객체에 기본 글꼴, 크기 형식을 선택합니다.

Button: 버튼 객체에 기본 글꼴, 크기 형식을 선택합니다.

WindowMaker Toolbars

WindowMaker toolbars 아이콘들은 공통적인 기능으로 그룹화 되어있습니다. 예로, Arrange Toolbar는 Arrange메뉴에 있는 명령의 대부분들을 빠르게 적용하기 위해 사용될 수 있는 도구들을 포함합니다.

WindowMaker안에 toolbar들은 WindowMaker 의 모서리에 고정될 수 있고, 또한 유동적으로 움직일 수 있습니다. Toolbar들이 유동적으로 움직일 때 toolbar의 이름은 title bar안에 보여집니다.

아이콘 위에 커서가 위치할 때 Tool Tips 상자가 툴의 이름을 보여줄 것입니다



General Toolbar

General toolbar는 File메뉴에 속한 윈도우 명령의 대부분을 실행하는 아이콘들을 구성합니다. 아이콘은 또한 Edit 메뉴 안의 마이크로소프트 윈도우 클립보드를 포함합니다.



Button Description

-  File 메뉴의 New Window 명령을 실행하고 새로운 윈도우를 생성하기 위해 Windows Properties를 엽니다.
-  File메뉴의 Open Window 명령을 실행하며, Windows to open 대화상자가 열리고, 열고자 하는 윈도우를 선택합니다..
-  File메뉴의 Close Window 명령을 실행하며, Windows to close 대화상자가 열리고, 닫고자 하는 윈도우를 선택합니다.
-  File 메뉴의 Save Window 명령을 실행하며, Windows to save 대화상을 열어, 저장하고자 하는 윈도우를 선택합니다.
-  현재 열려 있는 모든 윈도우가 자동적으로 저장됩니다. 이 메뉴는 윈도우를 저장할 때 확인을 요구하지 않습니다. 윈도우에 수정된 모든 것을 자동적으로 저장합니다.
-  윈도우에 현재 선택된 객체를 복제하기 위해 Edit 메뉴에 Duplicate명령을 실행합니다.
-  Edit 메뉴에 Cut 명령을 실행합니다. 현재 선택된 객체를 잘라 윈도우 클립보드에 복사합니다.

-  Edit메뉴에 Copy를 실행합니다. 현재 선택된 객체를 복사해서 윈도우 클립보드에 복사합니다. (복사된 객체는 윈도우로부터 지워지지 않습니다)
-  Windows Clipboard에 자르거나 복사하는 어떠한 객체를 붙이기 위해 Edit 메뉴에 Paste명령을 실행합니다.(붙이기 모드로 커서가 변경됩니다. 윈도우에서 복사된 것을 붙이거나 복사하기 위해 클릭합니다.)
-  객체에 적용된 명령이나 지난 행동을 되돌리기 위해 Edit메뉴에 Undo명령을 실행합니다.
-  객체에 적용된 명령이나 지난 undo 행동을 되돌리기 위해 Edit메뉴에 Undo명령을 실행합니다
-  데이터베이스, 윈도우 정보와 QuickScripts를 프린트 하기 위해 사용되는 WindowMaker Printout 대화상자를 열기 위해 File 메뉴의 Print 명령을 실행합니다.

Wizards Toolbar

Wizards toolbar는 wizard Selection dialog 상자와 SmartSymbol Wizard에 접근하기 위한 도구들을 포함합니다. 설치되어 있는 Wizard 나 ActiveX control를 tool bar에 추가 할 수 있습니다.



Note: InTouch Application에서는 Wizard tool bar에 Embed ArchesrA Graphic Wizard 버튼은 보여주지 않습니다. 이것은 InTouch Managed Application에만 이용 가능한 특징입니다.

Button Description

-  Application 에 Wizard 객체를 사용하기 위해 Wizard Selection 대화상자를 보여줍니다.
-  Application 에 ArchesrA Symbol들을 사용하기 위해 Embed ArchesrA Graphic Wizard 대화상자를 보여줍니다. Managed Application에서만 사용 가능 합니다.
-  Application 에 SmartSymbol들을 사용하기 위해 SmartSymbol Wizard를 보여줍니다.

Format Toolbar

Format toolbar는 Text 메뉴에 속한 문자 객체 형식 명령을 실행하는 도구들로 그룹화 되어있습니다. 선, 채우기 , 문자, 윈도우 배경에 Color palette를 사용하기 위한 도구들을 포함하고 객체의 색상을 변경합니다.



Button Description

-  폰트와 스타일, 크기를 선택하기 위해 사용되는 **Font** 대화 상자를 여는 **Text** 메뉴의 **Font** 명령을 실행합니다.
-  **B** **Text** 메뉴에 **Bold** 명령을 실행하며, 선택된 문자 필드를 **Bold** 스타일로 변환합니다.
-  **I** **Text** 메뉴에 **Italic** 명령을 실행하며, 선택된 문자 필드를 **Italic** 스타일로 변환합니다.
-  **U** **Text** 메뉴에 **Underline** 명령을 실행하며, 선택된 문자 필드에 **Underline** 스타일로 변환합니다.
-  **A** 크기 조정 가능한 폰트의 크기를 줄이기 위해 **Text** 메뉴에 **Reduce Font** 명령을 실행합니다. 문자열을 선택하고 뒤 이 메뉴를 클릭함으로써 명령이 적용됩니다.
-  **A** 크기 조정 가능한 폰트의 크기를 늘리기 위해 **Text** 메뉴에 **Enlarge Font** 명령을 실행합니다. 문자열을 선택하고 뒤 이 메뉴를 클릭함으로써 명령이 적용됩니다.
-  단일 또는 다중 선택된 문자열과 숫자 값 필드를 왼쪽 정렬하기 위해 **Text** 메뉴에 **Left Justified** 명령을 실행합니다.
-  단일 또는 다중 선택된 문자열과 숫자 값 필드를 가운데 정렬하기 위해 **Text** 메뉴에 **Centered** 명령을 실행합니다.
-  단일 또는 다중 선택된 문자열과 숫자 값 필드를 오른쪽 정렬하기 위해 **Text** 메뉴에 **Right Justified** 명령을 실행합니다.
-  선 객체나 객체의 외각 선의 색을 선택하기 위해 **Color palette**를 엽니다.
-  객체 색 채우기를 선택하기 위해 **Color palette**를 엽니다.
-  문자 객체에 색을 선택하기 위해 **Color palette**를 엽니다.
-  윈도우 배경색을 선택하기 위해 **Color palette**를 엽니다.
-  비트맵 객체의 투과색상을 선택하기 위해 **Color palette**를 엽니다.

Drawing Toolbar

Drawing toolbar는 간단한 그래픽 객체들(사각형, 타원, 선 또는 문자 객체)과 복잡한 객체(실시간 트랜드, 이력 트랜드, 비트맵)와 3-dimensional 버튼(버튼과 라벨)을 그리기 위해 모든 도구와 그룹화 되어집니다..



Button Description

- 원도우에서 객체를 선택하기 위해 **Selector mode**를 사용합니다.
- Rectangle tool** 을 사용하여 사각형을 그립니다.
- Rounded rectangle tool**을 사용하여 둥근 사각형을 그립니다..
- Ellipse tool** 을 사용하여 원 또는 타원을 그립니다.
- Line tool** 을 사용하여 여러 각도에서 선을 그릴 수 있습니다.
- Line tool** 을 사용하여 직 교선을 그립니다.
- Line tool** 을 사용하여 다각선을 그립니다..
- Shape tool** 을 사용하여 다각면을 그립니다.
- Text tool** 을 사용하여 문자객체를 입력합니다..
- 비트맵은 윈도우 클립보드 또는 그래픽파일 (.BMP, .JPG, .PPCX, .TGA) 을 가져오기 위해 사용합니다.
- 실시간 트랜드 객체를 그리기 위해 사용합니다.
- 이력 트랜드 객체를 그리기 위해 사용합니다.
- 3D버튼과 라벨을 그리기 위해 사용합니다.

View Toolbar

View toolbar는 View 메뉴에 속한 윈도우 명령들을 실행하는 도구들로 그룹화되어 있습니다. 이를 명령은 WindowMaker 윈도우의 상태를 조작하기 위해 사용됩니다.



Note: InTouch Application에서는 View toolbar는 ArchesrA Graphic Anchor Points 아이콘이 존재하지 않습니다. 이것은 InTouch Managed Applications에만 사용 가능한 툴입니다.

Button Description

- Application Explorer Classic View 보여주기/숨기기
- Application Explorer Project View 보여주기/숨기기
- 모든 도구 바들을 숨기기 또는 보이게 하기 위해 View 메뉴에 Hide All 명령을 On, off 합니다.
- 전체 화면 Mode 를 실행합니다.
- Grid 를 보여주거나 숨기기 위해 사용합니다.
- Ruler 를 보여주거나 숨기기 위해 사용합니다.
- ArchesrA Graphic Anchor Points toggle 버튼이며, 애니메이션을 목적으로 ArchesrA Symbol 을 특정 위치에 빠르게 고정하여 나타나도록 합니다.

Snap to Grid

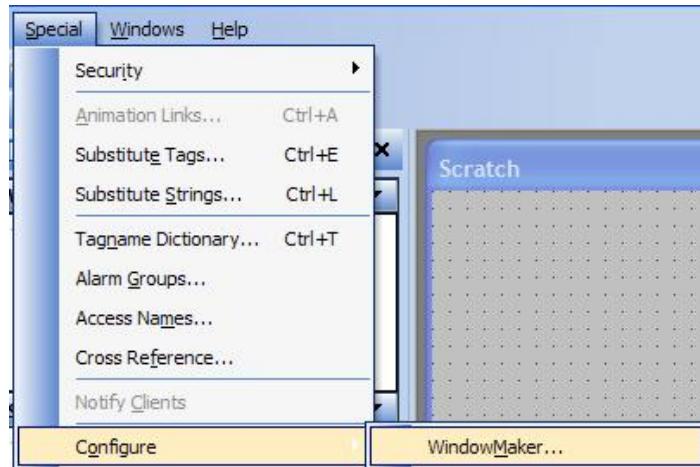
Grid 옵션을 작동시키게 되면, 윈도우에 추가된 객체들을 정렬할 경우 그래픽 객체가 Grid의 상위 왼쪽 픽셀의 간격에 맞춰집니다. 만일 다수의 객체를 선택한다면 객체는 최초선택 객체가 상위 왼쪽 모서리에 맞춰지게 됩니다.

Grid 옵션을 작동시킨 상태에서는, 객체 이동 시 방향키만을 사용하여 이동할 경우는 10 픽셀씩, Shift를 누르고 방향키를 움직이면 20 픽셀, Ctrl을 누르고 이동할 시에는 40 픽셀씩 이동합니다.

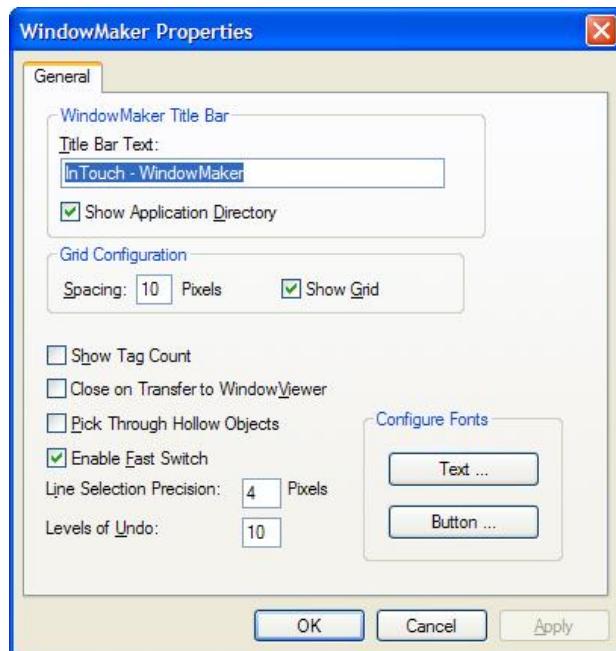
이러한 Grid 옵션은 객체를 보기 좋게 정렬하거나, 적합한 위치로 이동시킬 경우에 매우 유용합니다.

Configuring the Grid

- a. Special / Configure / WindowMaker를 선택합니다.



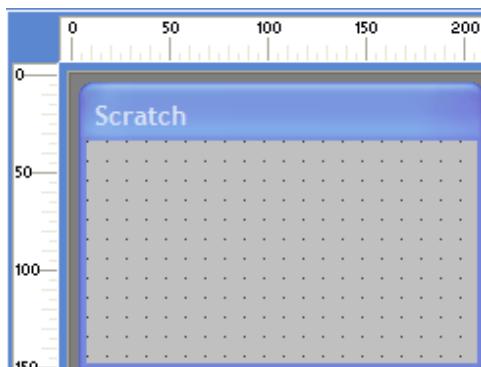
WindowMaker Properties 대화상자가 나타납니다.



- b. Spacing상자에 픽셀 수를 입력합니다.
 c. WindowMaker의 grid에 snap 기능이 이용 가능할 때 윈도우에서 visible grid를 원하면 Show Grid 옵션을 선택합니다.
 d. 변경을 저장하기 위해 OK를 선택하거나, WindowMaker Properties 대화상자를 종료하기 위해 Cancel를 선택합니다.

Ruler

 WindowMaker ruler 는 원도우에서 객체의 정확한 정렬을 하기 위해 사용될 수 있으며, 애니메이션이 순차적으로 발생하는 동안 객체의 움직임을 결정합니다.

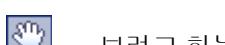


Pan and Zoom Toolbar

Pan 과 Zoom toolbar 는 화면 확대 축소를 할 수 있으며, 원도우에서 빠르게 움직일 수 있습니다.



Button Description

-  클릭한 곳으로부터 Zoom out
-  클릭한 곳으로부터 Zoom in
-  기본 확대 단계에 Zoom Normal 상태로 전환합니다. (100%)
-  Zoom in 에 특정한 지역을 선택하여 보기 위해 마우스를 사용하여 Rubber Band Zoom 을 사용 합니다.
-  Thumbnail 화면이 보여지며, 축소 화면 안에서 Red box 를 통하여 원도우를 줌인 Zoom out 할 수 있습니다.
-  보려고 하는 지역으로 위치를 이동합니다..
-  100% 확대 하고자 하는 레벨을 입력하거나 선택하기 위해 사용합니다.

Arrange Toolbar

Arrange toolbar 는 Arrange 메뉴에서 속한 객체 정렬 명령을 실행하는 도구들로 그룹화 되어있습니다. Arrange toolbar 요소들 아이콘들은 하나 또는 그 이상의 객체가 선택되면 활성화됩니다.

Note: Arrange Toolbar에서 다음의 기능은 Archestra Graphics에서는 활성화 되지 않습니다 : Make Symbol, Break Symbol, Make Cell, Break Cell



Button Description

- Arrange / Align 하위메뉴에 Align Left 명령을 실행합니다. 선택된 모든 객체를 왼쪽정렬 합니다.
- Arrange / Align 하위메뉴에 Align Center 명령을 실행합니다. 선택된 모든 객체를 중앙 정렬합니다.
- Arrange / Align 하위메뉴에 Align Right 명령을 실행합니다. 선택된 모든 객체를 오른쪽 끝으로 정렬합니다.
- Arrange / Align 하위메뉴에 Align Top 명령을 실행합니다. 선택된 모든 객체를 가장 상위에 맞춰 정렬합니다.
- Arrange / Align 하위메뉴에 Align Middle 명령을 실행합니다. 선택된 모든 객체를 가장 중앙에 정렬 합니다.
- Arrange / Align 하위메뉴에 Align Bottom 명령을 실행합니다. 선택된 모든 객체를 가장 아래 쪽 객체에 맞춰 정렬합니다.
- Arrange / Centerpoints 하위메뉴에 Align Bottom 명령을 실행합니다. 선택된 개체를 상하 좌우의 정 중앙으로 모아줍니다.
- 선택되지 않은 객체 뒤로 선택된 객체를 놓기 위해 Arrange 메뉴에 Send to Back 명령을 실행합니다.
- 선택되지 않은 객체 앞으로 선택된 객체를 놓기 위해 Arrange 메뉴에 Send to Front 명령을 실행합니다.
- 선택된 모든 객체의 간격을 좌우로 맞춰 정렬합니다.
- 선택된 모든 객체의 간격을 상하로 맞춰 정렬합니다.



심볼이라고 불리는 단일 객체로 다중 객체들을 결합하기 위해 **Arrange** 메뉴에 **Make Symbol** 명령을 실행합니다. **ArchestrA Graphics**에는 적용되지 않습니다.



심볼을 쪼개기 위해 **Arrange** 메뉴에 **Break Symbol** 명령을 실행합니다. **ArchestrA Graphics**에는 적용되지 않습니다.



셀이라고 불리는 단일 조합 안으로 선택된 다중 객체들을 조합하기 위해 **Arrange** 메뉴에 **Make Cell** 명령을 실행합니다. **ArchestrA Graphics**에는 적용되지 않습니다.



선택된 셀을 쪼개기 위해 **Arrange** 메뉴에 **Break Cell** 명령을 실행합니다. **ArchestrA Graphics**에는 적용되지 않습니다.



선택된 객체를 90도 시계 방향으로 회전 시키기 위해 **Arrange** 메뉴에 **Rotate Clockwise** 명령을 실행합니다.



선택된 객체를 90도 시계 반대방향으로 회전 시키기 위해 **Arrange** 메뉴에 **Rotate CounterClockwise** 명령을 실행합니다.



선택된 객체를 수평으로 반전시키기 위해 **Arrange** 메뉴에 **Rotate Flip Horizontal** 명령을 실행합니다.



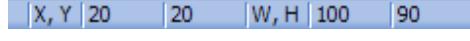
선택된 객체를 수직으로 반전시키기 위해 **Arrange** 메뉴에 **Rotate Flip Vertical** 명령을 실행합니다.



다각형 또는 다각선 모양을 수정하기 위해 **Edit** 메뉴에 **Reshape Object** 명령을 실행합니다.

WindowMaker Status Bar

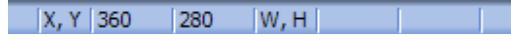
윈도우에 객체가 선택될 때, **WindowMaker status bar**는 객체의 상단 왼쪽의 X, Y의 픽셀 좌표와 객체의 높이와 넓이를 보여줍니다. X와 Y 픽셀 좌표들은 윈도우로부터 분리되어 객체에 보여집니다.



다중 객체가 선택 될 때 **status bar**는 전체 그룹에 대한 넓이와 높이를 보여줍니다.



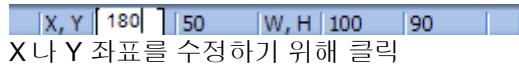
윈도우의 공백 부분을 클릭했을 때, **status bar**는 윈도우 커서의 현재 위치에 대한 X와 Y의 좌표를 보여줍니다.



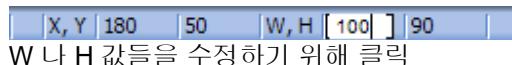
Manual Positioning and Sizing

WindowMaker status bar 바에서 객체의 위치와 크기를 수동으로 변경할 수 있습니다..

윈도우에 객체를 선택하고 정확한 X,Y 좌표와 W,H 값을 입력하기 위해 컨트롤들을 클릭합니다. 또한 여러 객체를 선택할 수 있고 이동하거나 그들을 그룹처럼 재 조정할 수 있습니다.



X나 Y 좌표를 수정하기 위해 클릭



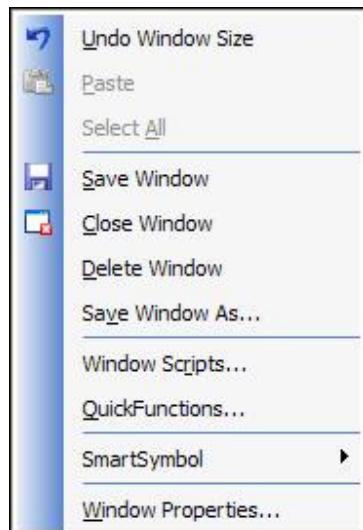
W나 H 값을 수정하기 위해 클릭

사각형, 둥근 사각형을 그리고 타원과 정확한 원을 생성하기 위해 Shift 키를 누른 상태에서 객체를 그립니다.

Note: 객체의 위치와 크기를 수동으로 바꿀 때, 입력 후 Enter를 입력 해야 합니다.

Right-Click Window Menus

열려있는 윈도우의 빈 공간에서 오른쪽 마우스 클릭을 하면, 윈도우에 적용될 수 있는 다양한 명령들에 접근할 수 있습니다.

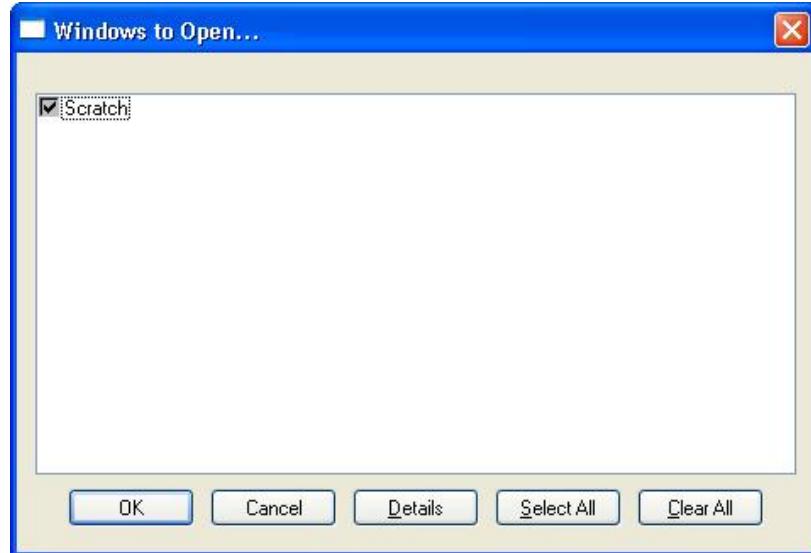


Opening Windows

Application 을 개발할 때, 컴퓨터 메모리가 지원하는 만큼 많은 윈도우를 열수 있습니다.



- Open Window 아이콘을 클릭합니다.
Windows to Open 대화 상자가 나타나며, Application에 모든 윈도우의 이름이 목록화됩니다. 기본적으로, 모든 현재의 열린 윈도우는 체크되어 있습니다..
- 열고자 하는 윈도우의 이름 옆에 체크박스를 클릭합니다.
- OK를 클릭하고 대화상자를 닫으면, 선택된 윈도우를 열 수 있습니다.



Note: Application Explorer에서 윈도우를 빠르게 열기 위해, Application 안에 모든 윈도우의 이름의 확장된 목록에 Windows를 더블 클릭합니다. 또한 윈도우 이름을 오른쪽 마우스 클릭으로 Open 을 선택합니다.

Saving Windows

새로 생성된 윈도우를 닫거나, 존재하는 윈도우의 그래픽, Script 속성을 닫을 때, 윈도우를 저장해야 합니다.



- Save Window 도구 를 클릭합니다. Windows to Save 대화상자는 저장이 필요한 모든 윈도우들의 이름들 목록을 나타냅니다.
- 저장하고자 하는 윈도우의 이름 옆에 체크박스를 클릭합니다.
- Scratch 윈도우를 저장하기 위해 OK를 클릭합니다.

Note: 수정된 윈도우를 닫기 위해 시도한다면, WindowMaker 가 윈도우를 닫기 전에 변경들을 저장 할 것입니다.

Note: Application Explorer에서 윈도우를 빠르게 저장하기 위해 윈도우 이름에서 오른쪽 마우스 클릭을 한 후, Save 를 선택합니다. 또한 윈도우의 어떠한 공백지역을 오른쪽 마우스 클릭을 해 Save Window 를 선택 할 수 있습니다. 현재 열린 모든 윈도우를 빠르게 저장하기



위해, Save All 윈도우 도구  또는 File / Save All Windows 를 클릭합니다.

Closing Windows



- Close Window 도구 를 클릭합니다. Windows to Close 대화상자가 현재 열린 모든 윈도우의 이름을 목록화하여 보여줍니다.
- 닫고자 하는 윈도우의 이름 옆에 체크박스를 클릭합니다.
- 대화상자를 닫기 위해 OK를 클릭하고 선택된 윈도우가 닫힙니다.

Note: Application Explorer에서 윈도우를 빠르게 닫기 위해, 윈도우 이름에 오른쪽 마우스 클릭 후, Close를 선택합니다. 또한 윈도우의 빈 공간에서 오른쪽 마우스 클릭한 후 Close Window를 선택할 수 있습니다.

Deleting Windows

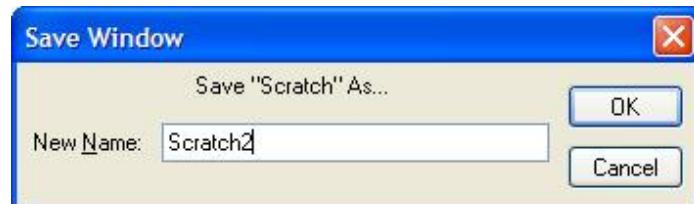
삭제 된 윈도우들은 저장되지 않았다면 복구 될 수 없습니다. 선택한 각 윈도우의 삭제를 확인 할 것 입니다.

- File / Delete Window를 선택합니다. Windows to Delete 대화상자는 현재 열린 모든 윈도우들의 이름 목록을 나타냅니다.
삭제할 윈도우의 이름 옆에 체크박스를 클릭합니다. OK를 선택하고 대화상자를 닫으면 윈도우가 삭제됩니다.
- 윈도우 삭제하지 않고 대화 상자를 종료 하기 위해 Cancel를 클릭합니다.

Duplicating Windows

윈도우를 존재하는 윈도우와 똑같이 복제하여 생성합니다. 윈도우가 열려있어야만 복제될 수 있습니다.

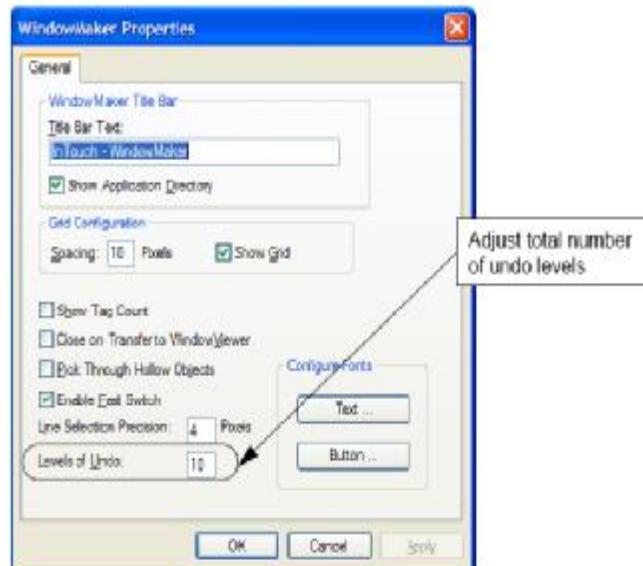
- WindowMaker 메뉴로부터 File / Save Window As를 선택합니다.
Window to save under new name 대화 상자가 나타나며, 현재 열린 윈도우들의 이름 목록을 보여줍니다.
- 복사(오직 한 윈도우 이름이 선택될 수 있습니다.)하기 원하는 윈도우를 선택합니다.
예제에서는, Scratch를 선택합니다. Save Window 대화상자가 나타납니다.
- New Name 상자에 새로운 윈도우에 대한 유효한 이름을 입력합니다.



- 대화 상자를 닫기 위해 OK를 클릭하면 복제된 윈도우가 생성됩니다.
예제에서는, Save Window 대화 상자를 종료하기 위해 Cancel를 클릭합니다.

Other WindowMaker Properties

Levels of Undo: 저장할 undo /redo 단계의 숫자를 입력합니다. Undo/redo 기능을 off 하기 위해 0를 입력합니다.



Section 2 - WindowMaker 사용

Graphic Objects

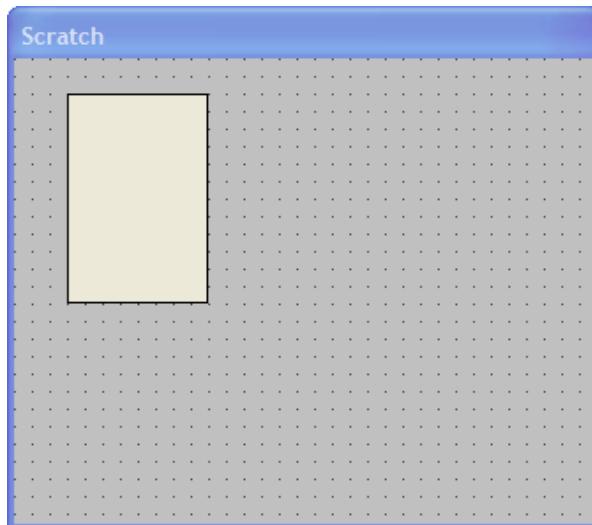
WindowMaker는 다양한 그래픽 객체를 수정, 정렬하기 위해 많은 도구들을 제공합니다.

Drawing Objects

- WindowMaker를 최대화합니다.



- Drawing **tool bar**에서 Rectangle 아이콘 을 클릭합니다.
- Scratch 윈도우 위로 커서를 이동합니다. 커서가 십자모양 (+)으로 변경됩니다.
- Scratch 윈도우를 클릭하고 작은 사각형(클릭하고 왼쪽 마우스 버튼을 떼지 않은채 아래에서 오른쪽 또는 왼쪽으로 드래그합니다.)을 그립니다. 사각형의 크기가 적절할 때 마우스 버튼을 높습니다.
- 새로운 사각형의 선택 해제를 위해 윈도우의 아무 영역이나 클릭합니다..

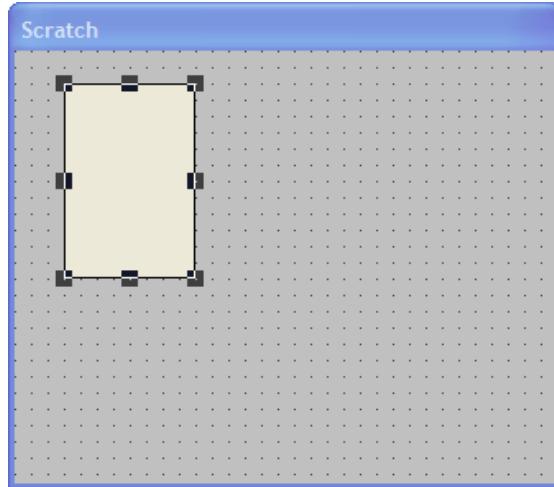


Selecting and Sizing Objects

선택된(selected)이라는 용어는 Windowmaker 의 객체와 그래픽을 수정하는 것에 대한 기본적인 핵심 개념입니다. 객체 위에 커서를 둔 상태에서 마우스 왼쪽 버튼을 클릭함으로 직접 객체를 선택 할 수 있습니다. 윈도우의 공백 영역을 클릭함으로 윈도우에서 현재 선택된 객체를 선택 해제 할 수 있습니다. 객체를 선택한 경우 작은 상자가 경계 주변에 나타납니다. 그 상자는

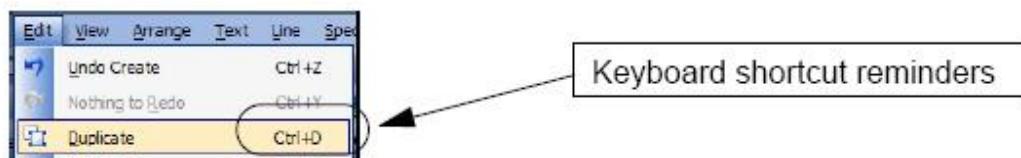
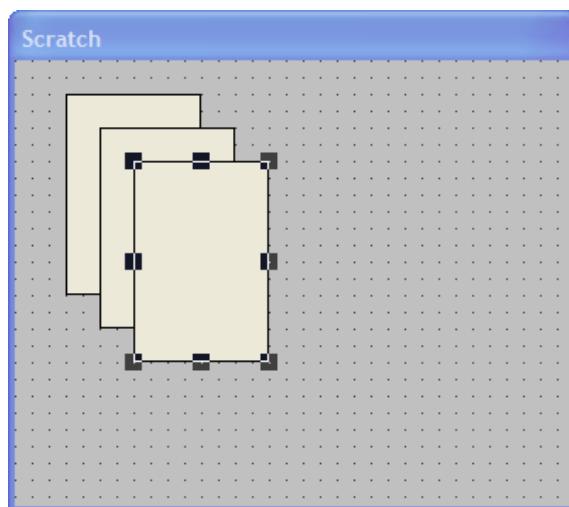
조작 할 수 있으며 객체의 크기를 조정하거나 모양을 변경 할 때 사용됩니다. 일반적으로 실행 명령은 모든 선택된 객체에 적용됩니다.
(당연히 그 명령은 객체에 유효한 것이어야 합니다.)

- a. 사각형안에 어떤 곳을 클릭합니다..



- b. CTRL+D를 두번 누릅니다.(중복을 위한 단축키)

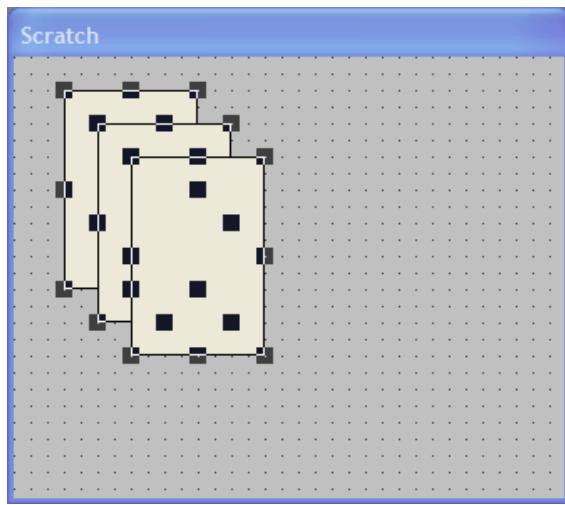
3개의 사각형이 나타나게 됩니다.



Selecting All Objects in a Window

a. 다음 방법중 하나를 사용하여 활성화된 윈도우에서 모든 객체를 선택 할 수 있습니다.

- | Edit / Select All 명령을 클릭합니다.
- | 열려 있는 윈도우의 빈 영역을 오른쪽 클릭하여 **Select All**을 선택합니다..
- | F2 키를 누릅니다.



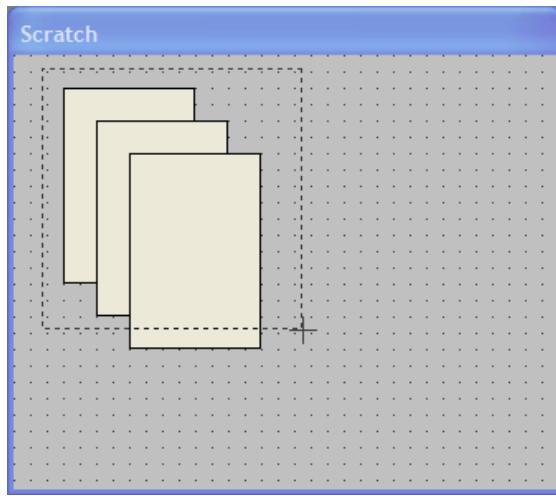
b. 윈도우의 빈 영역에 아무곳이나 클릭하면 객체 선택이 해제됩니다.

Selecting Multiple Objects

a. 여러 객체들을 선택하기 위해, 먼저 객체를 선택합니다. SHIFT키를 누른 상태에서 다른 객체들을 클릭합니다.

Selecting a Group of Objects

- a. 윈도우의 빈 영역에 커서를 이동합니다.
- b. 왼쪽 마우스 버튼을 누른채 마우스를 드래그합니다. 십자 모양의 커서가 점선으로 선택된 사각형 모양으로 나타납니다.



- c. 선택하기 원하는 모든 객체를 선택할 때까지 마우스를 드래그 합니다..
- d. 마우스 버튼을 해제합니다. 사각형 내의 모든 객체는 선택 되어 지게 됩니다..

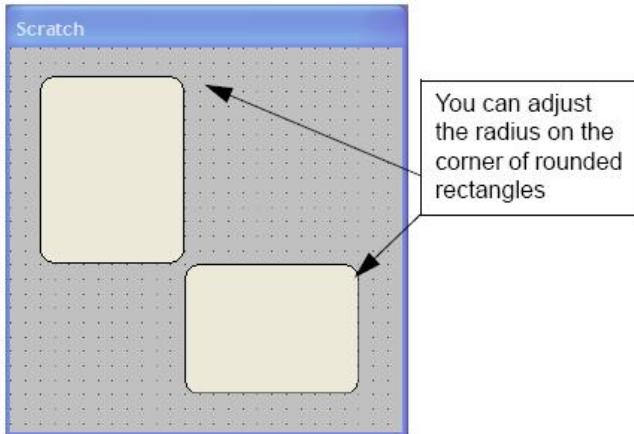
Deselecting a Group of Selected Objects

선택된 사각형을 그리는 동안 **shift** 키를 누르고 있었다면, 선택된 모든 객체는 선택 해제 될 것입니다. 이 기능은 또한 다른 객체 위에서 선택 상자는 시작하는데 사용 될 것입니다..

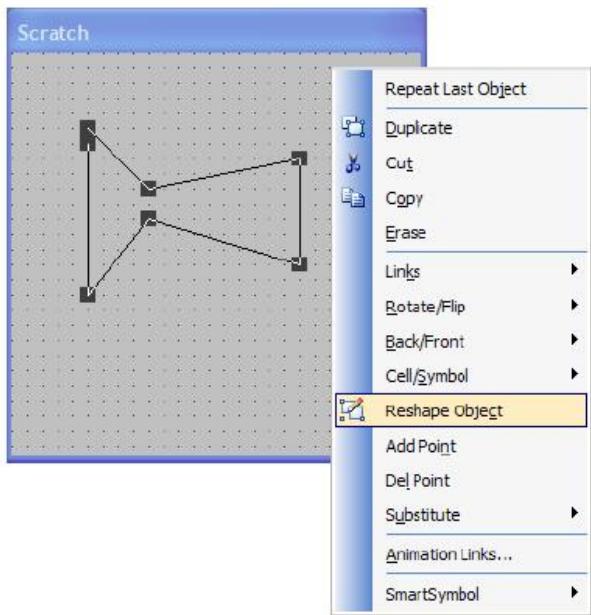
Other Drawing Notes



Rounded Rectangles: 가장 자리가 둥근 사각형 객체의 반경을 조정할 수 있습니다.
SHIFT 키를 누른 채 숫자패드에 (+) 키를 눌러 반지름 값을 증가시키거나 (-) 키를 눌러 감소시킵니다.



Polyline: 윈도우 내에 다양한 장소를 클릭함으로 복잡한 선을 그릴 수 있습니다. 더블 클릭함으로 모양을 완성합니다. 마우스를 해제하면 사이즈를 조정할 수 있게 보여집니다. 각각의 조정은 객체의 외곽을 클릭하기 전까지는 움직일 수 있습니다. 객체외곽을 클릭한 이후에 크기 조정은 전체 객체만 조정할 수 있습니다.(전체크기, 넓이)



객체 외곽 클릭 후 다각선을 수정하기 위해 Edit / Reshape Object 를 선택하거나 다각선을 오른쪽 클릭하고 Reshape Object 또는 객체를 선택 한 후 **Ctrl-R** 을 클릭함으로 객체의 포인트 상태 크기 조정이 보여집니다.



Polygons: 윈도우 내에 다양한 곳을 클릭함으로 복잡한 모양의 다각형을 그릴 수 있습니다. 더블 클릭 함으로 모양을 완성하고 마우스로 해제 하면, 크기 조정 핸들이 보여집니다.

각각의 핸들은 객체의 외곽을 클릭하기 전까지는 움직일 수 없으며, 객체의 외곽을 클릭 한 후에는 전체 객체의 크기만을 조정 할 수 있습니다.(전체 높이,넓이) 객체 외곽 클릭 후 다각형을 수정하기 위해 **Edit / Reshape Object**를 선택하거나 다각형을 오른쪽 클릭하고 **Reshape Object**를 선택하거나 객체를 선택하고 객체의 크기 조정을 보여주기 위해 **Ctrl- R** 을 클릭합니다.

다음 랩에 삼각형 모양 뱀브 이미지를 생성하기 위해 다각형 도구를 사용합니다. 이 매뉴얼의 페이지 **2-22** 페이지에 다른 **toolbar** 버튼들의 그리기와 실습이 있습니다.

Deleting Objects

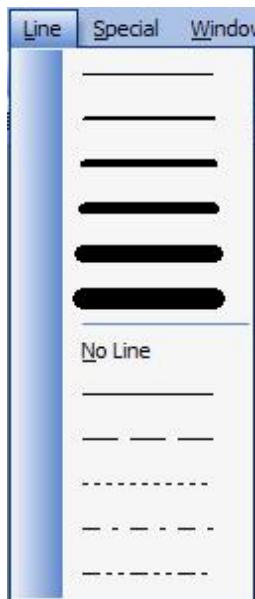
- a. 삭제하기 위해 객체를 선택 합니다.
- b. **Edit / Erase**를 클릭하거나 객체를 선택하고 **DEL**키를 누릅니다.

Note: 삭제된 객체는 윈도우 클립보드로 복사 되지 않습니다.

Lines and Outlines

비트맵 또는 이미지, 다각형, 사각형, 타원의 외각선을 포함하는 선 객체의 넓이 또는 타입을 변경할 수 있습니다. 선 스타일 또는 넓이의 변경은 하나의 객체 또는 함께 선택된 다수의 객체에 한꺼번에 적용 될 수 있습니다.

Line 메뉴는 두 구역으로 나눠집니다. 위 구간은 선 넓이를 포함하고 아래 구간은 선 스타일을 포함합니다.



Applying a Line Command

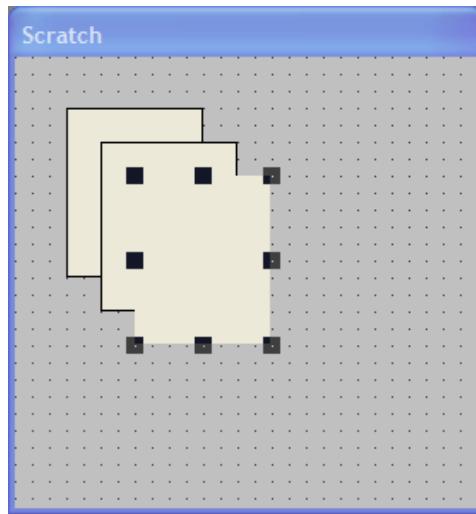
- a. 객체를 선택하고 원하는 **Line** 메뉴에 선 스타일이나 넓이를 클릭합니다.

Tip: 선 스타일, 넓이 또는 색을 선택할 때 객체를 선택하지 않으면, 모든 선과 면, 그리고 **Drawing Toolbar** 안의 문자 도구들에 기본 설정이 적용 될 것입니다.

Note: 굵은 선들의 넓이만 변경 될 수 있으며, 점선들은 단일 픽셀입니다. 넓은 선들은 런타임 시에 더 길게 나타납니다.

Removing an Object's Outline

- a. 하나 또는 그 이상의 사각형 객체를 선택합니다.



- b. 메인 WindowMaker 메뉴로부터 Line / No Line를 선택합니다. 객체의 외곽선이 제거됩니다.

Arranging Objects

WindowMaker는 원도우 안에 객체들을 정렬하기 위한 도구들을 제공합니다. **Arrange Toolbar**은 **Arrange** 메뉴에 속한 명령들을 빠르게 적용하기 위해 사용 될 수 있는 도구들을 포함하고 있습니다.

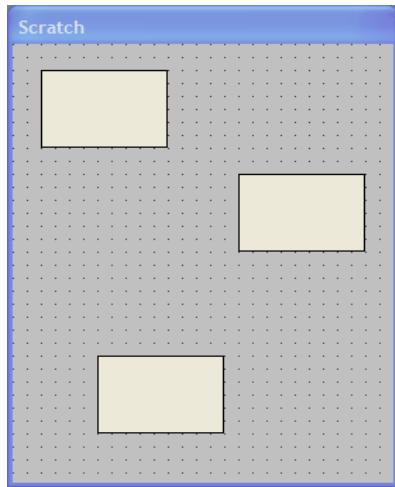


Aligning Objects

객체들은 그들의 상하 좌우 중앙에 맞춰 정렬될 수 있습니다.

Alignment Tools

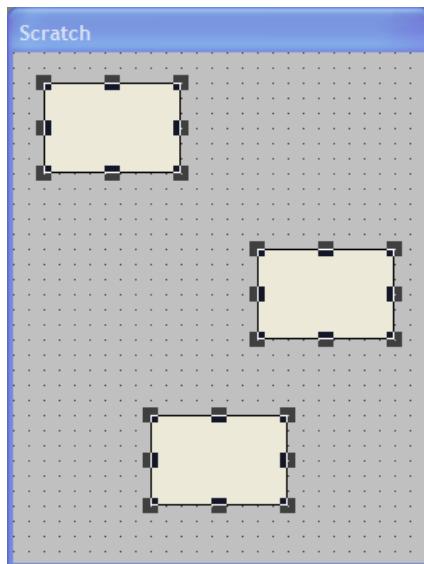
- a. 사각형 객체를 선택하고 다른 위치에 드래그 합니다.
b. 원도우 안에 사각형 객체를 임의로 늘어 놓습니다.



- c. 모든 객체를 선택합니다.
- d. Space Horizontal 나 Space Vertical 정렬 아이콘을 클릭합니다.



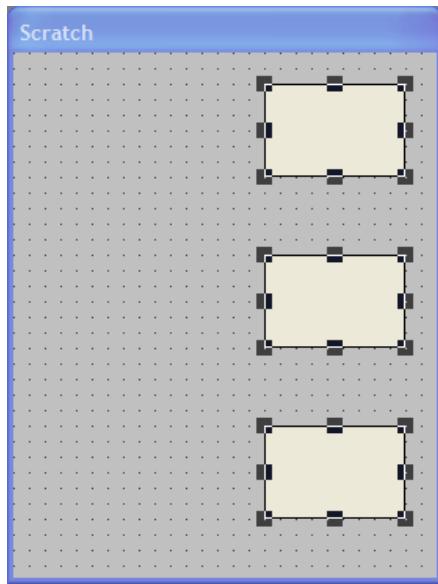
객체들은 수평적 수직적으로 고르게 분산 되어집니다.



- e. Align Left, Align Center, 또는 Align Right 정렬 아이콘을 클릭합니다.



아래의 그림처럼, 오른쪽 정렬 도구는 선택되면 객체의 오른쪽 면에 일렬로 사각형
객체들은 이동됩니다.



Layering Objects

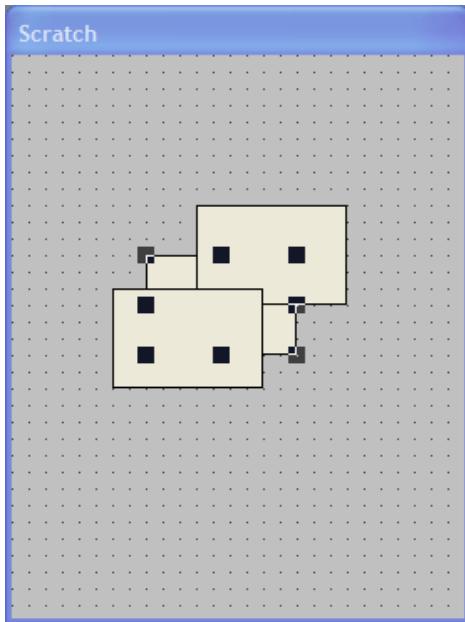
윈도우의 객체들은 각각 서로의 위로, 혹은 아래로 배치될 수 있습니다.

Positioning an Object Behind Another Object

- 사각형 객체를 선택합니다.



- Send to Back 도구 아이콘 . . . 또는 메인 WindowMaker 메뉴로부터 Arrange / Send to Back 을 선택합니다. 선택된 객체는 윈도우에서 선택되지 않은 객체 뒤로 보내집니다.



Positioning an Object in Front of Another Object

- a. 객체를 선택합니다.



- b. Bring to Front 도구 또는 Arrange / Bring to Front를 클릭합니다. 선택된 객체는 윈도우에서 선택되지 않은 객체 뒤로 보내집니다.

Rotating Objects

대부분의 객체(bmp,jpg,pcx,tga, 이미지와 문자 객체)는 WindowMaker Arrange Toolbar 를 이용해 회전되어질 수 있습니다. 객체들은 90 도씩 증가하면서 시계방향이나 시계 반대방향으로 회전됩니다.

Note: WindowMaker 안에서 객체들의 회전은 RUNTIME 시에 **rotation** 애니메이션 회전 객체와 관련이 없습니다. 문자 객체들은 사킬 수 없으며. 비트맵 또는 이미지들은 애니메이션 링크에 의해 회전될 수 있습니다.

Rotating Objects and Text using the Arrange Toolbar

- a. 객체나 문자를 선택합니다.



- b. Arrange Toolbar에 Rotate CounterClockwise **tool bar** 버튼 또는 Rotate



Clockwise **tool bar** 버튼 를 선택합니다. 선택된 객체는 선택 방향에서 **90도**로 회전됩니다. 객체 회전은 Arrange / Rotate Clockwise 나 CounterClockwise 선택에 의해 메인 메뉴로부터 실행됩니다.

Tip: 객체를 **180** 도로 회전하기 위해 이 과정을 반복합니다.

Mirroring and Flipping Objects

대부분의 WindowMaker 객체들은 수평이나 수직적으로 반전되어 질 수 있습니다. (bmp, JPEG, PCX, TGA 파일, 문자는 이 기능을 사용할 수 없습니다) 객체가 반전될 때, 그것은 수평, 수직으로 반사된 이미지로 변형됩니다.

- a. 객체를 선택합니다.



- b. Flip Horizontal 도구 버튼 또는 Arrange Toolbar에 Flip Vertical 도구 또는 메인 메뉴로부터 Arrange / Flip Horizontal(또는 Flip Vertical)를 클릭합니다. 선택된 객체가 반전될 것입니다.

Creating Symbols and Cells

여러 객체들은 **Symbol** 과 **Cell** 이라는 두 가지 방식으로 조합될 수 있습니다.

- | **Symbols:** **Symbol** 은 여러 개의 객체를 하나의 객체로 조합합니다. 선택된 객체 중의 하나에 애니메이션이 설정 되어있다면, 그 링크는 새로 만들어진 **Symbol** 에 설정이 됩니다.
- | **Cells:** **Cell**은 **Cell** 안에 합쳐진 각각의 객체에 애니메이션 링크를 설정할 수 있습니다. (그러나 **Cell**로 묶인 객체는 애니메이션링크를 지원하지 않습니다.) 여러 개의 **Cell**은 하나의 **Cell**로 다시 묶일 수 있으며, **Cell** 안의 각 객체를 특별한 관계로 고정 시켜 유지합니다. **Cell** 은 회전 될 수 없고, **Symbol**로 만들어질 수 있습니다. 또한 **Cell** 안의 Text font, size, 라인의 넓이 등을 변경 할 수 없습니다.

Cell	Symbol
<p>개별적으로 애니메이션 링크를 가진 객체나 심볼 그룹, 비트맵, 버튼, 트랜스, 위저드와 같은 객체들은 셀에 포함 될 수 없습니다.</p> <p>다른 셀과는 포함 될 수 있습니다.</p> <p>크기를 조절할 수 없습니다.</p>	<p>하나의 애니메이션 링크를 가진 개별적 객체그룹입니다.</p> <p>심볼을 생성해서 애니메이션 링크를 지정 할 수 있습니다.</p> <p>단순히 간단한 객체만 포함합니다.</p> <p>크기 조절이 가능합니다.</p>

Note : 선택된 객체들이 하나 이상의 애니메이션 링크를 가진다면 **Symbol** 은 만들어 질 수 없습니다. 또한 두 개의 심볼을 묶어서 하나의 심볼을 만든다면, 원래 심볼의 성질은 잃게 됩니다. 그러므로 새 심볼을 해체한다면 그것은 원래의 심볼에 대한 개개의 구성요소를 잃어버리게 됩니다.

Creating a Symbol or Cell

- a. **Symbol** 이나 **Cell**로 만들 객체들을 선택합니다.



- b. **Make Cell tool bar** 버튼 또는 **Make Symbol tool bar** 버튼 을 클릭합니다.

Tip: Cell 을 더블 클릭하면, Substitute Tagname 대화상자가 나타납니다. (Symbol 객체는 애니메이션 대화상자를 나타냅니다.)

Breaking a Symbol or Cell

- 심볼이나 셀을 선택합니다.

b. Break Cell **tool bar** 버튼 이나 Break Symbol **tool bar** 버튼 을 클릭합니다.

Tip: Symbol 과 Cell 을 빠르게 생성하기 위해, 객체를 선택하고 오른쪽 마우스 클릭 명령을 사용할 수 있습니다.

또한, Symbol 과 Cell 을 분리하는 것도 오른쪽 마우스 클릭을 사용할 수 있습니다.

Note: Symbol 에 애니메이션 링크를 설정 하면, 자동적으로 링크는 link paste buffer 에 저장됩니다.

Text Objects

폰트, 폰트 스타일, 폰트 사이즈, 자리맞춤 그리고 문자 객체의 회전은 WindowMaker에서 수정 할 수 있습니다. 또한 문자 객체들은 90~ 360 도로 회전 할 수 있습니다.

Note: Orientation 애니메이션 링크는 문자 객체에 적용될 수 없습니다. 그래서 문자 객체들은 태그의 값에 따라 WindowViewer 안에서 회전 할 수 없습니다.

Format Toolbar 는 Text 메뉴에 속한 명령들을 선택된 객체에 적용할 수 있는 tool 로 구성되어 있습니다.



Formatting Text Objects

WindowMaker 안의 모든 Text 명령은 하나 또는 다중 선택에 의해 한번에 변경될 수 있으며,

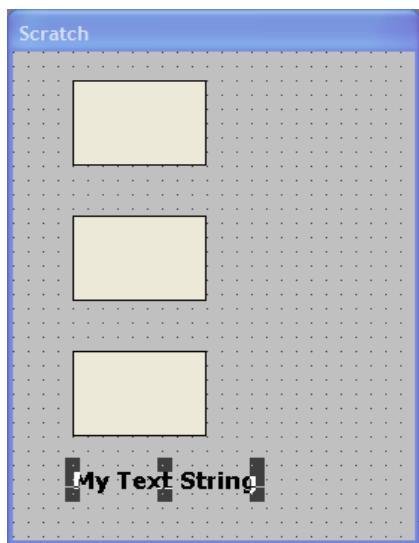
Text 메뉴가 실행될 때, 선택된 객체가 없다면, 자동적으로 Format Toolbar 나 Draw Object Toolbar의 기본 설정 값으로 반영됩니다.

문자 자리맞춤 속성설정은 문자 값을 동적인 값을 표시하는데 특히 중요합니다. 자리맞춤은 Runtime 시에 길이 변화의 필드들이 어떻게 보여질 것인지를 결정합니다.

Creating a Text Object



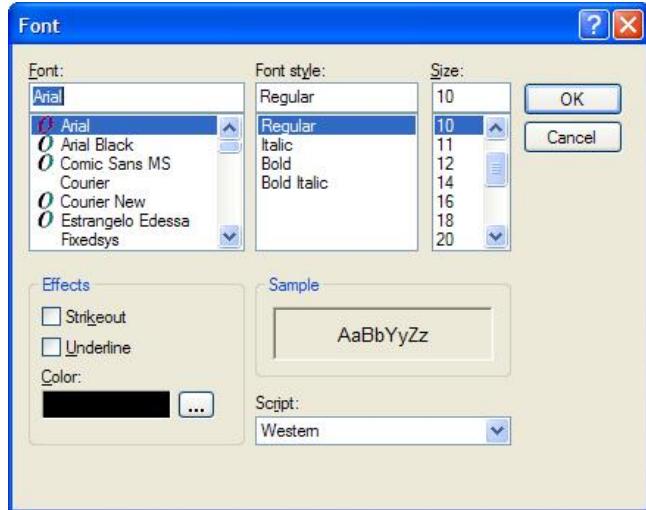
- a. Drawing Toolbar에 Text 도구를 클릭합니다
- b. 윈도우 안의 문자입력 할 위치에 커서를 놓습니다.
- c. 문자열을 입력합니다.
- d. 문자를 클릭합니다. 선택된 문자를 설정 할 수 있습니다.



Changing Font, Font Style, and Font Size



- a. 변경 하기 원하는 문자를 선택하고 Format Toolbar에 폰트 도구를 클릭합니다. 윈도우 표준 폰트 대화 상자가 나타납니다.

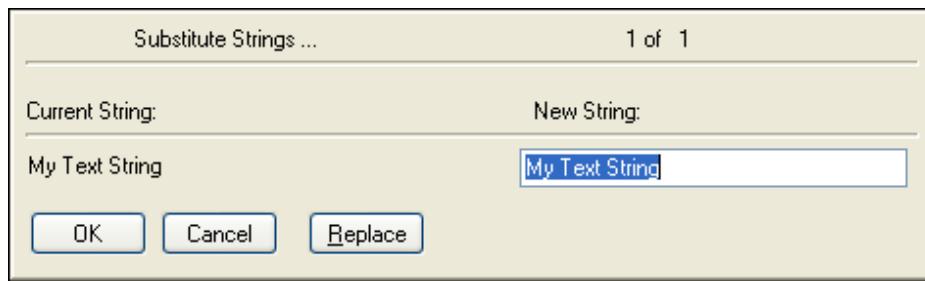


- b. Font 목록(Font 필드 안에 폰트 이름이 나타날 것입니다.)으로부터 원하는 폰트를 선택합니다. 폰트 크기가 선택 될 때, 선택된 스타일과 크기가 Sample 필드 안에 나타날 것입니다.(위 예제를 보십시오)
- c. OK를 클릭합니다.

Note: 폰트의 크기포인트는 선택된 폰트에 이용 가능한 폰트 크기의 범위에 따라 증가하거나 감소 합니다. 예로, 시스템 폰트는 크기조정을 할 수 없습니다. 크기를 변경하기 전에 Arial 와 같은 윈도우 True-Type 폰트를 선택합니다.

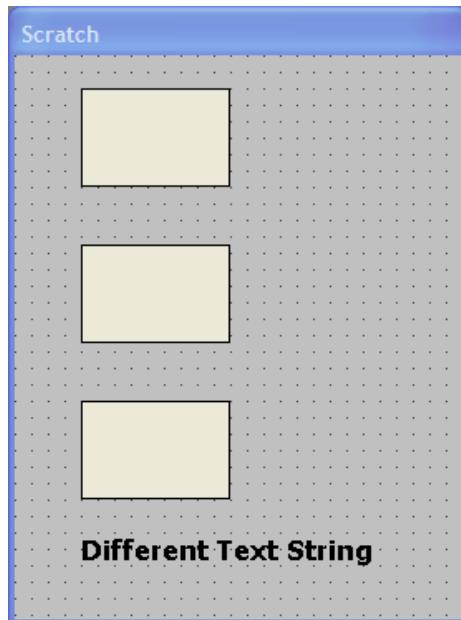
Editing Text Objects

- a. 문자 객체를 선택합니다.
- b. Special / Substitute Strings 또는 문자 객체 오른쪽 클릭 그리고 Substitute Strings 선택 또는 Ctrl+L키를 누릅니다. Substitute Strings 대화상자가 나타납니다.



- c. New String 상자 안에 새로운 문자열을 입력합니다.
- d. OK를 클릭합니다.

윈도우 안에 새로운 문자열이 나타납니다.

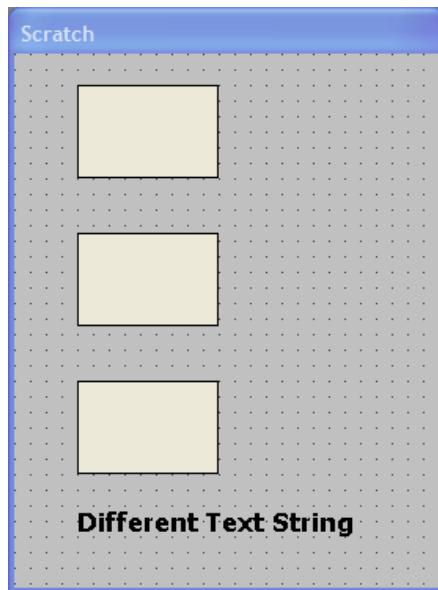


Tip: 또한 심볼이나 셀 안에 포함된 문자열에 Substitute String 명령을 사용하거나 버튼의 라벨을 변경하기 위해 사용 할 수 있습니다.

Replacing a Portion of a Text Object

문자 객체의 문자 일부가 변경되면, 선택된 문자 객체 중 문자가 같은 것을 전체적으로 변경 할 수 있습니다.

- 여러 개의 문자 객체들을 선택합니다.



- Special / Substitute Strings를 선택합니다.

Substitute Strings 대화 상자가 나타납니다.



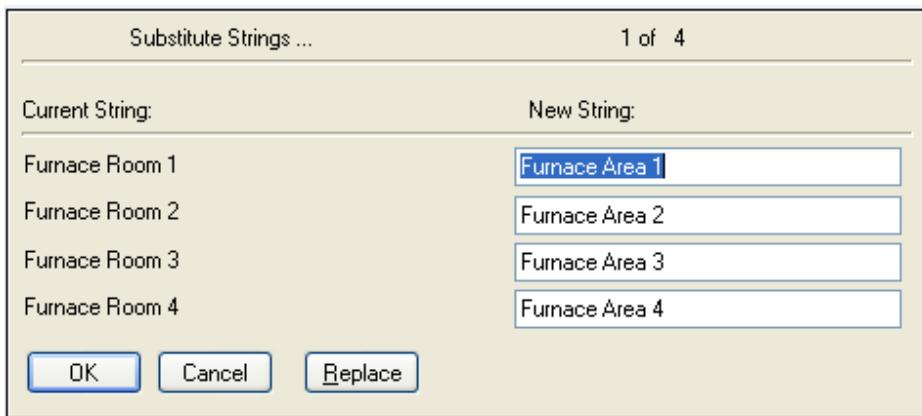
- Replace를 클릭합니다.

Replace Text 대화 상자가 나타납니다.



- d. Old Text 필드에 교체를 원하는 문자열의 부분을 입력합니다.
- e. New Text 필드에 교체되는 문자를 입력합니다.
- f. OK를 클릭합니다.

선택된 문자열에 변경 모드를 보여주는 Substitute Strings 대화 상자가 나타납니다.

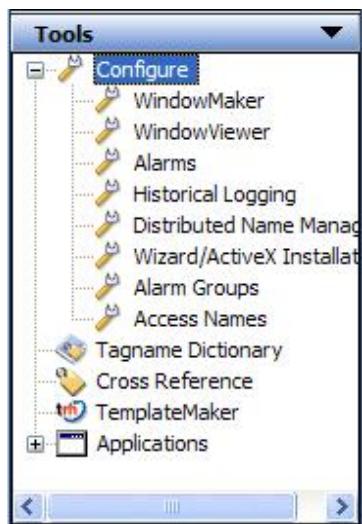


- g. OK를 클릭합니다. 선택된 문자열이 자동적으로 수정됩니다.

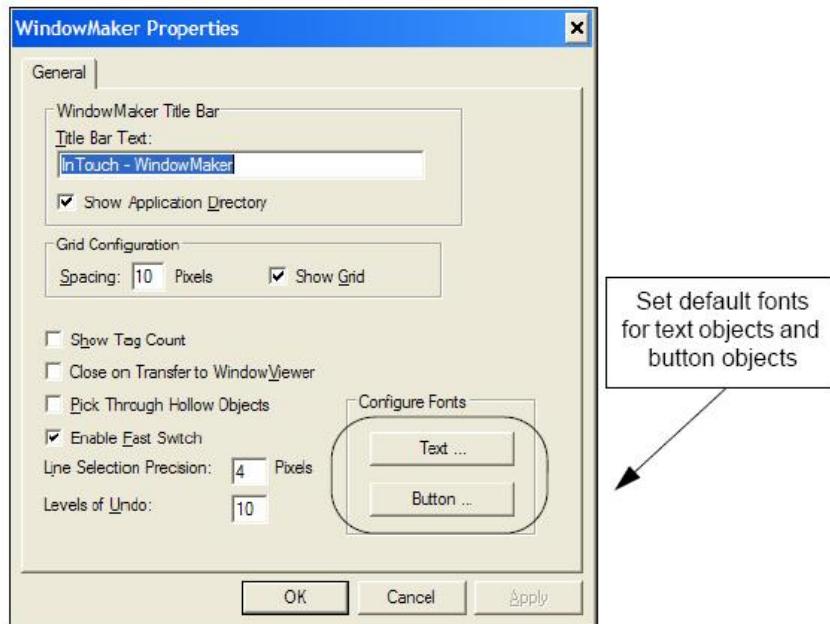
Configuring Default Fonts

WindowMaker에서는 문자나 버튼의 기본 폰트를 설정 해 줄 수 있습니다. 설정된 기본 폰트는 문자객체나 버튼을 생성할 때 적용됩니다.

- a. Application Explorer에서 Configure / WindowMaker를 더블 클릭합니다.

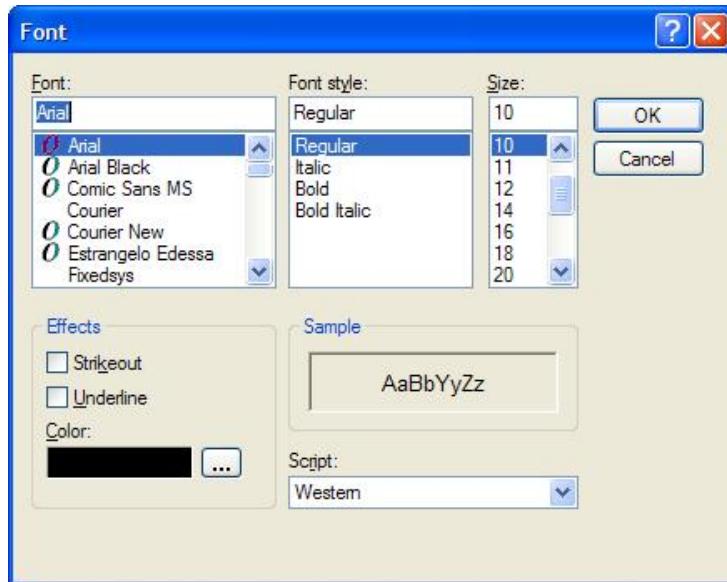


WindowMaker Properties 대화상자가 나타납니다.



b. Text 또는 Button을 클릭합니다.

Font 윈도우가 나타납니다.



- c. 크기 조정할 수 있는 기본 폰트를 할당하기 위해 윈도우 True-Type 폰트를 선택합니다.
- d. 변경을 저장하기 위해 OK를 선택하거나 WindowMaker Properties 대화상자를 종료하기 위해 Cancel를 선택합니다.

Images and Bitmaps

다른 윈도우 프로그램들 안에서 생성된 이미지, 화면 캡쳐, .Jpg, pcx 그리고 tga 파일형태와 같은 그래픽 이미지를 WindowMaker 안에 비트맵 항목으로 **import** 하여 사용할 수 있습니다.

주의 할 점

- | **WindowMaker**에서 **Bitmap**은 단일 객체로 인식됩니다.
- | **Symbol** 안에는 비트맵을 포함 시킬 수 없습니다.
- | **Cell** 안에는 비트맵들을 포함 시킬 수 있습니다.
- | 비트맵 객체에 투명색을 설정하여, 다른 객체 위에 놓을 수 있습니다.
: 그러한 작업에 의해, 투명색 필터가 적용되는 지역을 통해 윈도우 배경색이나 어떤 객체 뒤로 비트맵이 보여 질것입니다.(오직 하나의 투명색이 비트맵에 의해 사용될 수 있습니다.)
- | WindowMaker를 종료하지 않고 Microsoft Paint 프로그램을 사용하여 윈도우의 비트맵 이미지를 수정할 수 있습니다.
: Microsoft Paint를 설치하고 비트맵을 수정하기 위해 비트맵을 오른쪽마우스 클릭 후 Edit Bitmap를 선택합니다.

Note: Paint 안에 비트맵을 수정하는 동안 WindowMaker를 사용 할 수 없습니다.

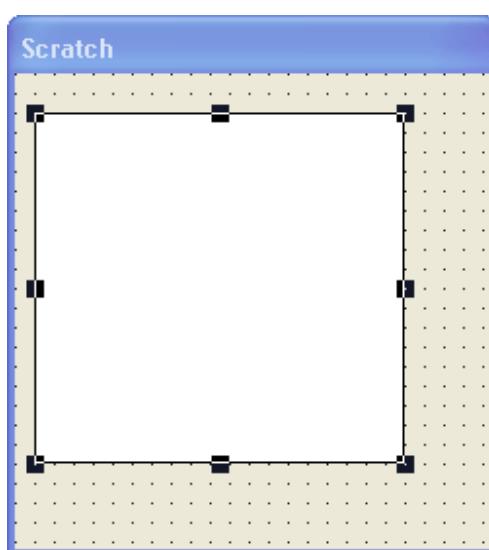
WindowMaker로 전환하기 위해 Microsoft® Paint를 종료합니다. WindowMaker에서 변경사항은 비트맵 객체에 저장되며, **import** 된 원래 비트맵 파일은 저장되지 않습니다.

Importing a Bitmap, JPEG, PCX, or TGA File



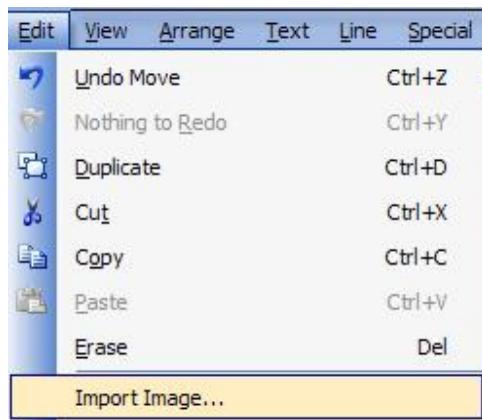
- a. Bitmap 도구를 클릭합니다. 커서가 +로 전환됩니다.
- b. 윈도우에 비트맵 컨테이너가 그려집니다.

아래의 그림은 Scratch 윈도우 안에 비트맵을 보여줍니다.

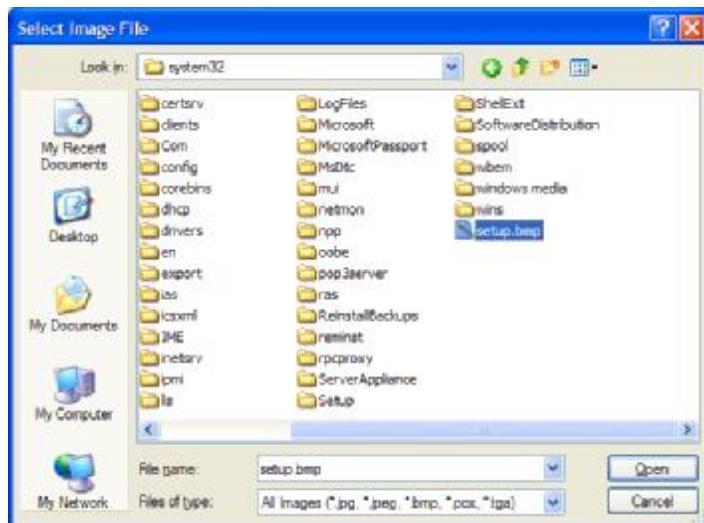


- c. 비트맵 컨테이너를 선택한 후, 오른쪽 마우스 클릭을 합니다.

드롭다운 메뉴가 나타납니다.

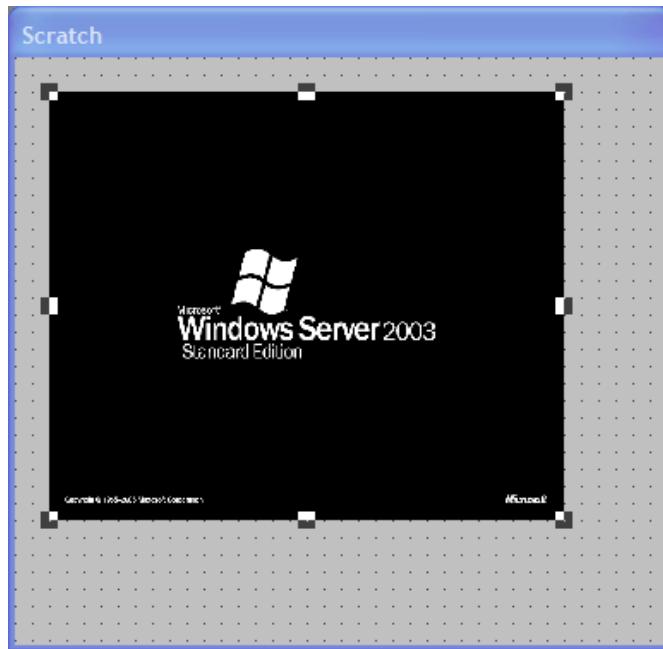


- d. 컨테이너 안에 이미지를 붙이기 위해 Import Image를 선택합니다.
Select Image File 대화 상자가 나타납니다.



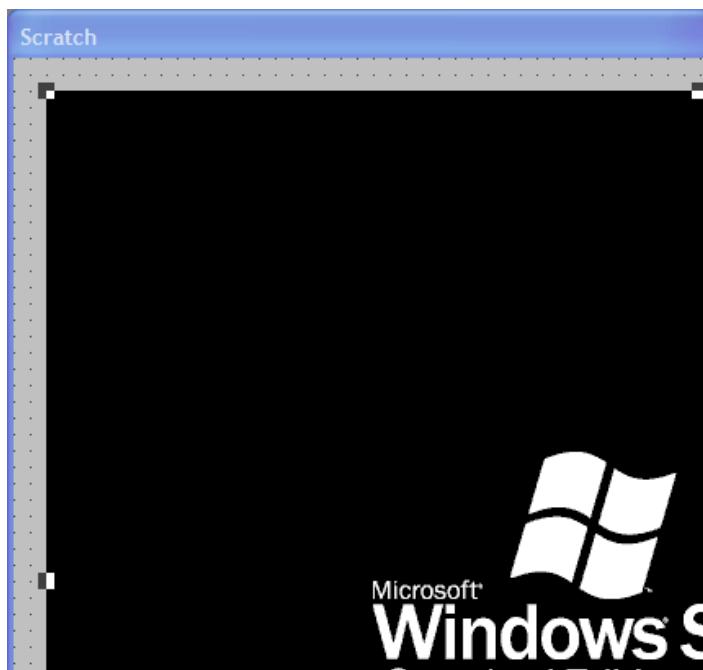
- e. 파일을 선택합니다. 이전 그림에서 c:드라이브에 WINNT\System32 폴더로부터 setup.bmp 이미지가 선택됩니다.
f. Open를 클릭합니다.

비트맵 객체에 이미지가 붙여집니다.



- g. 비트맵의 원래 크기로 비트맵을 전환하기 위해 비트맵에서 오른쪽 마우스 클릭 후, **Bitmap - Original Size**를 선택합니다.

비트맵의 원래 크기에 비트맵이 다시 그려집니다.

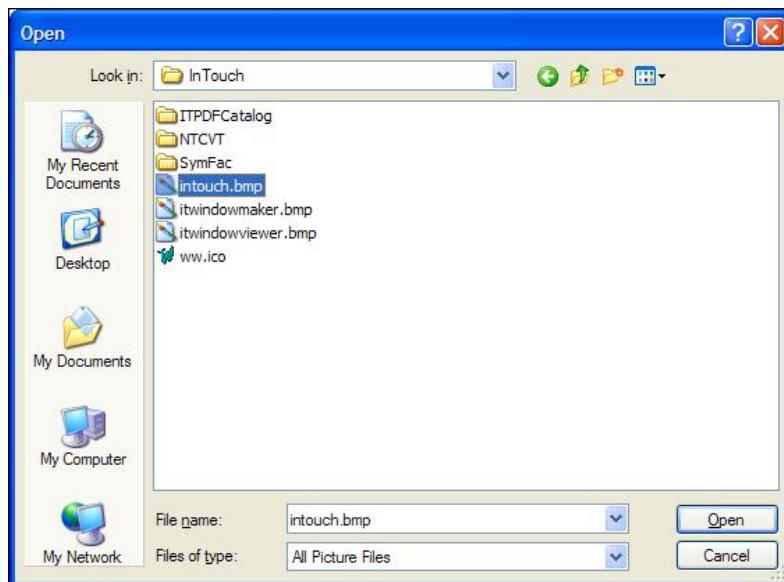


- h. 비트맵 객체를 삭제합니다.

Pasting a Bitmap from the Windows Clipboard

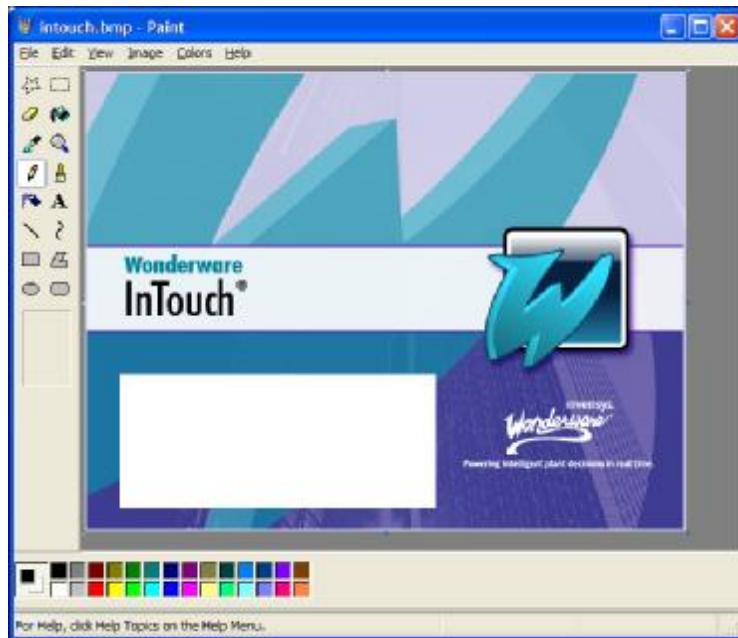
InTouch 비트맵 객체 안에 윈도우 클립보드로부터 그래픽 파일들은 직접 복사 될 수 있습니다.
이 예제는 InTouch 화면을 사용할 것입니다.

- a. Microsoft Paint (Programs / Accessories / Paint)를 엽니다.
- b. Paint 메뉴로부터 File / Open를 선택합니다.
- c. Program Files / Wonderware / InTouch안에 InTouch.bmp를 검색합니다.



- d. Open를 클릭합니다.

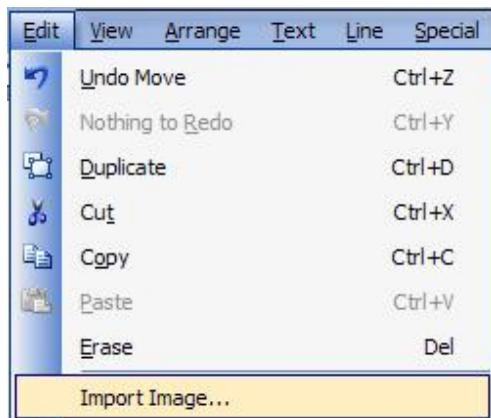
Paint 윈도우 안에 이미지와 **size handle** 이 나타납니다.



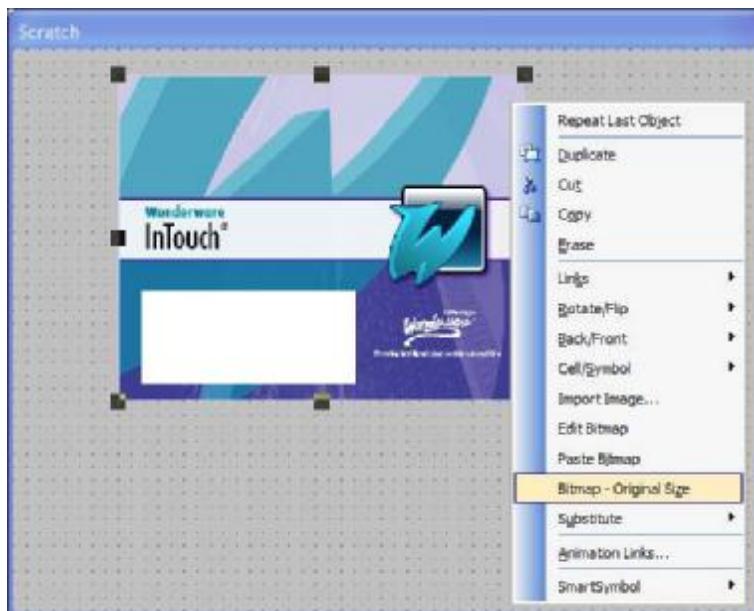
- e. 그래픽 이미지를 선택하기 위해 **Ctrl+A**를 누릅니다.
- f. 윈도우 클립보드에 이미지를 복사하기 위해 **Ctrl+C**를 누릅니다.
- g. 비트맵을 저장하고 WindowMaker로 전환하기 위해 **File / Exit**를 클릭합니다.
- h. Scratch윈도우 안에 비트맵 컨테이너를 삽입하기 위해 **Bitmap** 도구 를 사용합니다.
- i. 비트맵 컨테이너를 선택합니다.
- j. 오른쪽 마우스 클릭 후 드롭 다운 메뉴로부터 **Paste Bitmap**을 선택합니다. 윈도우 클립보드 안에 놓여진 비트맵은 비트맵 컨테이너 안으로 붙여집니다.

Note: Paste Bitmap 명령은 윈도우 클립보드 안에 이미지가 있을 때만 이용 가능합니다.

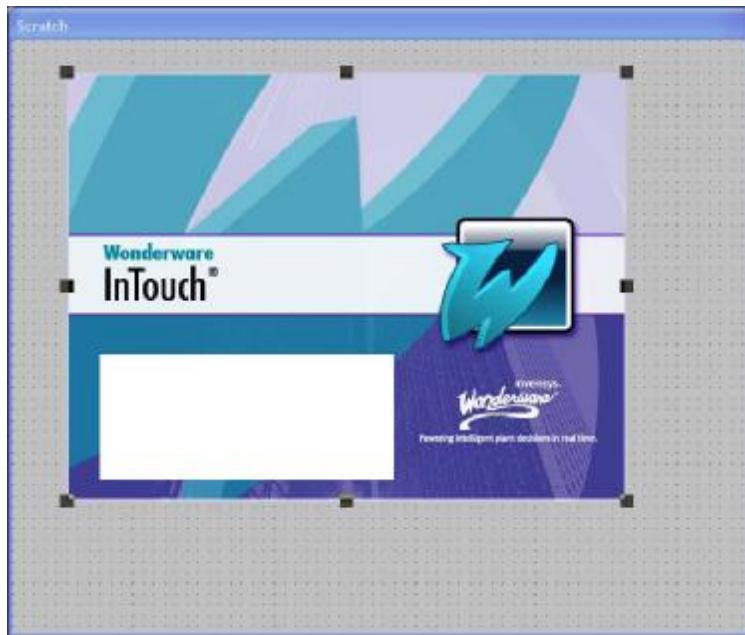
비트맵 복사, 붙이기 대신에, 저장된 비트맵 파일을 검색하기 위해 **Edit** 메뉴(밑의 그림과 같이)로부터 **Import Image** 옵션을 사용 할 수 있습니다.



Scratch 원도우에 비트맵이 가져와집니다. 그러나 이미지에 비트맵 컨테이너의 크기는 적절하지 않습니다.



- k. 드롭 다운 메뉴로부터 Bitmap - Original Size 옵션을 선택하고 비트맵 객체를 오른쪽 클릭합니다.



이미지가 원래 크기로 보여집니다

Note: 비트맵 크기를 마우스를 사용하여 수동으로 조정 할 수 있습니다.

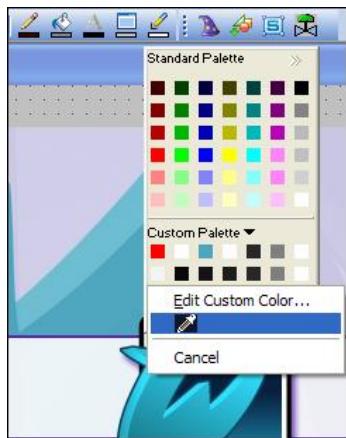
Creating a Transparent Bitmap

비트맵이나 이미지는 투명색을 정의 함으로써 그래픽이나 배경색을 비추어 보여 줄 수 있습니다. 투명색을 정의함으로써 비트맵 이미지 뒤의 그래픽 객체나 배경이 설정된 투명색이 사용된 부분에 보여지게 될 것입니다.

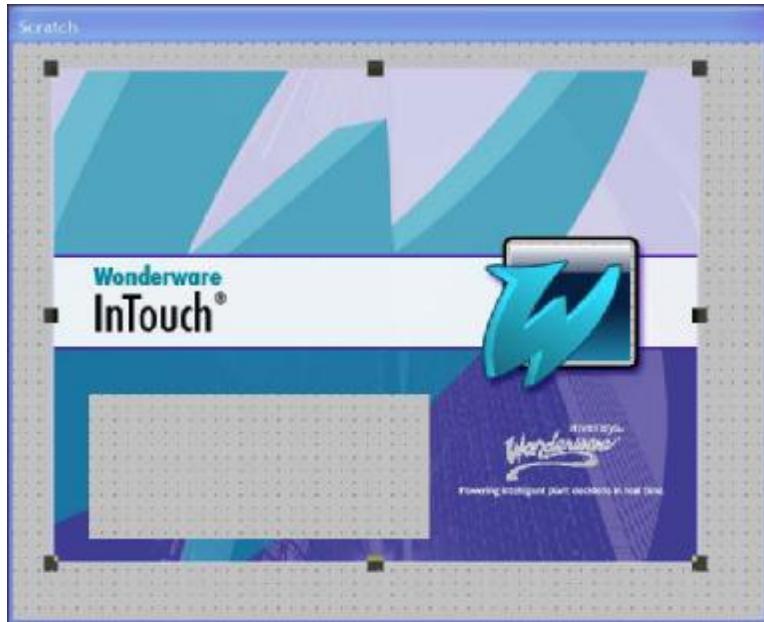
- a. 비트맵 객체를 선택합니다.



- b. Format tool bar에서 Transparent Color도구를 클릭합니다. Color Palette가 나타납니다.
- c. Color palette의 하단에 Custom Palette 부분 안에 빈사각형에서 오른쪽 마우스 클릭을 합니다.



- d. Blotter 도구  를 선택합니다.
- e. InTouch 그래픽 안에 하얀 상자를 클릭하고 blotter-shaped 커서를 드래그합니다.
Transparent Custom Color Palette안에 선택된 상자에 색을 복사합니다.
- f. 투명으로 만들기 위해 색을 복사한 곳에 Custom Color square를 클릭합니다.
blotter도구와 함께 선택된 이미지의 부분은 투명하게 보여질 것입니다.



Note: 오직 하나의 투명색만이 비트맵에 적용 될 수 있습니다.

Editing Bitmap Images

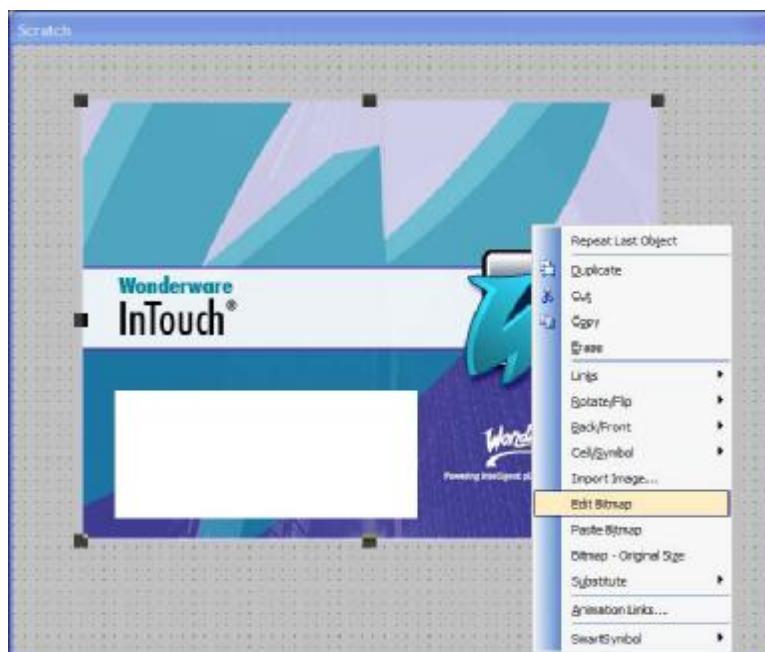
WindowMaker을 종료하지 않고 Microsoft Paint 프로그램을 사용하여 윈도우의 비트맵 이미지를 수정할 수 있습니다.

Microsoft Paint를 설치하고 비트맵을 수정하기 위해 비트맵을 오른쪽마우스 클릭 후 Edit Bitmap을 선택합니다.

Note: 페인트에서 비트맵이 수정되는 동안 WindowMaker는 사용 할 수 없습니다.

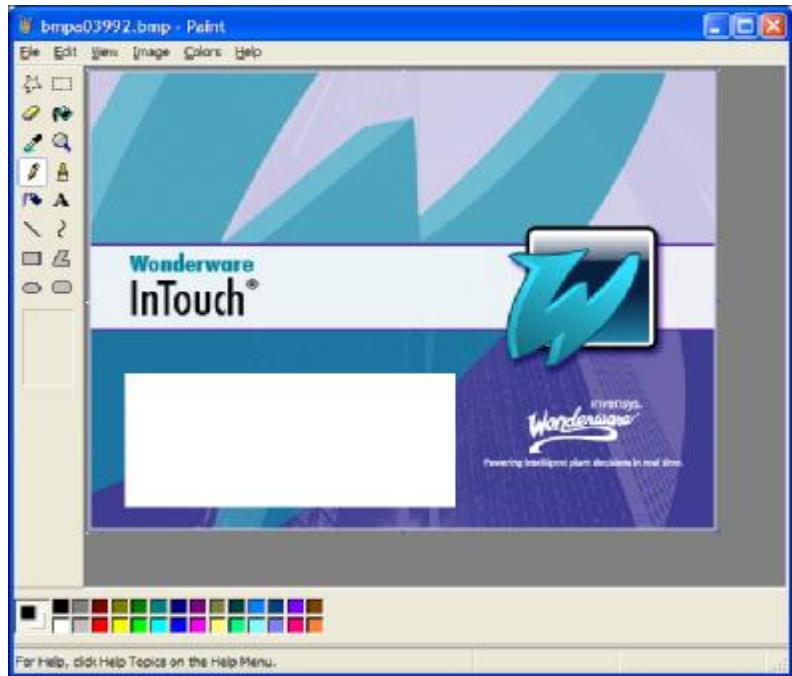
WindowMaker로 전환하기 위해 Microsoft Paint를 종료하고 하드 디스크에 이미지를 저장합니다.

- a. 위에서 생성된 투명색과 함께 비트맵 이미지를 삭제합니다.
- b. Scratch 윈도우 안에 새로운 비트맵 컨테이너를 놓습니다.
- c. Program Files / Wonderware / InTouch .bmp 파일을 다시 가져옵니다.
- d. Scratch 윈도우에서 Bitmap - Original Size와 비트맵 중앙을 선택합니다.

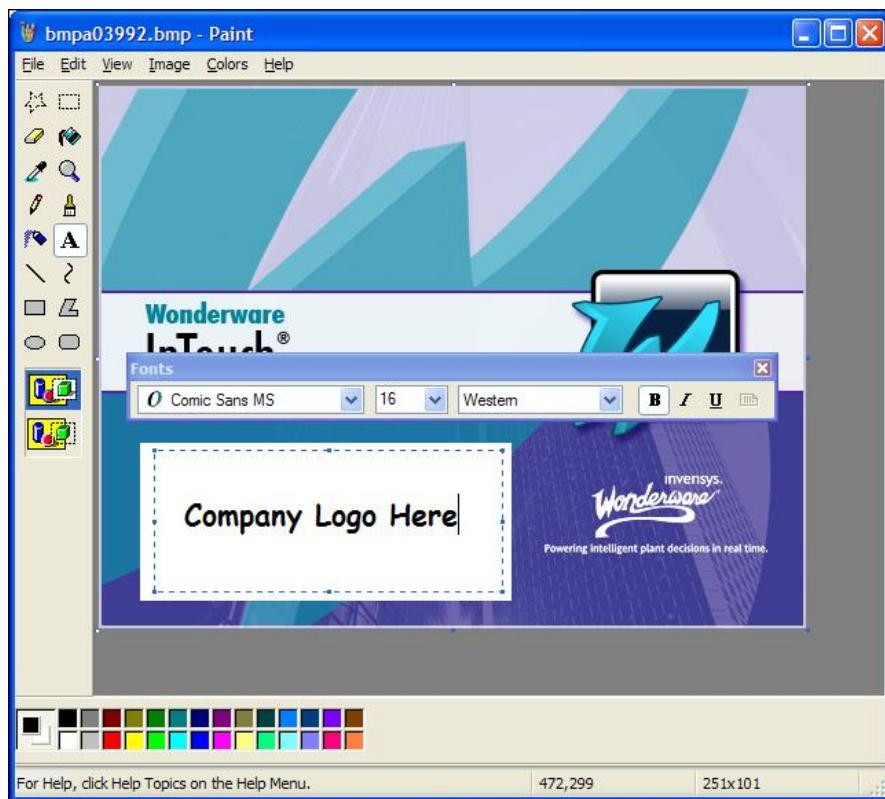


- e. Edit / Edit Bitmap를 선택하거나 비트맵 이미지 오른쪽 클릭을 하고 Edit Bitmap을 선택합니다.

Microsoft Paint가 실행되고, 임시 파일명이 자동적으로 할당됩니다.



Tip: Microsoft Paint의 다양한 기능을 사용하여 비트맵 이미지를 편집할 수 있습니다.

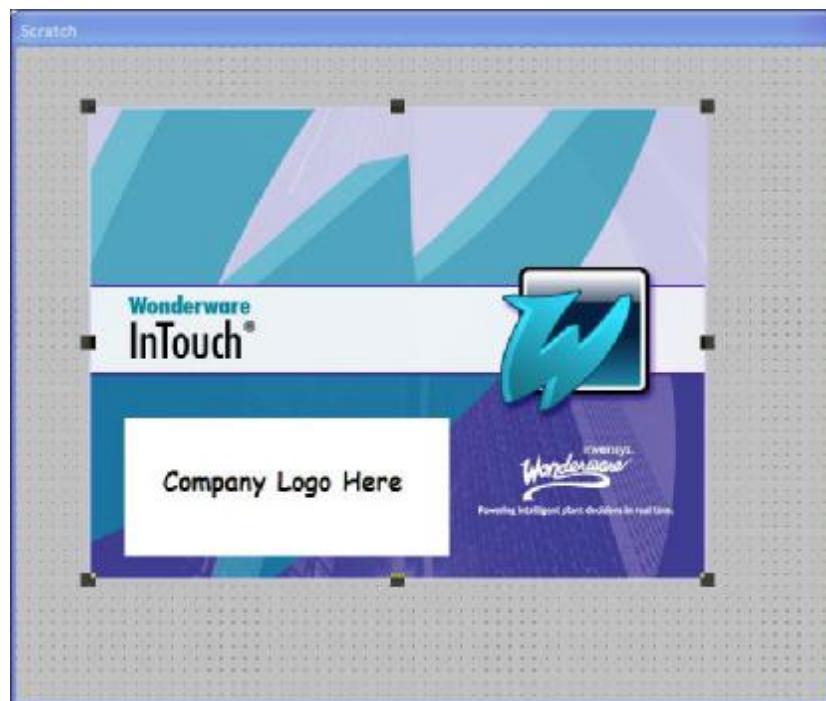


- f. 0] 예제에서, Company Logo Here 문자를 추가하고 폰트 속성들을 수정합니다.
- g. Microsoft Paint를 닫습니다. 임시 파일을 저장하기 위해 메세지창이 뜹니다.



- h. Yes를 클릭합니다. WindowMaker에서는 비트맵 객체에 변경사항들을 저장하고, 가져온 원래 비트맵 파일은 저장되지 않습니다.
- i. WindowMaker로 전환합니다.

윈도우는 다음 그림과 같습니다.



Tip: 원래 흰색 투명 블록을 만들기 위해 비트맵을 수정하고, 로고안의 어떤곳에도 사용되지 않은 유일한 색을 흰색 상자에 채우고 InTouch 윈도우안에 이 유일한 색을 투명색처럼 사용합니다.

Wizards

Wizard 는 미리 그려져 있는 그래픽 컴포넌트로써, WindowMaker에서 원도우에 가져다 쓸 수 있습니다.

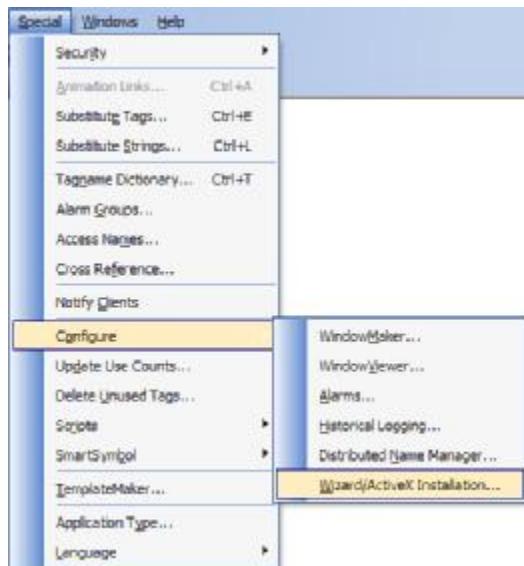
Wizard는 사용과 구성이 쉽기 때문에, Application의 개발 시간을 많이 절약할 수 있습니다. Wizard를 사용함으로써 개발자는 그림 그리는 시간, 설정 값과 애니메이션을 구성하는 시간을 줄일 수 있습니다.

예로, Slider Wizard의 경우, 태그 대입, 최대 최소값 입력, 슬라이더 라벨입력, 색상과 같은 옵션을 쉽게 구성 할 수 있습니다.

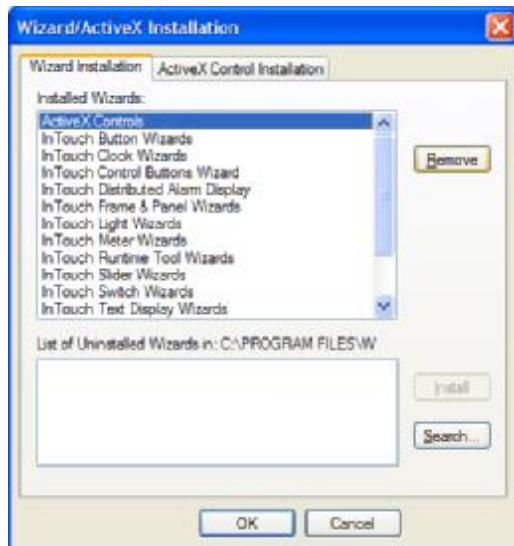
Installing or Removing Wizards

Wizard의 설치와 제거는 InTouch 윈도우 안에 그것을 사용하는 것과 상관없지만, 사용할 수 있는 Wizard를 Wizard 목록 안에 포함해야 합니다.

- a. WindowMaker 메뉴로부터 **Special / Configure / Wizard/ActiveX Installation**을 선택합니다.

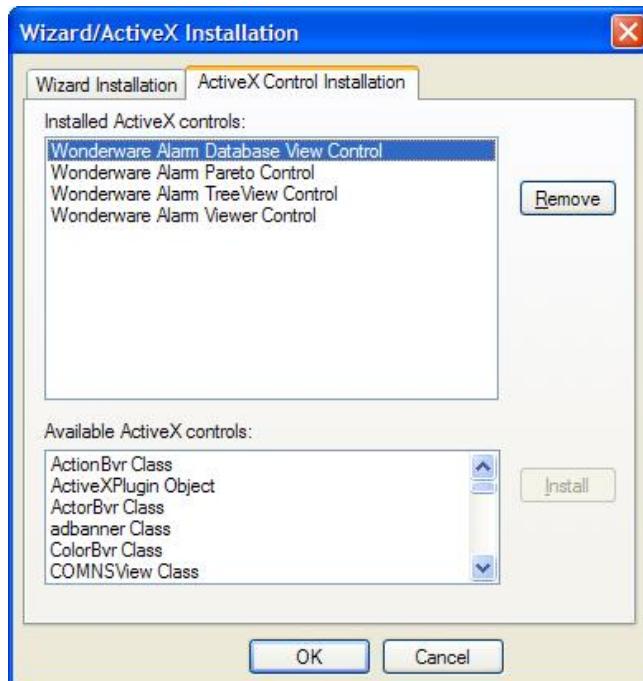


Wizard/ActiveX Installation 대화상자와 마법사 설치 속성 시트가 활성화 됩니다.



위의 팬은 이용 가능한 Wizard 객체들의 목록을 아래 팬은 제거된 객체들을 포함합니다.
설치된 Wizard를 목록에서 제거하기 위해 Remove버튼을 클릭합니다. 제거된 객체는
다시 선택하여 Install 함으로써 다시 설치할 수 있습니다.

- b. ActiveX Control Installation 탭을 선택합니다. 아래 팬에 이용 가능한 목록이 있습니다.

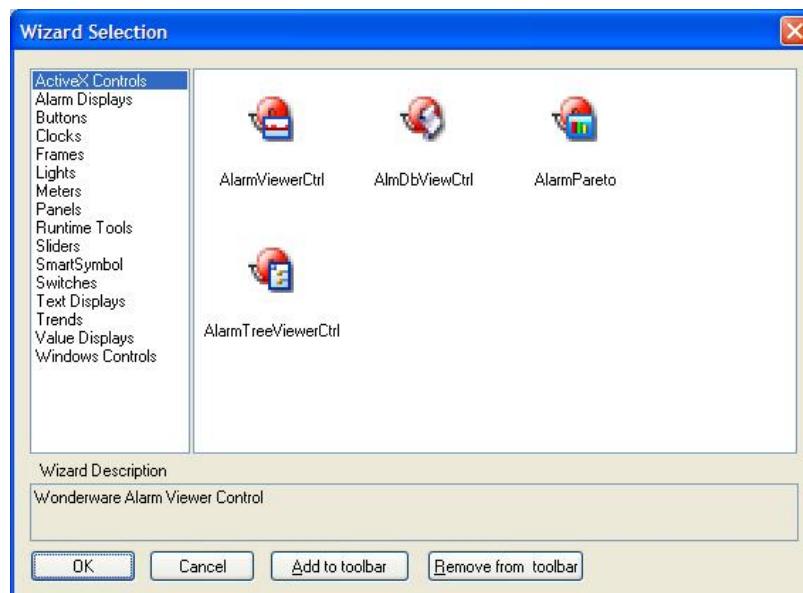


- c. ActiveX Control를 제거하기 위해 Remove 버튼을 이용하고, 이용 가능한 ActiveX Control들의 목록에 다시 이동 시킵니다.(이 예제에서는 어떠한 Wizard도 제거하지 않습니다)
- d. WindowMaker로 전환하기 위해 Cancel을 클릭합니다.

Placing Wizard Objects in a Window



- a. Wizard Hat toolbar 아이콘 을 클릭합니다.
Wizard Selection 대화 상자가 나타납니다.

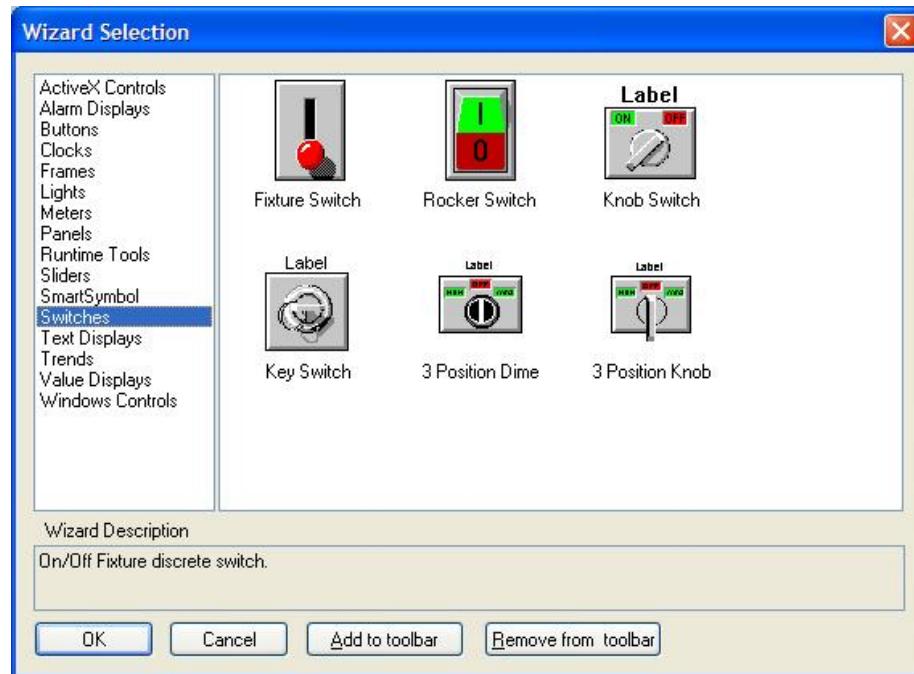


Tip: Wizards/ActiveX Toolbar에 Wizard를 추가하기 위해 Add to toolbar를 클릭합니다.

Wizards/ActiveX Toolbar에 Wizard를 추가하기 위해 Wizard를 선택 할 수 있고 언제라도 열린 윈도우에 붙여질 수 있습니다.

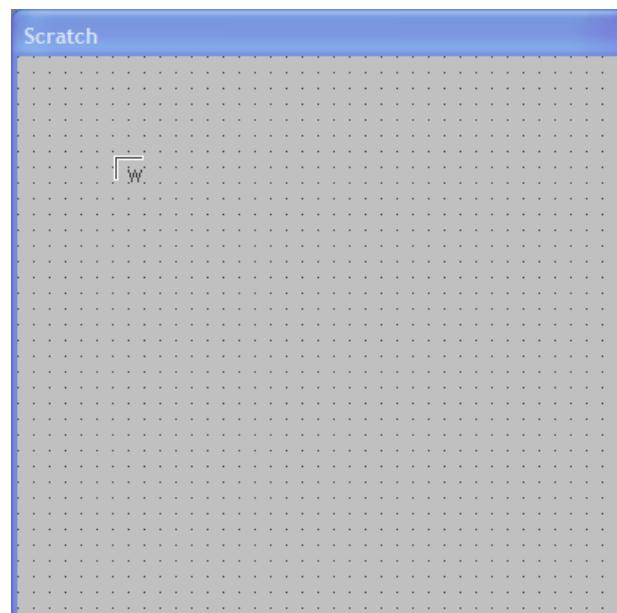
- b. Switches를 선택합니다.

메인 view pane에 이용 가능한 스위치 Wizard가 나타납니다.



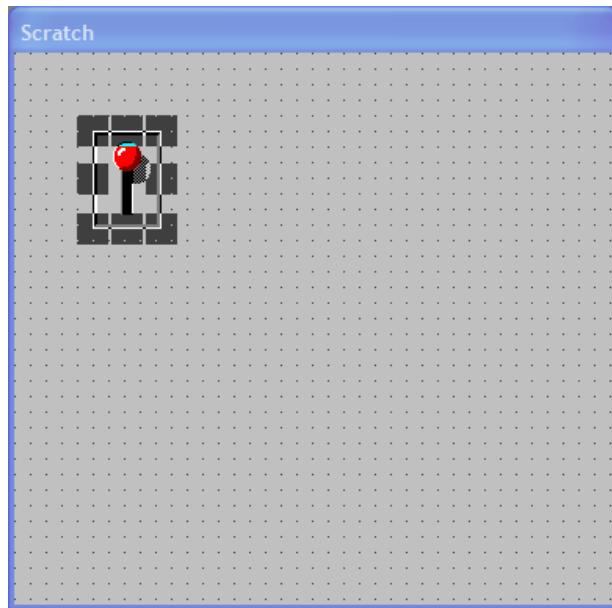
- c. Fixture Switch Wizard를 선택합니다.
- d. OK를 클릭합니다. 대화 상자가 닫히고 Scratch 윈도우가 나타납니다. 윈도우로 전환 될

때 커서가 로 변경 됩니다.

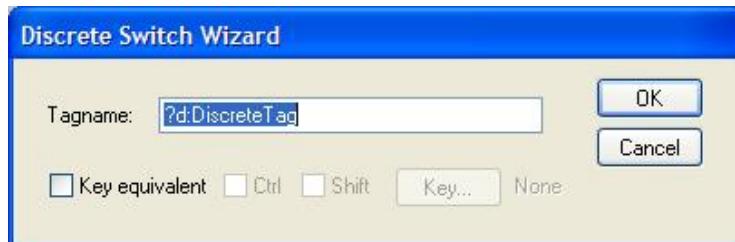


- e. Wizard를 붙이기를 원하는 곳의 위치를 클릭합니다.

Wizard 가 윈도우에 나타납니다.



- f. Wizard를 더블 클릭하면, 구성 창이 보여집니다.

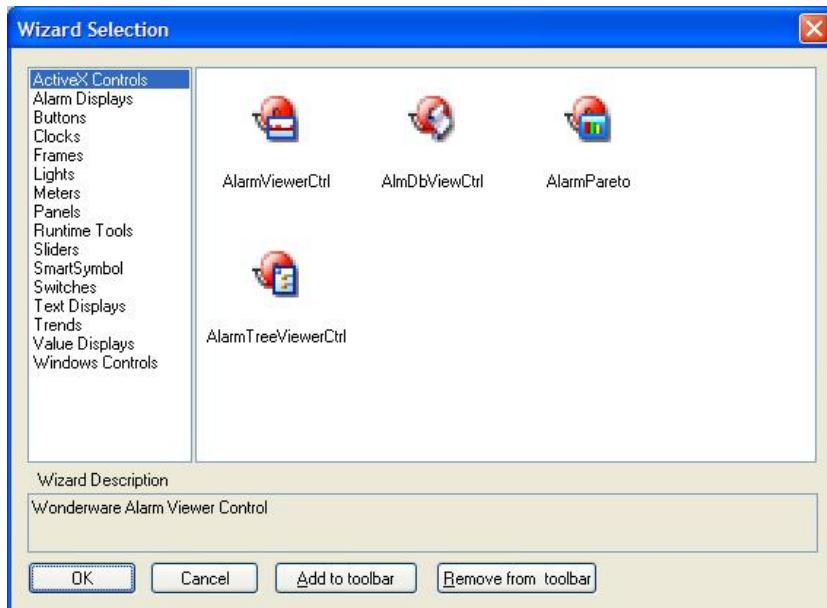


- g. Discrete Switch Wizard 대화상자를 종료하기 위해 Cancel을 클릭합니다. Wizard와 ActiveX 객체들은 나중에 설정 해볼 것입니다.

Note: Reduce Font tool, Line Color tool, Fill Color tool과 같은 tool bar의 메뉴는 Wizard에 바로 사용될 수 없습니다.

Removing Wizards from the Toolbar

Wizard와 ActiveX controls를 Wizard tool bar에 추가하면, 메인 Wizard Selection 대화 상자 안에 Remove from toolbar 버튼이 활성화 됩니다.



- a. Remove Wizard from Toolbar 대화상자를 보여 주기 위해 Remove from Toolbar를 클릭합니다.



- b. Toll bar로부터 제거하려는 Wizard를 선택합니다.
- c. OK를 클릭합니다.

Lab 1 - 새로운 InTouch application, 윈도우들, 그래픽들 생성

ArchesrA IDE 실행

ArchesrA IDE 를 실행 시킴으로써, ArchesrA IDE 에서 Managed InTouch application 들을 생성합니다.

1. Start / Program Files / Wonderware / ArchesrA IDE를 클릭합니다.



Connect to Galaxy 윈도우가 보일 것입니다 :

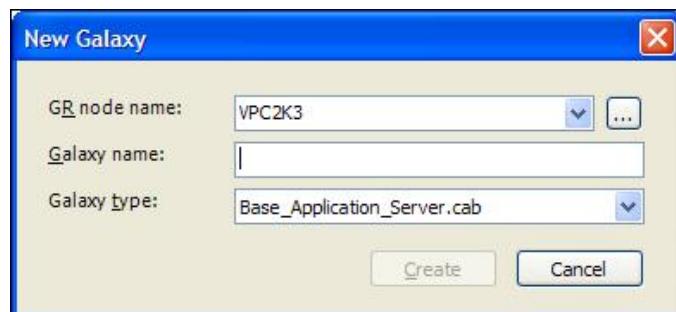


새로운 Galaxy 생성

2. 새로운 Galaxy를 생성하기 위해, **New Galaxy** 버튼을 클릭합니다.

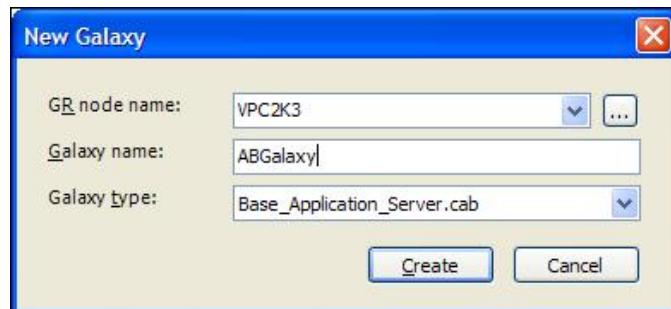


새로운 Galaxy 창이 나타납니다.

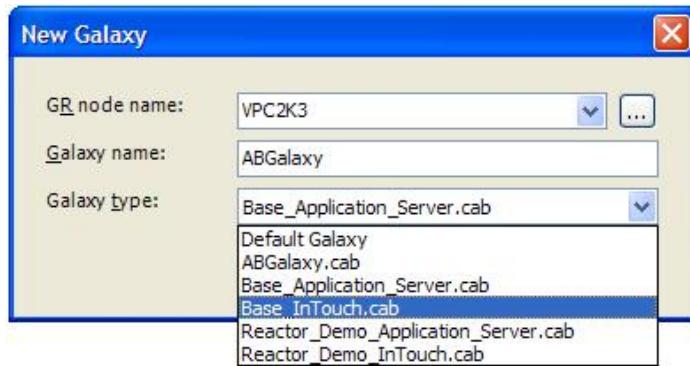


3. Galaxy Name의 공란에 새로운 Galaxy를 위한 이름을 입력합니다.

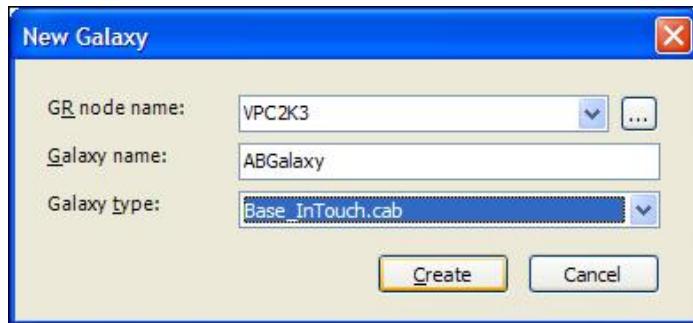
Note : 회사를 대표하거나 자신을 의미하는 단어나 문자로 Galaxies 이름을 지을 수 있습니다. 예로, Arthur Burton은 ABGalaxy라는 Galaxy를 생성할 수 있습니다.



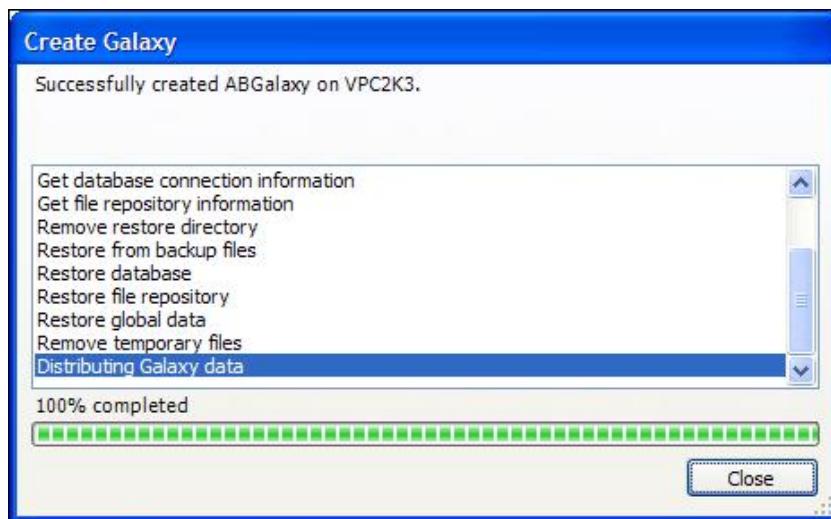
4. Galaxy Type의 목록을 밑으로 내려서, **Base_InTouch.cab** 옵션을 선택합니다.



5. Create 버튼을 클릭합니다.



ArchesrA IDE는 정보를 모으고, **ABGalaxy database**를 생성하기 위해 필요한 구성요소들을 할당합니다.



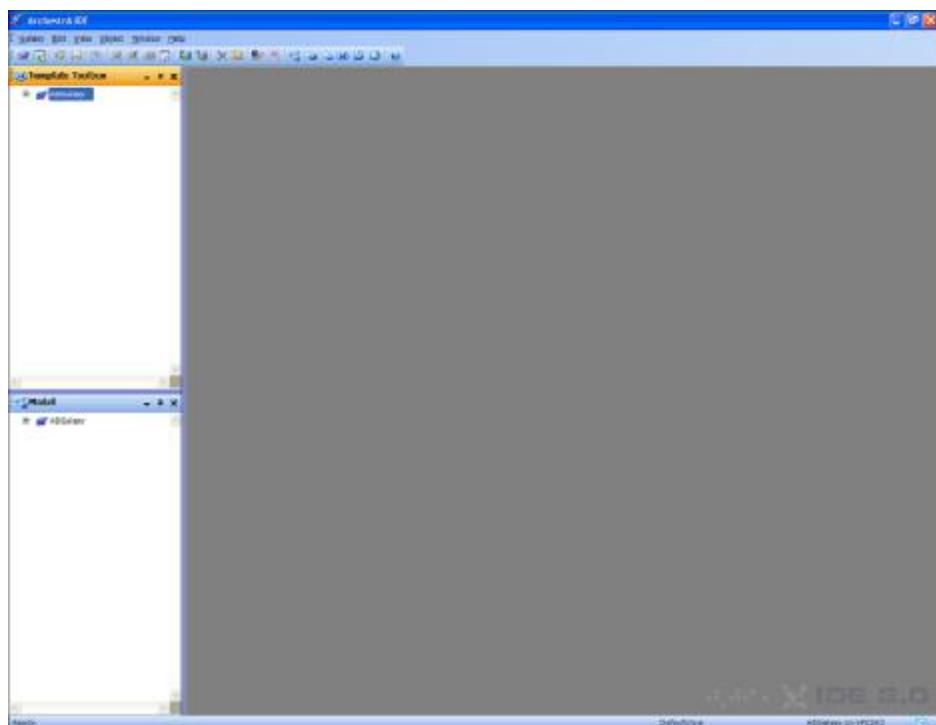
6. Create Galaxy 창에서 하단의 슬라이드가 **100% completed**로 완료 되면, Connect to Galaxy 창으로 돌아가기 위해 Close버튼을 클릭합니다.

Galaxy에 연결

7. **Connect to Galaxy** 창에서, **Connect** 버튼을 클릭합니다.

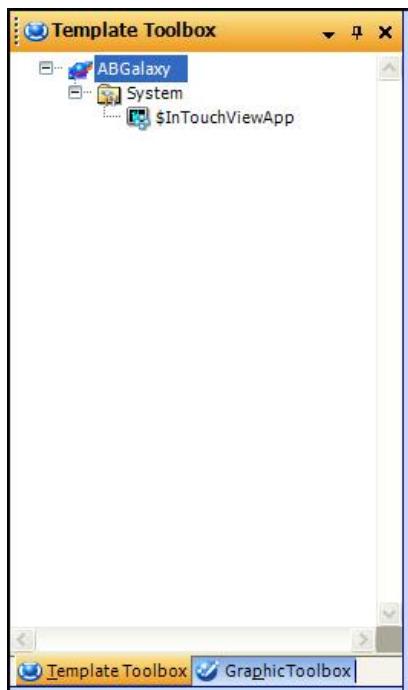


ArchestrA IDE를 열면, 새롭게 만들어진 Galaxy를 볼 수 있습니다.

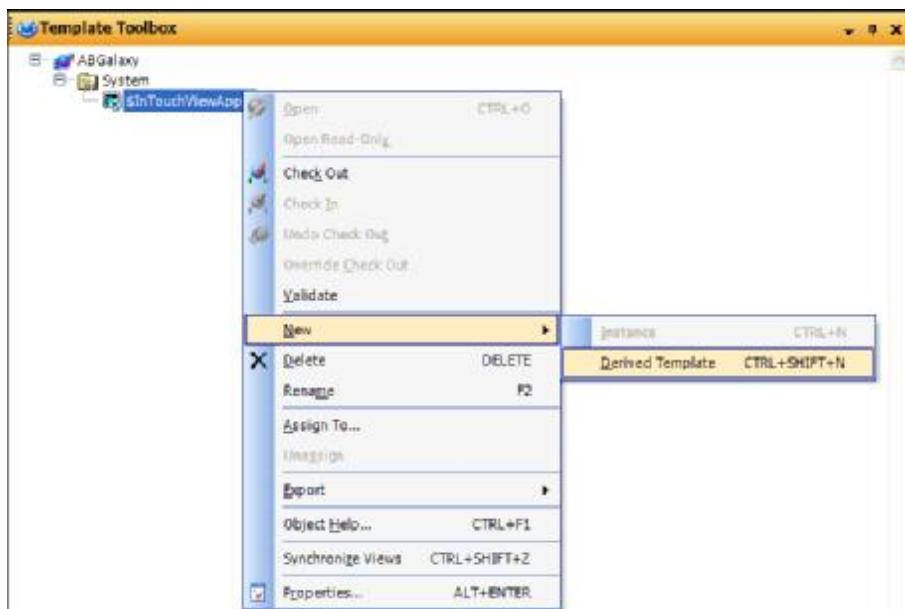


새로운 InTouch Derived Template 생성

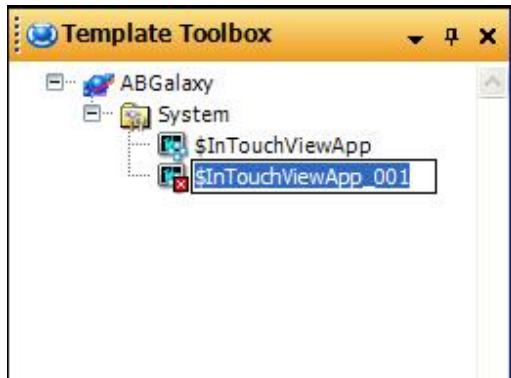
8. Template Toolbox panel안에 Galaxy 항목을 확장합니다.
9. System toolset 풀더를 확장합니다.



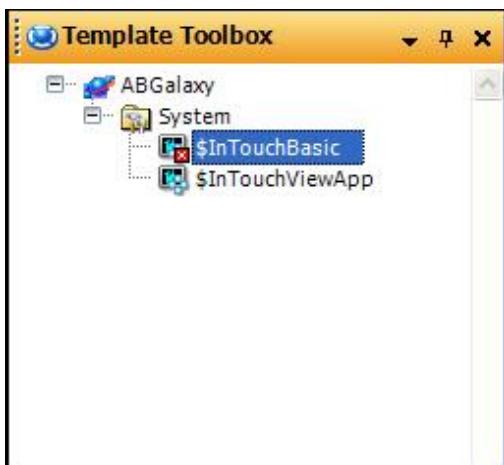
10. 기본 template인 \$InTouchViewApp을 오른쪽-클릭하고, New를 선택한 뒤, Derived Template 옵션을 선택합니다.



잠시 후, **\$InTouchViewApp_001**이라는 새로운 derived template이 나타납니다. 이 template은 사용 가능하며, 이름을 변경할 수도 있습니다.



11. 새로운 derived template를 \$InTouchBasic이라는 이름으로 변경합니다.

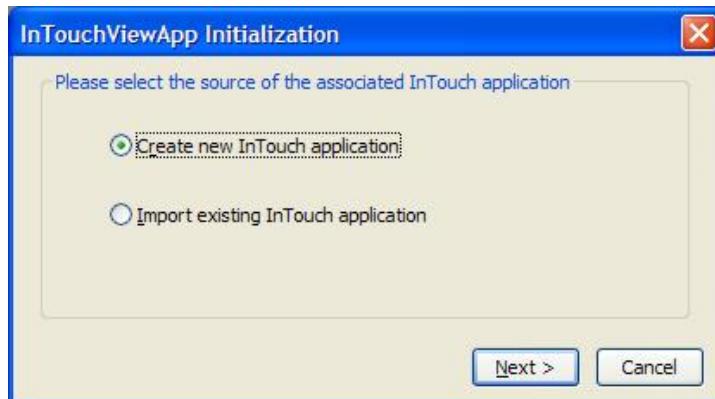


Note : 새로운 derived template의 이름을 재정의 하기 전에, 아래의 요소들을 고려해야 합니다 :

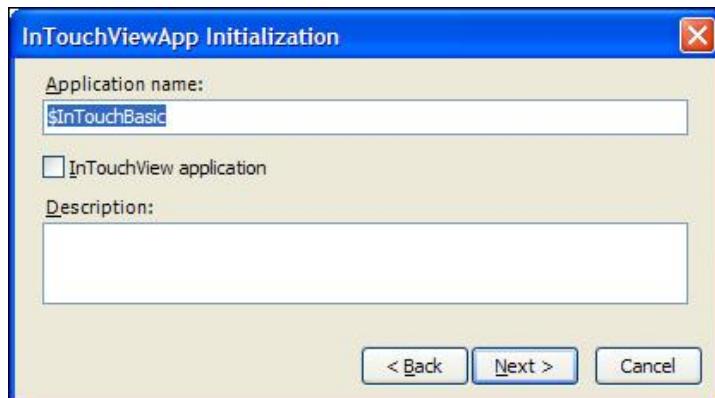
- | 선택한 InTouchView derived template의 이름은 항상 \$와 같이 시작되어야 합니다. \$를 생략하고 입력한 후에, **Enter** 키를 입력하면, \$가 나타날 것입니다.
- | InTouchView derived template의 이름은 최대 32개의 알파벳 또는 특수문자들을 포함 시킬 수 있습니다.
- | InTouchView derived template의 이름에 공백을 포함할 수는 없습니다.
- | Derived template 이름에서 두 번째 문자로 \$는 올 수 없습니다.
- | 최소한 한 개의 문자는 이름에 포함되어야 합니다.

새로운 IDE-Managed InTouch Application 생성

12. Derived template인 \$InTouchBasic을 더블-클릭합니다.
InTouchViewApp Initialization 창이 나타납니다.



13. Create New InTouch Application을 선택하고, Next 버튼을 클릭합니다.

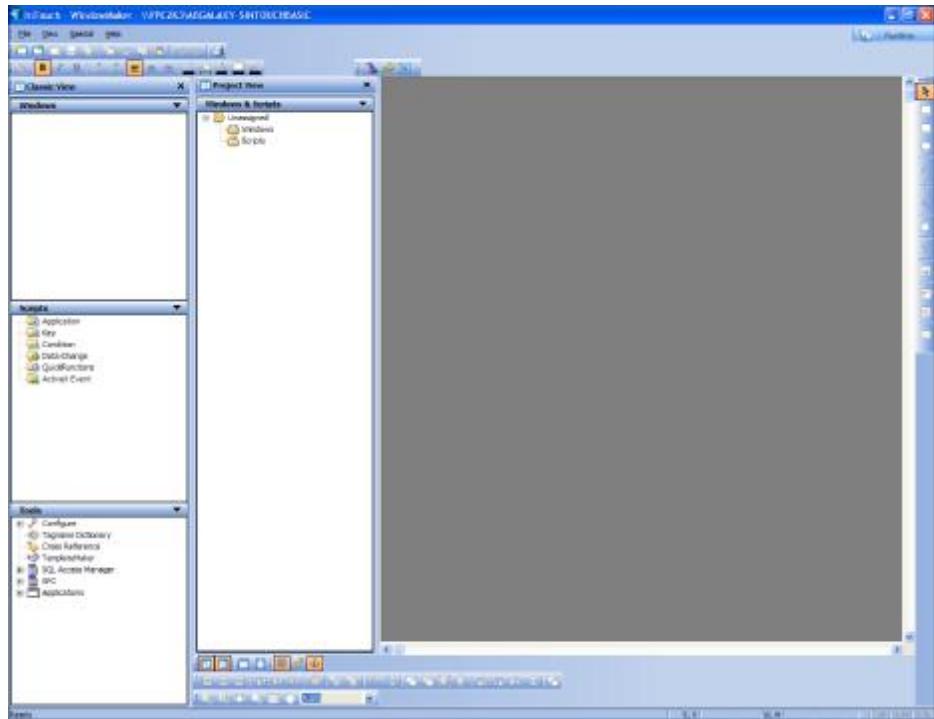


14. Derived template으로부터 생성된, InTouch application의 이름으로 **\$InTouchBasic** 을 입력합니다.(Description은 옵션사항입니다.)

Note : InTouchView Application 옵션의 선택이 해제 되어 있는지를 확인합니다. 외부의 데이터 소스처럼 ArchestrA 참조를 사용하는 Managed InTouch application을 생성했을 시에만 사용가능합니다.

15. **WindowMaker**를 시작하기 위해 **Next** 버튼을 클릭하고, 새로이 생성된 **\$InTouchBasic** managed application을 엽니다.

잠시 후에, WindowMaker가 실행되면, 위에서 생성한 **\$InTouchBasic** application을 엽니다.



Note : Application Explorer (Classic View 와 Project View) 의 두 개의 View 모두 application 을 개발하기 위해 필요한 모든 요소들을 포함하고 있습니다. 두 View 모두 사용하거나, 하나만을 사용할 수도 있지만, 여기서는 주로 Project View 를 사용 할 것입니다.

랩을 진행하는데 있어서, graphic 화면을 크게 쓰는 것이 유리하기 때문에, Application Explorer 의 Classic View panel 을 숨길 것을 권장합니다.:

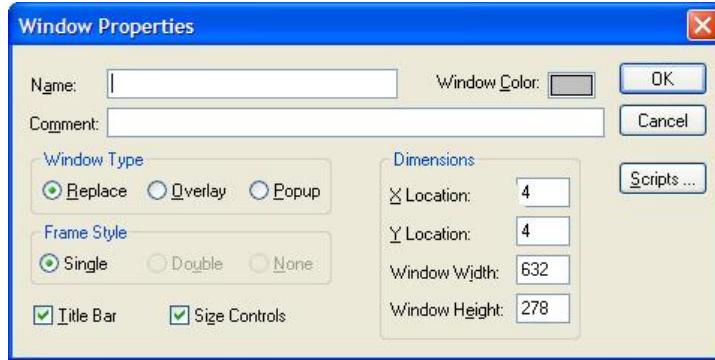
16. View toolbar 에서 **Classic View hide/show toggle** 버튼 을 클릭합니다.

새로운 윈도우 생성

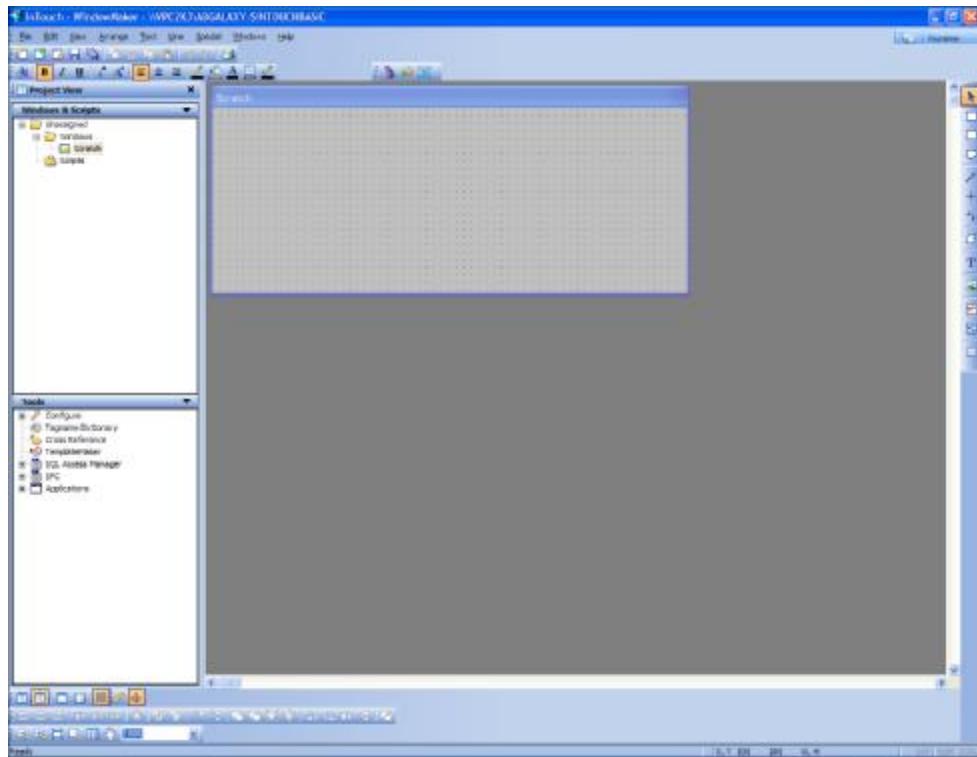
17. **New Window** 버튼 을 클릭합니다.

Hint : General toolbar에서 **New Window** 버튼을 찾습니다.

Window Properties 창이 나타 납니다 :



18. Name 공란에 **Scratch**를 입력하고, **OK** 버튼을 클릭합니다.
WindowMaker에 새로운 윈도우가 나타납니다.

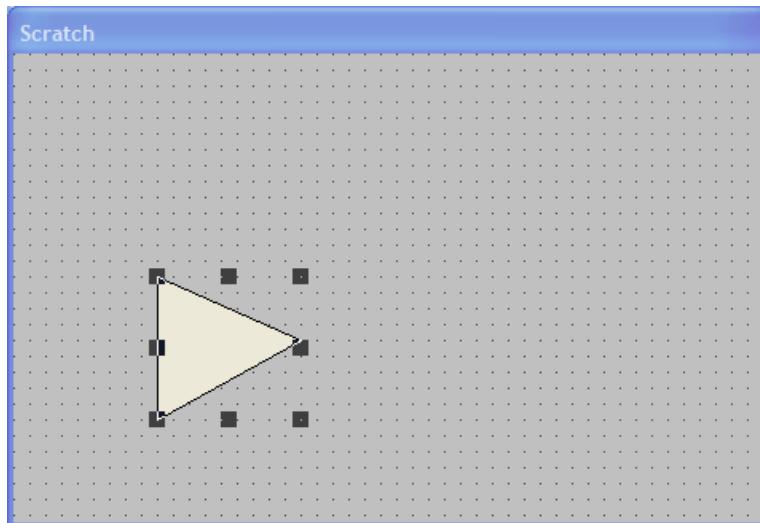


19. 새로운 윈도우의 크기를 조정하기 위해서, 윈도우의 하단-오른쪽 모서리를 클릭하고 끌어당기기를(drag) 합니다.

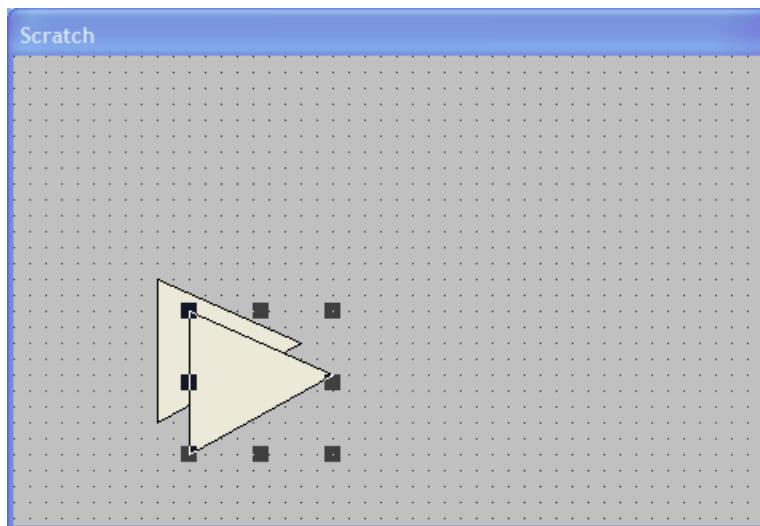
새로운 그래픽들 생성

이번 랙에서는 새로운 윈도우에 간단한 뱀브 객체와 버튼을 생성합니다.
Scratch 윈도우가 열려 있지 않았다면, 지금 열어 놓습니다.

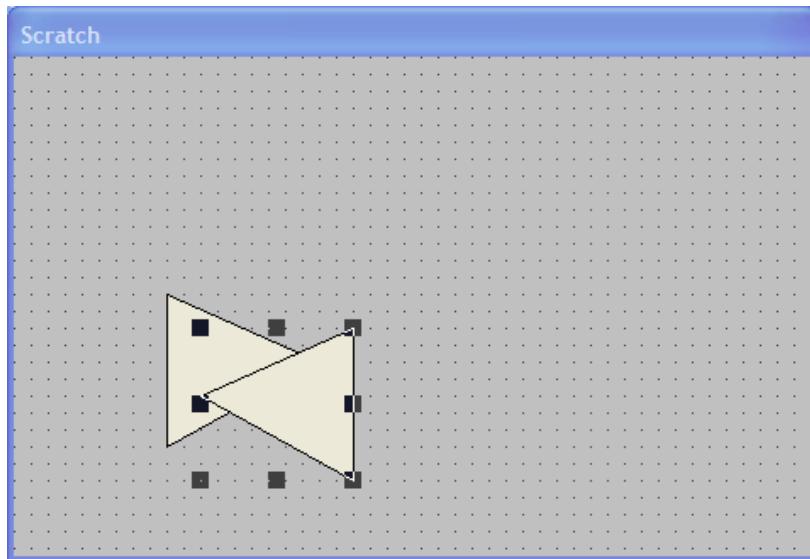
20. Toolbar에서 **Polygon** 을 클릭하고, 삼각형을 그립니다.



21. 삼각형을 선택하고, toolbar에서 **Duplicate** 를 클릭합니다.



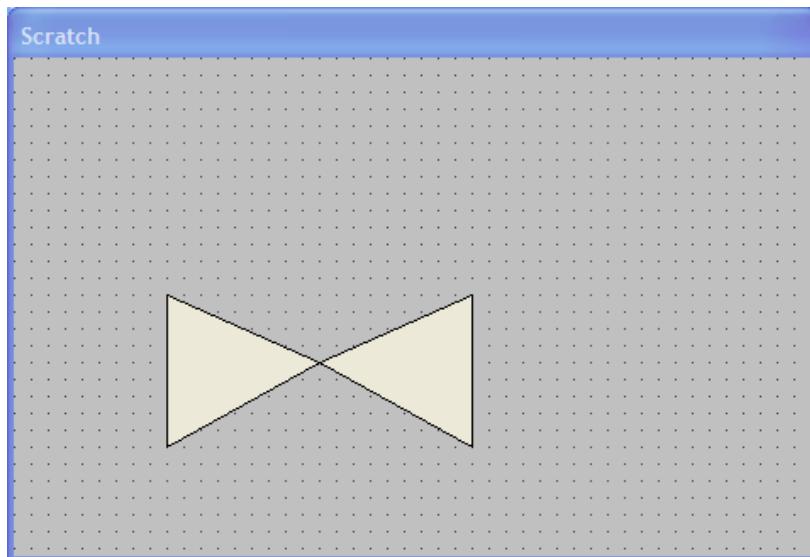
22. Toolbar에서 **Arrange Flip Horizontally** 를 클릭합니다.



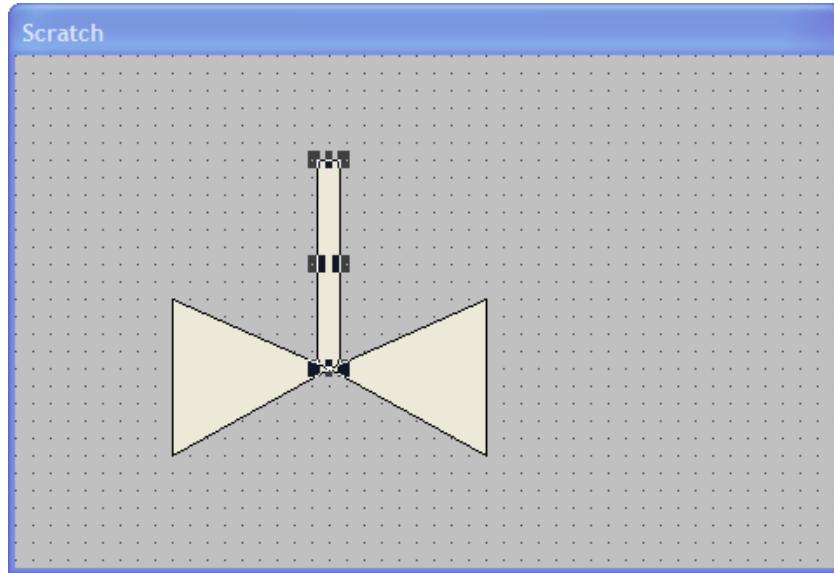
Tip : 객체가 이동되는 것을 막기 위해서는, 객체를 클릭할 경우, **SHIFT**키를 같이 누릅니다.

마우스나 방향키를 사용해서, 그림처럼 밸브의 양쪽 반형들을 각각 위치시킵니다.

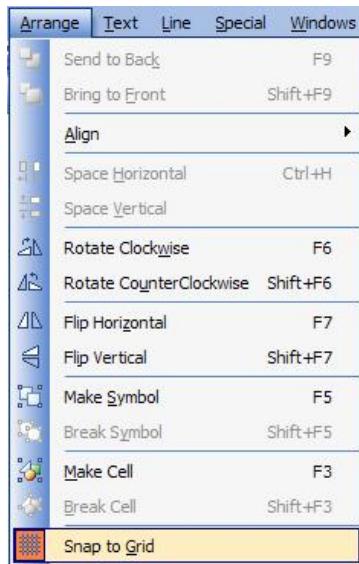
23. **Shift+Ctrl** 키를 누른 상태에서, 객체를 각각 **10** 또는 **50** 픽셀 이동시키기 위해 방향키를 사용합니다.



24. Toolbar에서 **Rectangle** 을 선택합니다.
25. 다음의 그림과 비슷하게 긴 사각형을 그리고 위치에 둡니다.



26. Main WindowMaker 윈도우에서 **Arrange** 메뉴를 선택합니다.
Snap to Grid 항목이 선택되어 있을 것입니다.

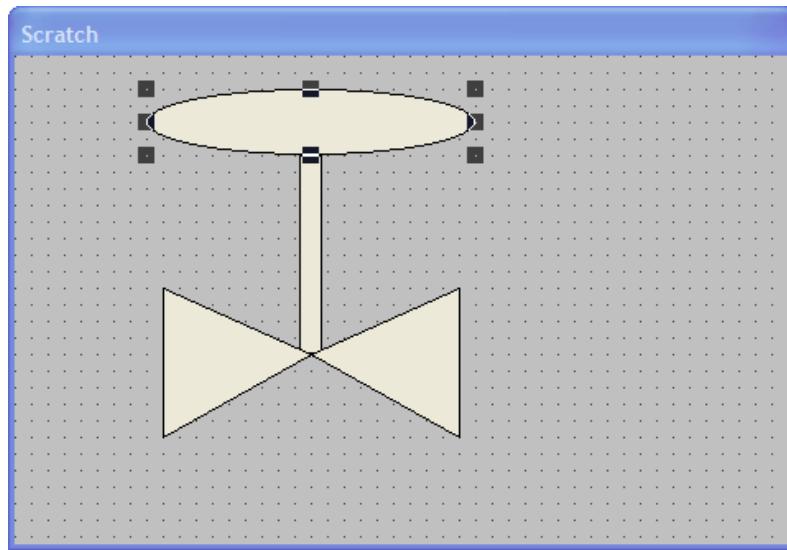


Snap to Grid 속성을 선택해제 한다는 것은 객체를 더욱 정확하게 정렬할 수 있게 만든다는 것입니다. Scratch 윈도우 위에 점자 무늬는 **Snap to Grid** 속성을 끌 경우, 사라지며, 픽셀 하나씩 객체를 이동시킬 수 있습니다.

View toolbar 위에 Grid 버튼 을 클릭하거나, Arrange menu에서 grid를 On/Off 시킬 수 있습니다.

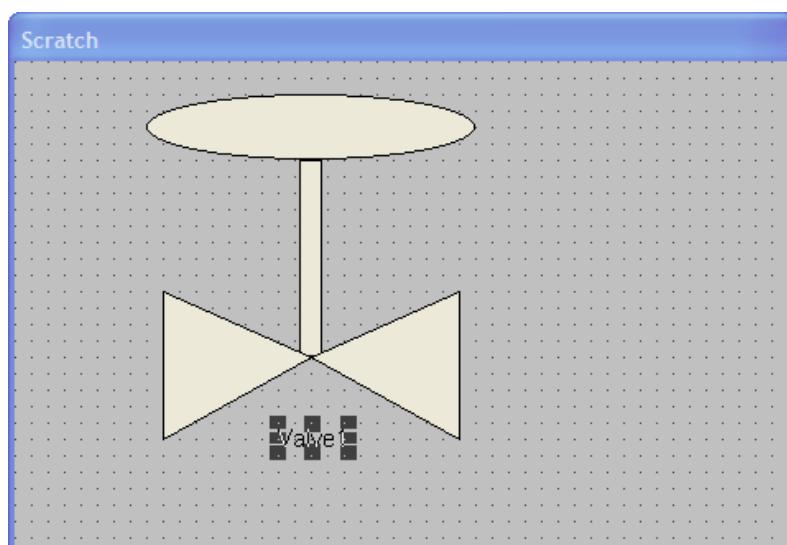
27. 사각형의 객체를 선택하고 **Send to Back** 을 클릭합니다.
이 객체의 위치를 다른 객체들보다 뒤로 보내게 되며 그래픽을 깔끔하게 만듭니다.
28. Toolbar에서 **Ellipse** 를 클릭합니다.

29. 타원 형태의 뱀브 핸들을 만들고, 아래 그림과 같이 위치시킵니다.



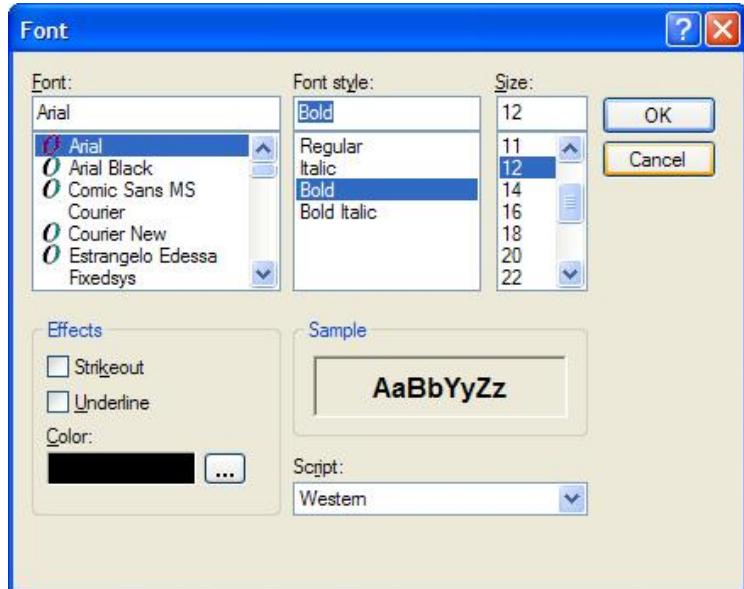
30. Toolbar에서 **Text** 를 선택합니다.

31. 아래의 그림과 같은 위치에 Valve1이라는 문자를 추가합니다.

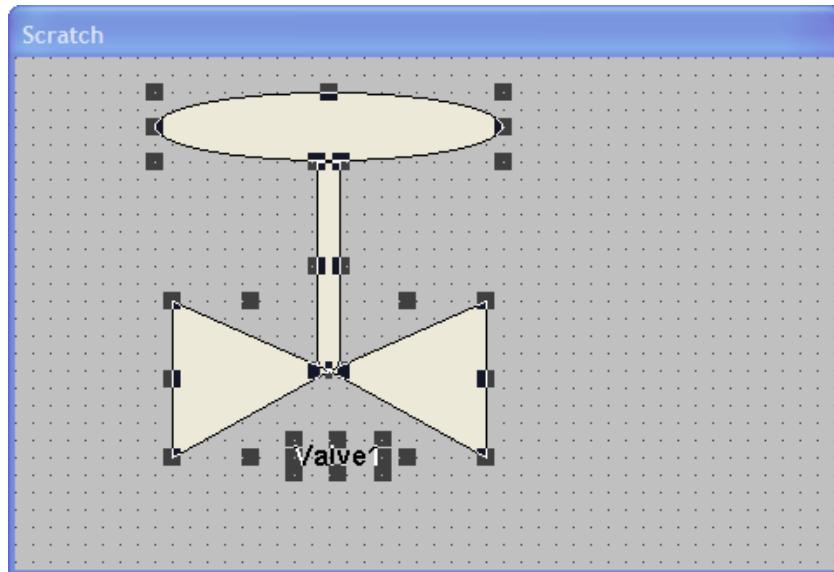


32. Text 객체에서 오른쪽-클릭을 하고, Fonts를 선택합니다.

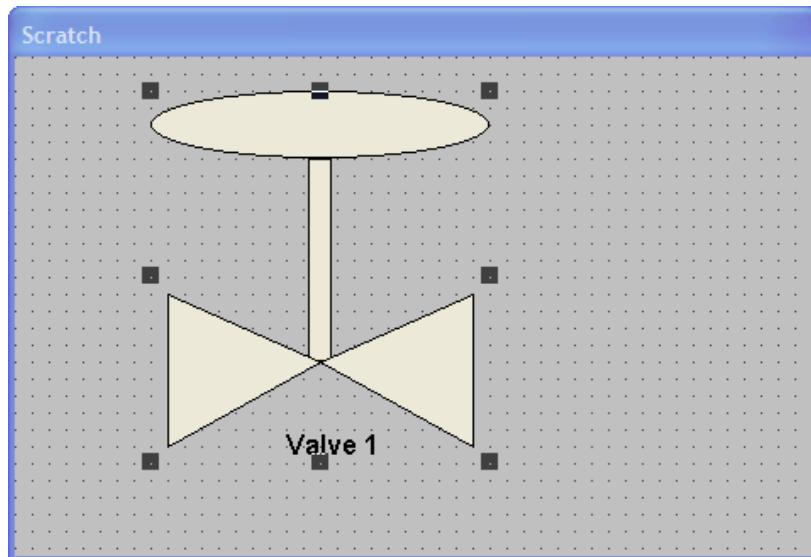
폰트 창이 다음과 같이 나타납니다 :



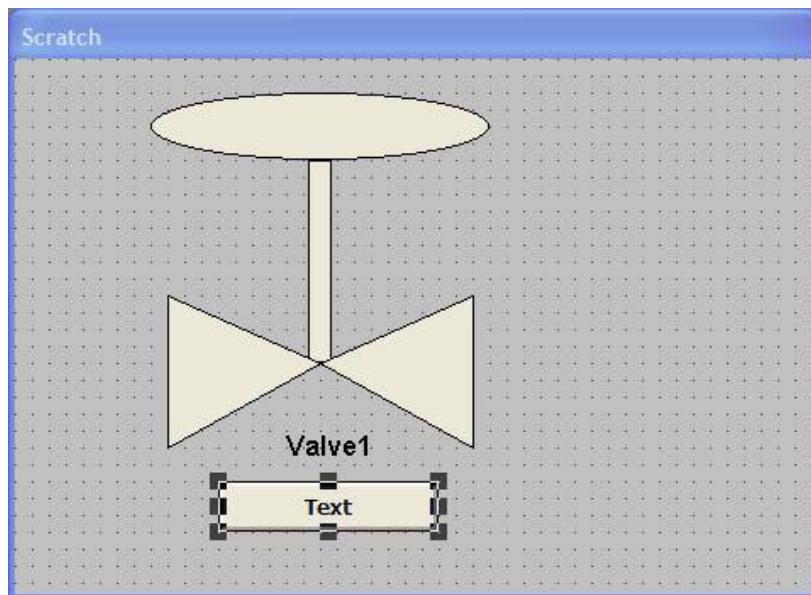
33. Arial 폰트, Bold, 12 size로 설정합니다. 문자가 다른 시스템에 사용되거나 크기를 재조정해도 True-Type 폰트를 지키면 text는 일관되게 적용됩니다.
34. OK버튼을 클릭합니다. 뱀브는 현재 새로운 4개의 분리된 객체들과 하나의 문자 객체로 구성되어 있습니다.
35. 윈도우의 모든 객체를 선택하기 위해 F2 키를 누릅니다.



36. Toolbar에서 Make Symbol [] 을 클릭하게 되면, 전체 symbol은 모서리 부분을 클릭하고 끌어당기기를 함으로써 크기를 재조정 할 수 있습니다.



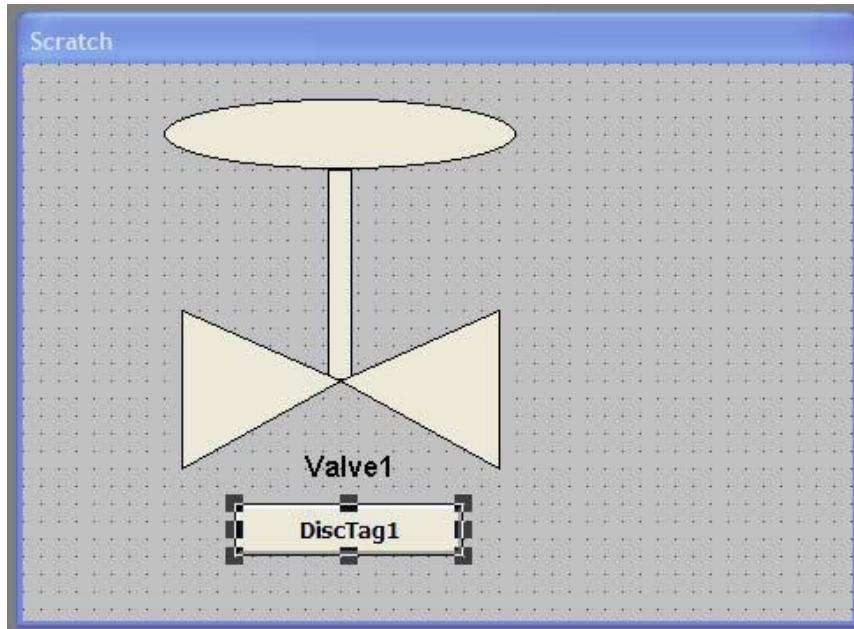
37. Toolbar에서 **Button** 을 클릭하고, 밸브의 아래에 symbol 버튼을 그립니다.



38. 버튼을 선택하고 **Ctrl+L**을 누릅니다.
Substitute Strings 창이 나타납니다.



39. **New String**의 공란에 **DiscTag1**을 입력합니다.
40. **OK** 버튼을 클릭합니다.
41. 문자 객체를 일치시키기 위해 버튼의 **Font** 속성을 수정합니다 :
Arial, Bold, 12 size로 설정합니다.
아래 그림과 비슷하게 Scratch 윈도우가 보여져야 합니다.



42. Scratch 윈도우를 저장 합니다.



Module 3

태그 명 사전(Tagname Dictionary)

Section 1 – 태그명 정의

Lab 2 – 태그 생성

Section 1 – 태그명 정의

태그란?

태그란 간단히 말하자면, ‘데이터 값’이라고 정의 할 수 있습니다. 이 ‘데이터 값’은 탱크의 용적 레벨, 온도, 통신 상태, switch position, 공정에서 올라오는 실시간 데이터 값 같은 물리적인 데이터를 표현할 수 있으며, 각각의 태그들은 AccessName, 최대/최소 값, 태그명(TagName) 같은 속성들을 가지고 있습니다.

InTouch에서는 각기 다른 TagType 들이 존재하고, 이런 TagType 들은 서로 다른 정보들을 가지고 있습니다. 예를 들자면, I/O-Type Tag 들은 외부의 데이터소스를 InTouch 내부에서 사용할 수 있도록 하기 위해서는 추가적인 정보들을 필요로 한다. 이런 정보들은 TagName Dictionary 에서 설정이 가능합니다.

태그명 사전(Tagname Dictionary)이란?

태그명 사전(실행 DB 라고 할 수 있겠다)은 InTouch 의 핵심이라고도 할 수 있습니다. RunTime 실행 시, 태그명 사전은 데이터베이스에 저장되어 있는 모든 태그들은 현재 데이터 값을 가지고 있으며, InTouch 는 실행 DB 를 만들기 위해, 생성되는 모든 변수들의 정보를 필요로 하며, 태그에는 태그명과 유형(Type)이 지정되어야 합니다.

RunTime 실행 동안, InTouch 어플리케이션은 태그명 사전에 접근 함으로써 애니메이션 링크, 스크립트에서 상세한 태그 정보를 제공하게 됩니다.

태그에 변경사항(데이터 변경이나 설정 변경 혹은 새로운 태그 생성 같은)이 있을 경우, 그 변경사항은 TagName Dictionary 에 저장 됩니다.

태그의 생성이나 설정 변경은 Tag Editor, 혹은 Tag Browser 에서 작업 할 수 있습니다.

태그명 정의 규정

태그명 정의는 User 와 InTouch 가 태그를 식별 가능하게 해주는 작업으로써, 최대 32 개의 문자까지 가능 하며, 반드시 첫 문자는 알파벳 문자(A-Z or a-z) 또는 숫자(0-9)로 시작 해야 합니다. 나머지 뒤에 오는 문자들은 A-Z, a-z, 0-9, !, @, -, ?, #, \$, %, _, \ 그리고 &가 가능합니다. 태그명에 첫문자를 숫자를 기입하는 경우, 최소한 하나이상의 알파벳문자가 뒤에 와야 합니다. 또한 태그명에는 빈칸이 와서는 안되며, 16 진수의 표현(0XABCDEF)은 불가능합니다.

Note : \ 기호는 Super Tag 에서만 정의가 가능합니다.

인덱싱(Indexing)

태그명은 자동으로 인덱싱되어 질 수 있습니다.. 예를 들어 **TagName Dictionary**에서 태그명으로 R4001 을 넣고 저장한 후, 다시 New 버튼을 클릭했다면, 새로운 태그명은 자동으로 R4002 로 목록화 됩니다.

문자 분리

만일 태그명이 문자와 숫자로 분리되어 있다면, 이것 또한 자동 목록화가 됩니다. 예를 들면,

N7-0 는 N7-1 로 목록화 됩니다.

(단, 숫자는 오직 증가가 되며 목록화가 됩니다.)

태그명에 ‘-’기호를 사용할 때에는 주의해야 합니다. ‘-’기호는 태그명에 사용하는 것이 가능합니다. (태그명이 숫자로 시작하는 태그명에서는 숫자뒤에 ‘-’를 두는 것은 불가능합니다) ‘-’기호가 수학적인 연산식이나 로직 소스에서 사용되는 연산자이기도 하기 때문에 혼동의 여지를 줄 수 있습니다. 예를 들어, A = B-C 라는 식은 B 빼기 C 는 A 와 같다라는 식입니다. 그러나 InTouch 에서는 B-C 라는 태그명 역시 가능하므로, A 태그에 B-C 태그를 대입하라는 명령으로 오인할 수 있습니다. 이런 오류를 방지하기 위해서는 A = B – C 라는 식으로 빈칸을 둘 것을 권합니다.

태그 유형

InTouch 에는 메모리-유형과 I/O-유형의 태그가 있다. 만약 I/O 서버나 다른 어플리케이션으로 데이터값을 쓰거나 받으려 할 경우, 반드시 I/O-유형의 태그를 써야만 합니다.

이처럼 태그를 정의 할 때에는, 사용하고자하는 용도에 맞는 정확한 태그유형을 지정해야 합니다.

메모리 유형

이것은 **InTouch Application** 안에서 내부적으로 사용하는 태그 유형입니다. 이 유형은 시스템 상수나 시뮬레이션을 만드는 데에 사용될 수도 있고, 다른 윈도우 프로그램에 의해 접근 될 계산된 변수를 만드는 데에도 유용합니다.

공정을 시뮬레이션 할 경우, 액션 스크립트에 의해 값이 변경되는 ‘COUNT’라는 메모리 태그를 정의함으로써 배경 QuickScript 로직을 제어하는 것이 가능하게 된다는 것입니다.

이 때의 QuickScript 는 현재 진행되고 있는 단계에서 다양한 애니메이션 효과를 나타내주는 역할을 합니다.

다음은 메모리 태그의 4 가지 종류입니다.

메모리 이산

값이 0 (거짓, Off) 혹은 1 (참, On)인 내부 이산 태그명.

메모리 정수

-2,147,483,648 부터 2,147,483,647 사이의 부호 달린 32 비트의 정수 값.

메모리 실수

부동 소수점 메모리 태그명. 부동 소수점 값은 $\pm 3.4e38$ 사이가 될 것입니다. 모든 부동 소수점 계산은 64 비트의 해답으로 실행되나, 결과는 32 비트로 저장됩니다

메모리 메시지

131 글자까지 가능한 문자열.

I/O 유형 태그명

이것은 외부에서 들어오는 데이터 혹은 외부로 나가는 데이터들을 읽고 쓰기 위한 태그 유형입니다. 이 유형의 태그들은 프로그래밍이 가능한 제어기, 프로세스 컴퓨터로부터의 모든 입, 출력과 네트워크상의 다른 노드로부터의 데이터를 받는 것이 가능합니다. I/O 태그유형은 Microsoft 의 Dynamic Data Exchange(DDE) 또는 Wonderware 사의 SuitLink(이 프로토콜을 추천하는 바이다) 통신 프로토콜을 통하여 접근이 가능합니다.

I/O 유형의 태그에 데이터값이 변경될 시에는, 즉시 원격 Application 에 적용되며, 반대로 Item 으로 연결되어 있는 태그명이 원격 Application 에서 수정될 때마다 해당 태그 역시 업데이트 될 것입니다. 기본적으로 모든 I/O 태그유형은 Read/Write 가 가능하나 필요하다면 Tagname Dictionary 설정에서 Read only 라디오 버튼을 선택함으로써 읽기전용으로 제한 할 수 있습니다.

다음은 I/O 태그의 4 가지 종류입니다.

I/O 이산

값이 0 (거짓, Off) 혹은 1 (참, On)인 이산 입/출력 태그명.

I/O 정수

-2,147,483,648 부터 2,147,483,647 사이의 부호 달린 32 비트의 정수 값.

I/O 실수

부동 소수점 메모리 태그명. 부동 소수점 값은 $\pm 3.4e^{38}$ 사이가 될 것입니다. 모든 부동 소수점 계산은 64 비트의 해답으로 실행되나, 결과는 32 비트로 저장.

I/O 메시지

131 글자까지 가능한 텍스트 문자열 입/출력 태그명.

기타 태그유형

동적 알람 디스플레이, 이력 트렌드, 이력 트렌드의 각각의 팬이 작성하는 태그를 모니터링하거나 제어하는 기능 등 이러한 복잡한 기능을 실행하는데 사용되는 몇 가지 특수 태그 유형이 있습니다. 또한, 다수의 원본에 태그명을 재지정할 수 있게 해주는 간접적인 태그명도 있습니다.

다음에서는 이러한 다양한 태그 유형을 설명하고 있습니다:

그룹 변수 (Group Var)

이 태그 유형은 동적 알람 디스플레이, 디스크 기록, 인쇄 기록을 만들기 위해서, 할당된 알람그룹을 태그로 사용합니다. 이 유형의 태그들은 특정한 그룹변수와 연관된 알람 모두를 디스플레이 하는 알람 윈도우 혹은 알람로그들을 생성할 수 있게 해줍니다. 또한, 그룹변수에 다른 알람그룹들을 할당함으로써, 로깅되거나 디스플레이 되는 알람을 관리할 수 있습니다.

그룹변수 태그명은 같은 알람 윈도우에 있는 공장의 서로 다른 경보들을 선택적으로 보여주기 위해 Operator 가 제어 할 수 있는 버튼을 만드는 경우에도 사용될 수 있습니다. 알람그룹과 관계가 있는 모든 **.fields** 는 그룹 변수에서 적용될 수 있습니다.

이력 트렌트 (Hist Trend)

InTouch 에서는 당신이 이력 트렌드를 생성 할 때 **Hist Trend** 유형 태그를 필요로 합니다. 이력 트렌드와 관계가 있는 모든 **.field** 는 **Hist Trend** 태그에 적용될 수 있습니다.

TagID Indirect and Super Tag

이것은 이력 트렌드 객체들과 함께 사용되어 지는 특별한 타입입니다. 이력 트렌드 안에서 작성되고 있는 태그명에 대한 정보를 가져오기 위해서 TagID 유형을 사용할 수 있습니다. 대부분의 경우에는, 특정한 팬에 할당된 태그의 이름을 보여주거나, 그 팬에 할당된 태그명을 수정하기 위해서는 **TagID** 를 사용합니다.

어떤 이력 트랜드 안에서 특정한 팬에 새로운 태그명을 선언 하기 위해서는 QuickScript 의
로직을 사용 할 수 있습니다. 예를들면, 다음과 같은 스크립트가 QuickScript 내에서 사용될 수
있습니다.

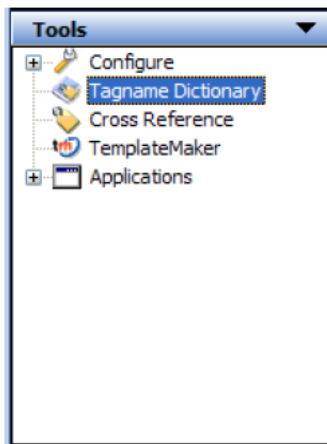
MyHistTrendTag.Pen1 = MyLoggedTag.TagID;

이 QuickScript 가 실행 되어 질 때, HistTrend 의 태그명 “MyHistTrend” 의 이력 트랜드내부에
가지고 있는 Pen1 은 이력 저장되는 데이터를 “MyLoggedTag” 에 트랜딩을 시작하게 됩니다.

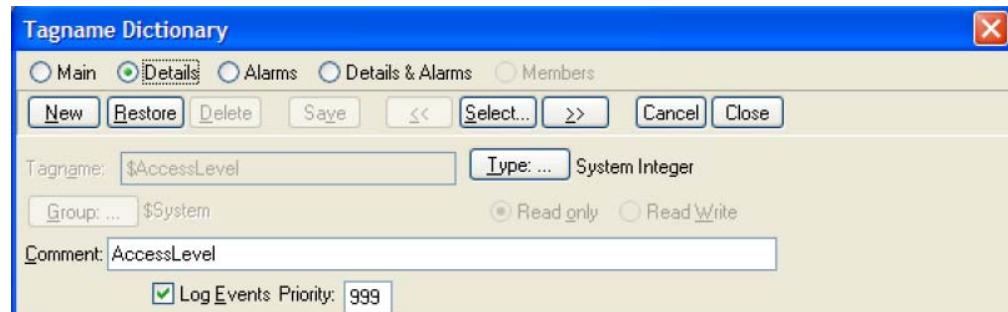
새 태그 생성

태그들과 그 태그명(다른 속성을 역시 마찬가지로)을 만들기 위해서는 tagname dictionary 의
Tagname Editor 를 사용합니다.

- a. **WindowMaker** 의 메뉴바에서 Special / Tagname Dictionary , 또는 왼쪽면에
Application Explorer 의 Tools panel 의 Tagname Dictionary 를 더블 클릭합니다.



Tangname dictionary 에 처음 접속을 하게 되면, 내부 시스템 태그명 \$AccessLevel 에
정의된 내용을 볼 수가 있습니다. Tangname dictionary 에 태그명을 한번이라도 정의
했다면, 다시 접속 할 때, 마지막으로 선언된 태그명이 보여집니다.

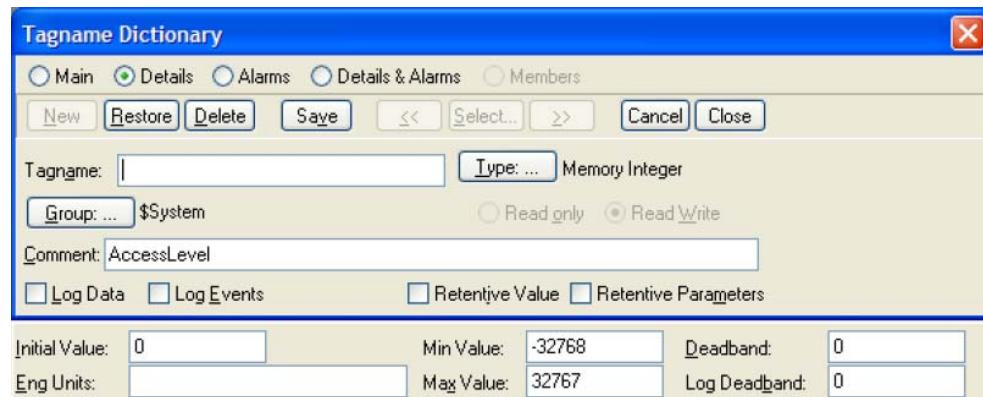


- b. Tagname dictionary에 현재 저장되어 있는 태그명을 보기 위해서는 << 또는 >>을 클릭 합니다.

(browse 버튼은 이전이나 다음에 표시할 태그명이 없을 때는 비활성화 됩니다)

- c. New 버튼을 클릭합니다.

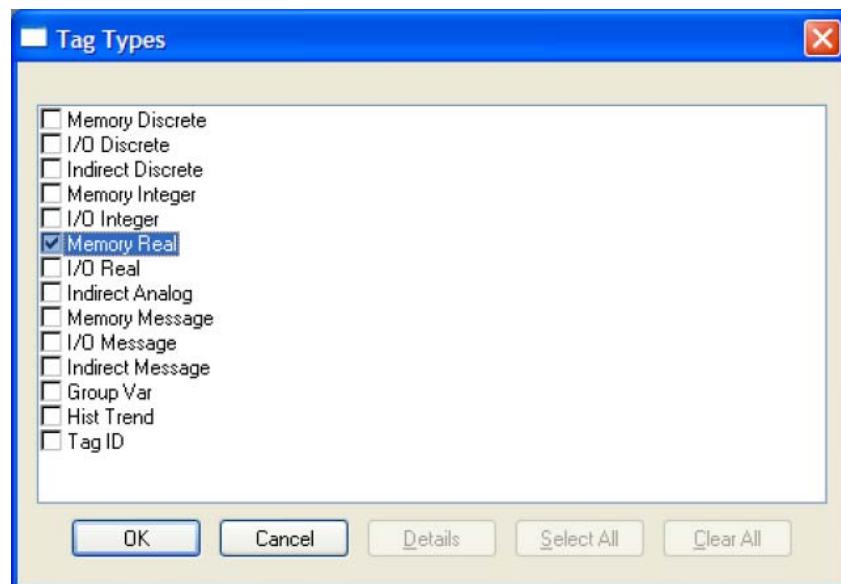
Tagname 필드가 활성화 됩니다.



- d. TagName 필드 안에 RealTag라는 태그명을 입력합니다.

- e. Type 버튼을 클릭합니다.

Tag Typeダイヤログ창이 보입니다:



- f. Memory Real 타입을 클릭.

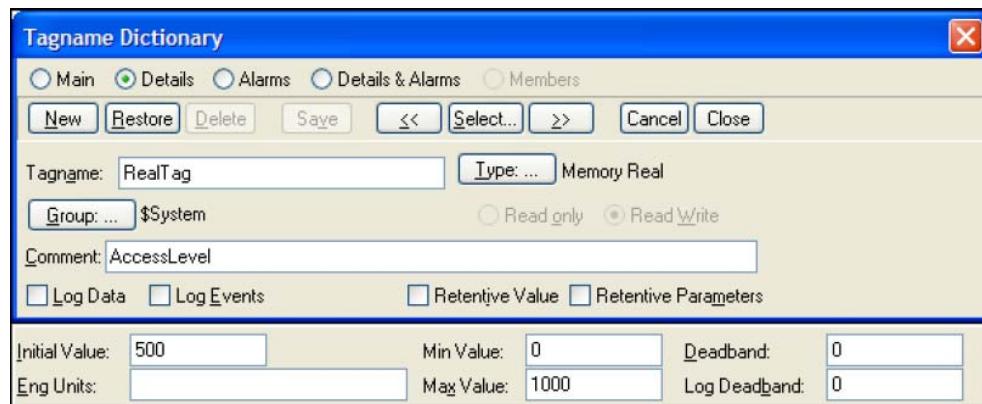
- g. OK 버튼 클릭.

h. 아래와 같이 RealTag 속성을 정의합니다.

Initial Value 필드에 500을 입력

- Min Value 필드에 0을 입력
- Max Value 필드에 1000을 입력
- 나머지 필드는 그대로 둡니다.

Tagname Dictionary 대화상자는 다음 그림과 같이 보일 것입니다:



i. Save 버튼 클릭

이제 Tagname dictionary은 다른 태그를 정의할 준비가 되었습니다.

Tagname Editor 설정 Option

태그명(Tagname) : 태그명은 32개의 문자까지 지원 가능합니다. 태그명이 숫자로 시작해도 가능하며, 뒤에 알파벳 e, E가 와서는 안됩니다. InTouch에서는 이것을 지수로 오인할 수도 있기 때문입니다.

그룹(Group) :

특정한 알람그룹에 태그를 할당합니다. 이 버튼을 클릭하면, 알람그룹을 할당할 수 있는 대화상자가 보일 것입니다.

만약 특정 알람그룹을 할당하지 않으면, InTouch에서는 기본적으로 \$System (root 그룹)으로 지정시킵니다.

일단 태그를 생성하고 알람그룹을 할당한 후, 다이얼로그 창을 닫지 않는다면, 추후 정의할 모든 태그들은 다시 수정하지 않는 한, 같은 알람그룹으로 할당 됩니다.

I/O 유형의 태그들은, 만약 Runtime 시 read-only로 태그를 제한하려 할 경우, Read Only 옵션을 선택합니다. I/O 유형의 태그들을 Runtime 시 read/write가 가능하게 하고 싶다면 Read / Write 옵션을 선택합니다.

주석(Comment) :

태그와 관련하여, 시스템적인 혹은 기타 다양한 설명을 50자까지 쓸 수 있습니다.

알람 코멘트(Alarm Comment) :

해당 태그에 알람이 발생할 경우, 알람 윈도우 창에 이 알람 코멘트를 보여주는 것이 가능합니다. 분산 알람 시스템에서는 ACK 된 알람과 관련된 Operator 의 코멘트를 저장하기 위해 태그의 알람 코멘트를 사용합니다.

Log Data : 실행시간 동안, 특정한 Log Deadband 값 보다 그 태그의 engineering unit 값이 더 많이 변화할 때마다, 혹은 기본적으로 한 시간에 한번씩 변화율에 상관 없이 이력 로그 파일에 그 태그 데이터를 기록합니다.

- 태그의 데이터값을 저장하기 위해서는, WindowMaker 메인메뉴의 Special / Configure / Historical Logging 을 선택한 후, 이력을 기록하는 것이 가능해집니다.
- 만약, 이 옵션을 취소한다면, 이전에 저장된 데이터값은 InTouch 의 이력트랜드에서 확인 할 수 없게 됩니다.
- 만약, 이 옵션을 다시 선택한다면, 모든 로그 데이터를 다시 확인 할 수 있게 됩니다. 하지만, 이력 트랜드에서 데이터 주이를 확인 할 경우 gap 이 생긴 것을 볼 수 있습니다.
- 또한, Windowviewer 실행되고 있는 동안, WindowMaker 상에서 로깅데이터가 수정되었다면, Windowviewer 가 다시 시작 되기 전까지는 반영이 되지 않습니다.

Log Event :

Operator, I/O, QuickScript 또는 시스템에 의해 초기화된 모든 태그의 변경 사항을 저장합니다.

- Event 모니터링을 하기 위해 태그를 설정 한 경우, 태그의 데이터값이 변경 될 때마다 알람 시스템에 기록이 되며, 이벤트 메시지는 데이터값이 어떻게 변경되었는지, 그리고 Operator, I/O, QuickScript 또는 시스템에 의해 초기화가 되었는지 아닌지를 표현하게 됩니다.
- Log Event 옵션이 선택되면, 우선순위 필드가 활성화 된다. 이 필드에는 1부터 999 까지 선택할 수 있으며, 1 이 가장 높은 우선순위를 가지게 됩니다.

Retentive Value :

Windowviewer 가 종료 될 경우, 태그의 현재 데이터값을 유지하며, Windowviewer 가 다시 시작될 경우에는 해당 태그에 마지막 데이터 값으로 초기화 될 수 있게 합니다. Retentive Value 는 Windowviewer 가 실행 중일 경우에는 기존의 태그 혹은 새로운 태그에서 선택하거나 제거할 수 없습니다.

Retentive Parameters :

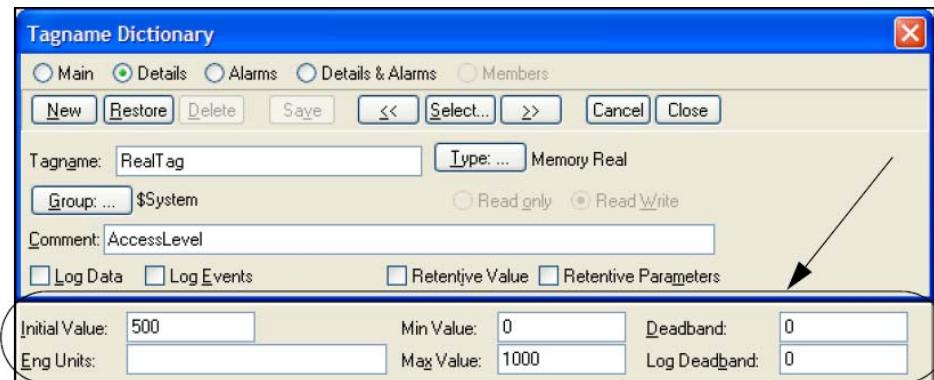
Operator 가 태그의 알람 한계값을 수정하였을 경우, 그 값을 유지하며, Windowviewer 가 다시 시작될 경우에는 그 값이 초기화됩니다.

Note : 변화되는 값은 즉시 반영 되므로, 자주 변하지 않는 데이터값을 가지는 태그들은 위의 두 가지 **retentive** 옵션을 추천합니다.

태그 Details

많은 태그들을 좀 더 다루기 쉽게 하기 위해서는 자세한 설명이 필요합니다. 자세한 설명 타입은

Alarm 타입과 Details 타입으로 정의 할 수 있습니다.



Note : Cancel 또는 close 버튼을 클릭하지 마십시오.

- **Initial Value :** 다른 데이터값이 들어오거나 **retentive** 옵션에 의해 값이 변경되기 전까지는 Runtime 시 Initial Value 값이 입력됩니다.
- **Eng Units :** Engineering Units. 태그 데이터값의 수치에 단위를 달아 놓을 수 있습니다. (kg, W, PSI 등)
- **Min/Max Value :** 태그의 최소/최대 수치 범위

Note : 메모리 태그의 최소/최대 값은 Input 애니메이션 링크를 통해서 정의할 수 있습니다.

- **Deadband :** 태그의 값이 갱신되는 조건을 만족하는 최소값. 보통 데이터값이 자주 업데이트되는 I/O 태그에서 설정 해주는 경우가 많습니다.
- **Log Deadband :** 태그의 값이 이력으로 저장될 경우, 기록될 최소 갱신값.

Other Notes

SuperTag 들의 이름은 TemplateMaker 에서 생성되거나 태그 Details 에 표시되기도 합니다.

Member tag 들은 태그유형으로 정의해야 한다. SuperTAgS 들은 두 단계 중첩 사용 64 멤버 태그들 만큼의 수를 가질 수 있는 것을 주목합니다.

SuperTag 들은 SuperTag TemplateMaker 를 사용해서 만들 수는 없고, 애니메이션 링크의 태그명 부분 또는 Expression 입력 창, Quick Script 를 통해서 생성하거나, 외부파일에서 생성시키는 방법으로 DBLoad Utility 를 사용하는 방법이 있습니다. Type(유형) field 를 참고하면, SuperTag 항목을 볼 수 있을 것입니다.

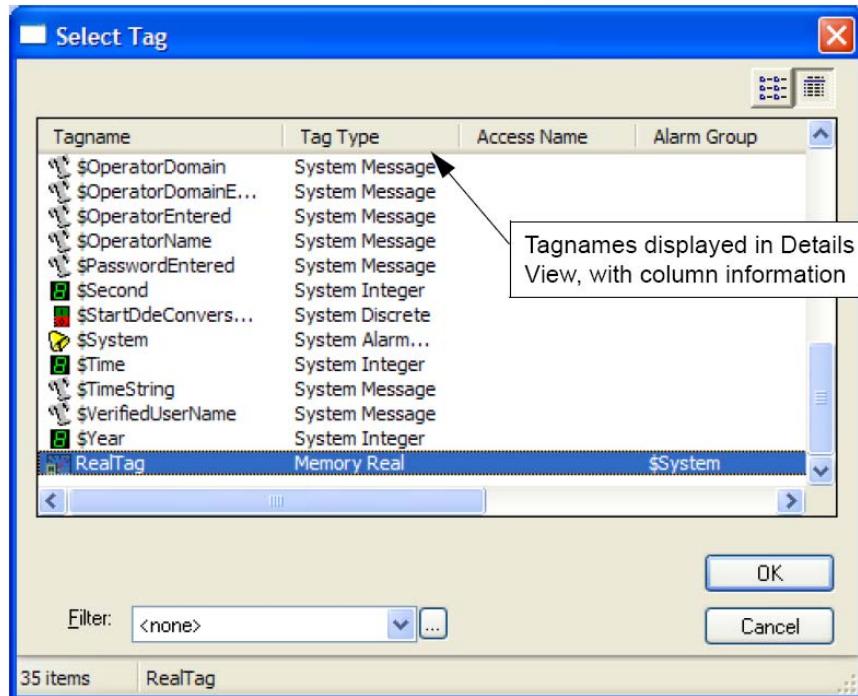
Note : 만일 태그가 Quickscript 안에서 사용 되거나, 객체에 연결이 되어있다면, 그 태그는 수정할 수가 없습니다.

Tag Browser

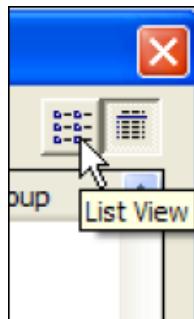
Tag Browser 에서는 Tagname Dictionary 에서 정의된 모든 태그들을 볼 수가 있습니다. Tag Browser 는 사용자에게는 로컬 혹은 원격 노드의 태그들, FactorySuite 어플리케이션의 태그닷필드(.field) 또는 InTouch Tagname Dictionary 를 지원하는 다른 소스의 태그들을 선택하거나 볼 수 있게 해주는 가장 중요한 툴이라 할 수 있으며, 기존의 태그를 선택하거나, 새로운 태그 추가가 가능하며, Tagname Dictionary 의 기본적인 정보를 볼 수가 있게 됩니다.

또한, Tag Browser 는 원격 노드의 태그를 참조하기 위한 태그설정이나 태그를 수정하기 위해 사용되는 Tag Editor 에 접근하기 위해서 사용되기도 합니다. 최초에 Tag Browser 를 열게 되면 기본사항으로써 로컬의 태그들이 선택됩니다. 즉, 로컬 어플리케이션의 Tagname Dictionary 의 태그명이 보인다는 것입니다. 추후, Tagname Dictionary 에 접근을 하게 되면 마지막으로 설정했던 태그명이 나타나게 됩니다.

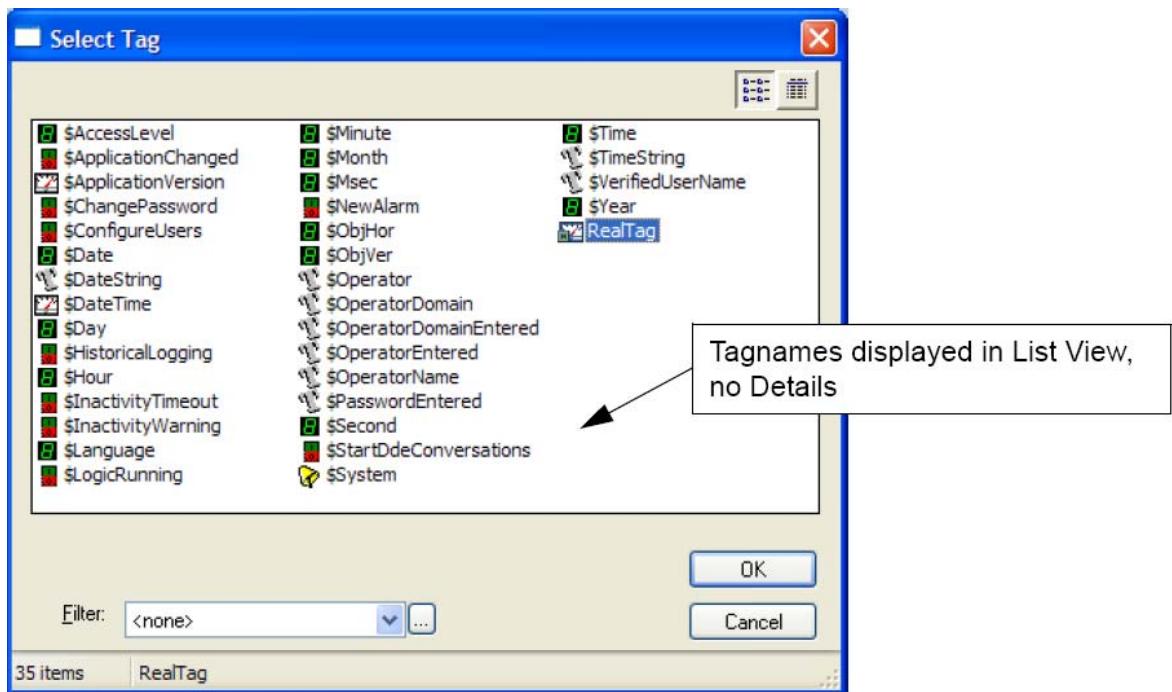
- Tagname Dictionary 창에서 'Select...' 버튼을 클릭합니다.
Tagname Dictionary 에서 Select Tag 창이 그림과 같이 보일 것입니다.



- List View 버튼을 클릭하면, 태그의 Details View 를 숨길 수 있습니다.



Select Tag 창에서 태그의 속성들이 사라지게 됩니다:



필터(Filter)선택 모드

Tagname Dictionary 창에서 Select를 클릭하게 되면, 현재 InTouch Application에서 사용할 수 있고 태그들을 제한적으로 보여줄 수 있는 기능을 가지고 있습니다. 같은 맥락으로 Runtime 동안 이력트랜드 pen에서 새로운 태그를 선택할 수도 있습니다.

- Tagname Dictionary에서 Select 버튼을 클릭합니다.
Tagname Dictionary를 통해 Tag Browser에 접속하면, 리스트에서 태그명을 선택한 후, OK 버튼을 클릭하면, Tagname Dictionary 태그 에디터가 나타나게 됩니다.

필터된 선택 모드에서 Tag Browser를 볼 수 있는 방법:

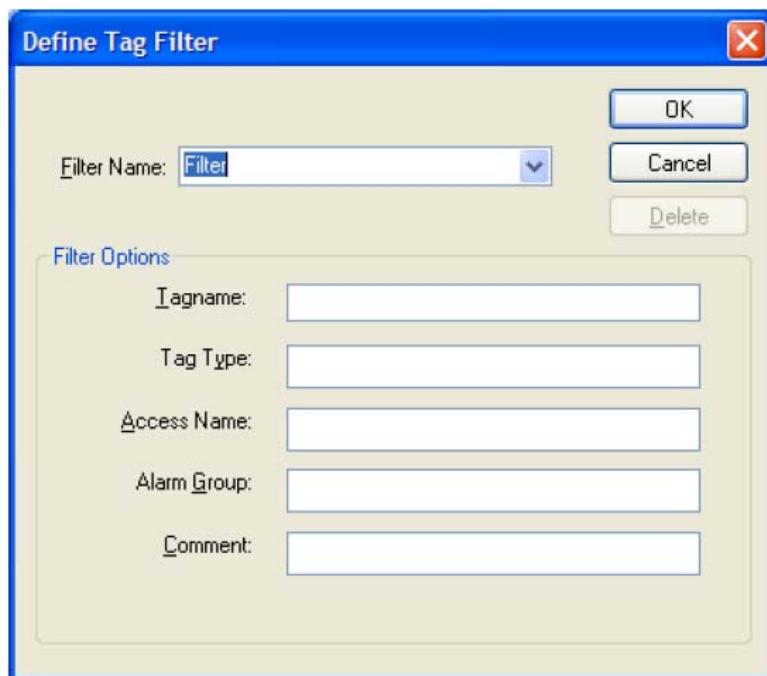
- a. 태그를 추가하거나 수정할 경우, Tagname Dictionary에 Select 버튼을 클릭합니다.
- b. 윈도우 메이커가 실행되고 있는 상황에서, Recipe Manager), 유닛(Unit), Template 정의 부분에 Unit# 열의 셀을 더블 클릭합니다
 - Runtime 시, 이력트랜드 셋업 창의 Pen# 버튼을 클릭합니다.
 - ◆ 이 경우, Tagname Dictionary에서 로그데이터(LogData) 옵션이 체크된 태그들만 Tag Browser에 나타나게 됩니다.
 - ◆ 이 기능은 개발모드에서 Allow Runtime Change 옵션이 이력트랜드에 의해 선택된 경우에만 사용 가능합니다.
 - Runtime 시, HTSelect() 함수와 연결된 객체를 클릭할 경우

태그 브라우저 필터링

브라우저 필터링을 함으로써, 태그명 리스트를 정렬시킬 수 있으며, 어떤 특정한 기준을 만족하는 태그들만을 보여주는 것이 가능합니다. 각 태그들은 태그명, 태그유형, 접속명, 알람그룹, 주석 같은 기준에 의해 정렬이 가능하며, 이 기준들은 하나만 혹은 다중 선택이 가능합니다. 한 번 정해놓은 필터링 기준들은 재사용을 위해 저장하는 것도 가능합니다.

예를 들어, 만약 4 만개의 태그들이 정의되어 있다고 가정 할 때, 특정 접속명에 할당된 약 20 여개의 태그들만 보고 싶다면, Tag Browser에서 그 기준을 만족하게, 접속명 혹은 알람그룹을 필터로써 만들어 주면 되는 것입니다.

- a. Select tag 윈도우 가장 아래 부분에 있는 Define Filter 버튼을 클릭합니다. 
아래와 같은 Define Filter 창이 보일 것입니다:



- **Filter Name :** 정의할 필터의 고유한 필터명을 입력합니다. 입력을 하게 되면, 드롭다운 리스트에 추가한 필터명이 보입니다.
- **Tagname :** 필터로 사용할 태그명입니다. 만약 공란으로 둔다면, 시스템은 필터링을 하지 않게 됩니다.
- **Tag Type :** 필터로 사용할 태그유형입니다. 만약 공란으로 둔다면, 시스템은 필터링을 하지 않게 됩니다.
- **Access Name :** 필터로 사용할 접속명입니다. 만약 공란으로 둔다면, 시스템은 필터링을 하지 않게 됩니다.

- **Alarm Group** : 필터로 사용할 알람그룹입니다. 만약 공란으로 둔다면, 시스템은 필터링을 하지 않게 됩니다.
- **Comment** : 필터로 사용할 주석입니다. 만약 공란으로 둔다면, 시스템은 필터링을 하지 않게 됩니다.
- **OK** 버튼을 클릭하면, 이 창을 마치게 됩니다.

Tag Browser 의 필터 리스트에서 저장한 필터 이름을 볼 수 있습니다. 그리고 언제든지 특정 기준에 의해 필터링된 태그명들을 볼 수 있을 것입니다.

Note : 필터 옵션 필드(태그명, 태그유형, 접속명, 알람그룹, 주석)들은 검색의 범위를 제한하는 특별한 기능도 지원하고 있습니다. 만약 필터를 모두 공란으로 둔다면, 현재 Tag Browser 보이는 모든 태그들이 나타나게 될 것입니다.

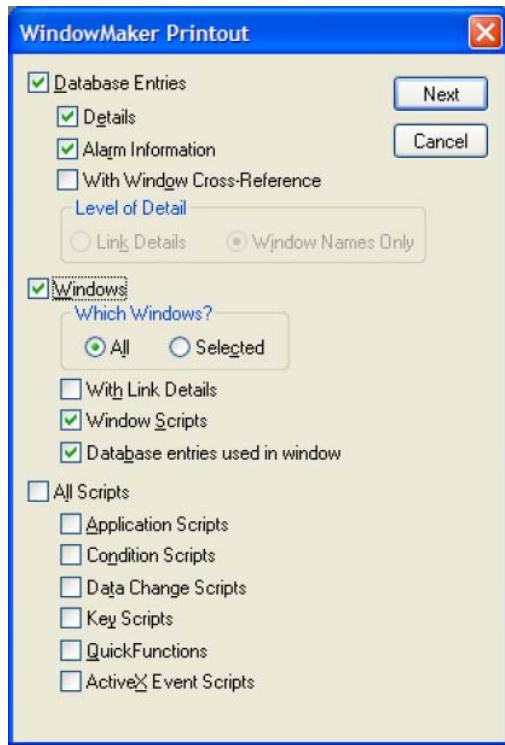
필터링 삭제

- 필터명의 드롭-다운 리스트를 연 다음, 삭제하고자 하는 필터명을 선택 후, Delete 버튼을 클릭하면 삭제됩니다.

Printing the Tagname Dictionary

Tagname Dictionary 의 속성들을 출력함으로써 태그를 사용하는데 있어서 도움을 줄 수가 있습니다. Tagname Dictionary 의 Details 속성들, 알람 정보, 스크립트들 뿐만 아니라, .csv 파일 역시 출력이 가능합니다.

- a. WindowMaker 의 메인 메뉴에서 File/Print 를 클릭합니다.
WindowMaker Printout 창이 나타납니다:



- b. 데이터베이스의 모든 정보를 출력하고 싶다면, Database Entries 를 체크한다. 체크를 하게 되면, 아래와 같은 옵션들이 활성화가 됩니다.

Details : 데이터베이스의 Details 속성들까지 출력됩니다.

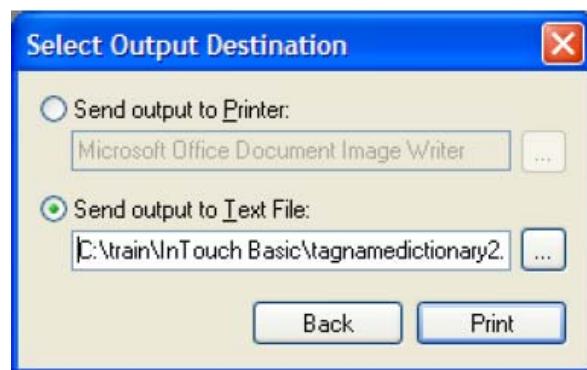
Alarm Information : 데이터베이스의 알람정보들까지 출력됩니다.

- **With Window Cross-Reference** : 데이터베이스(with window cross-reference)의 모든 항목들을 출력합니다.
- **Level of Detail:**
 - **Link Details** 항목은 사용되는 태그의 애니메이션 링크 Detail 을 출력합니다.
 - **Window Names Only** 항목은 오직 cross-reference 윈도우이름만을 출력합니다.
- **Windows** 항목을 체크하게 되면, 어플리케이션 윈도우에서 사용되는 데이터베이스 항목을 출력하게 되며, 아래 항목들이 활성화 됩니다.
- **All** : 어플리케이션의 모든 윈도우의 데이터베이스 항목들을 출력합니다.
- **Selected** : 특정윈도우의 데이터베이스 항목들을 출력합니다. **Selected** 항목을 체크하게 되면, Windows to print 창이 열리게 되고, 출력하려는 윈도우를 선택한 후 OK 버튼을 클릭합니다.(기본적으로 모든 윈도우들이 선택되어 있을 것입니다.)
- **With Link Details** : 윈도우와 연결된 details 를 출력합니다.
- **Window Script** : 윈도우와 관련된 스크립트들을 출력합니다.
- **Database entries used in window** : 윈도우에서 사용되는 태그명을 출력합니다.

- All Script : 선택된 원도우의 모든 스크립트를 출력합니다.
- Application Script : Application 스크립트를 출력합니다.
- Condition Script : 원도우와 관련된 Condition 스크립트를 출력합니다.
- Data Change Script : 원도우와 관련된 Data Change 스크립트를 출력합니다.
- Key Script : 원도우와 관련된 Key 스크립트를 출력합니다.
- Quick Function Script : 원도우와 관련된 Quick Function 스크립트를 출력합니다.
- Active X Event Script : 원도우와 관련된 Active X Event 스크립트를 출력합니다.

c. WindowMaker PrintOut 창의 Next 버튼을 클릭합니다.

Select Output Destination 창이 나타나면, 출력할 최종 노드를 선택해야 합니다.
(프린트 또는 텍스트 파일)



d. 만약 텍스트 파일로 출력을 원한다면, 입력란 옆의 [...] 버튼을 클릭하면
원도우브라우저 창이 나타나게 됩니다.

e. 만약 프린터로 출력을 원한다면, 정확한 프린터 이름을 찾아 입력합니다.

Note : 이러한 출력은 윈도우의 그래픽을 사실적으로 출력하지는 못합니다. 대신,
윈도우의 자세한 설명이나 정보를 출력하기에는 편리한 기능입니다.

LAB 2 – 태그 생성

메모리 태그 생성

아래의 설정에 따라 5 개의 태그를 만듭니다:

- Tagname: RealTag
- Tag Type: Memory Real
- Initial Value: 500
- Min Value: 0
- Max Value: 1000

- Tagname: IntTag
- Tag Type: Memory Integer
- Initial Value: 500
- Min Value: 0
- Max Value: 1000

- Tagname: DiscTag1
- Tag Type: Memory Discrete
- Initial Value: Off

- Tagname: DiscTag2
- Tag Type: Memory Discrete
- Initial Value: Off

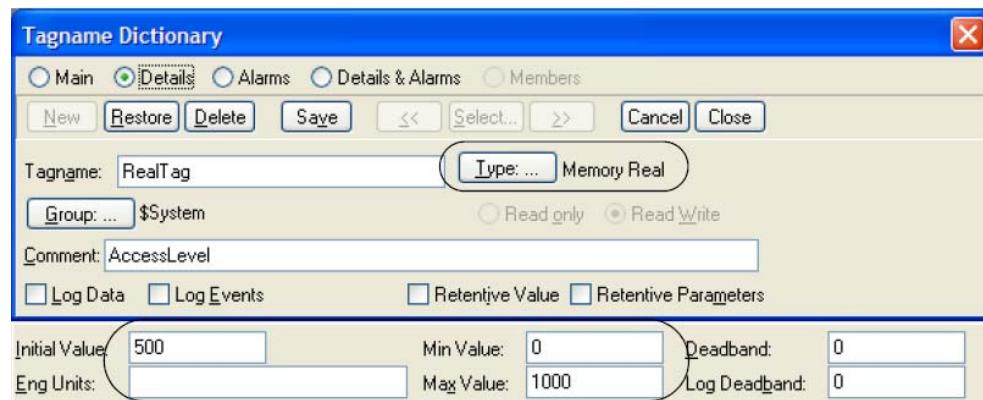
- Tagname: MessageTag
- Tag Type: Memory Message
- Initial Value: Default Value

Real Tag 생성

1. WindowMaker 의 메뉴바에서 Special / Tagname Dictionary 를 클릭
2. Details 옵션을 선택합니다.
3. New 버튼 클릭

Note : 위 내용을 실행하는 동안 이미 Real Tag 가 생성된다면, NEW 버튼을 클릭하지 말고 Select...버튼을 눌러 기존의 RealTag 를 수정 가능하게끔 선택합니다.

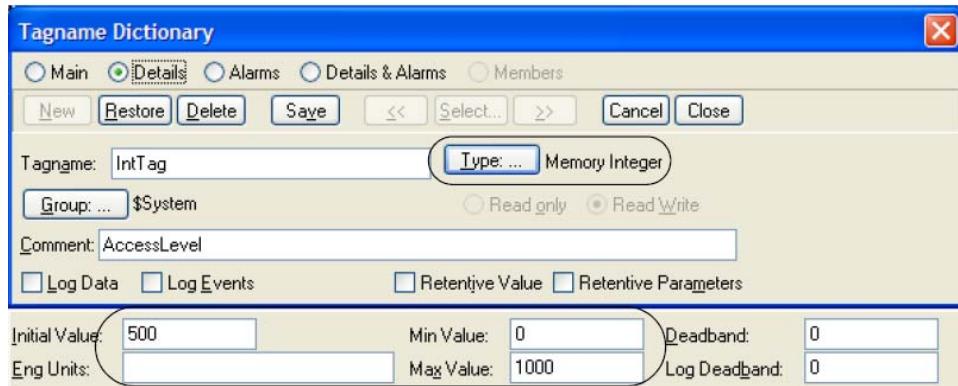
4. Tagname 의 입력란에 RealTag 를 입력합니다.
5. 아래 그림과 같이 details 및 태그 유형을 맞춰줍니다.



6. Save 버튼을 클릭합니다.

Memory Integer 태그 생성

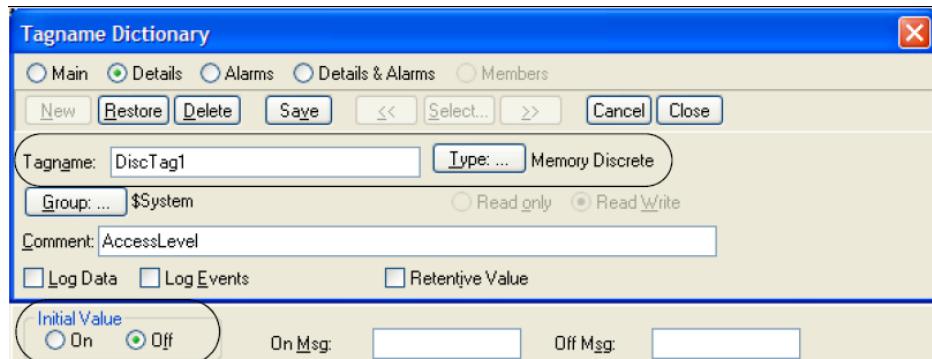
7. New 버튼 클릭
8. Tagname 의 입력란에 IntTag 를 입력합니다.
9. 아래 그림과 같이 details 및 태그 유형을 맞춰줍니다.



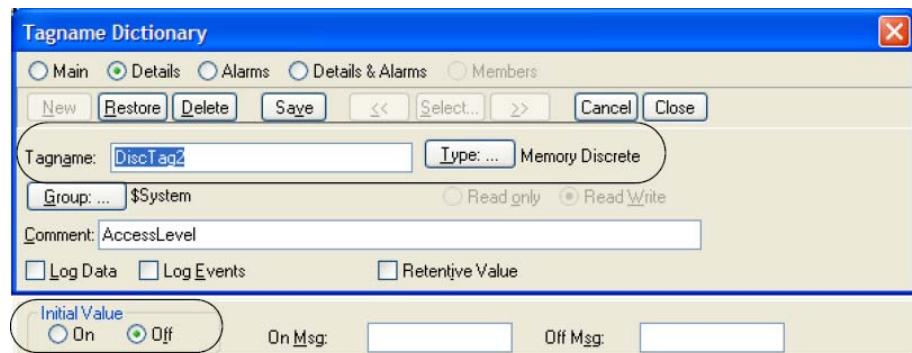
- Save 버튼을 클릭합니다. 태그명이 IntTag인 태그는 메모리 Integer 태그로 정의되고, 생성됩니다.

Memory Discrete 태그 생성

- Tagname Dictionary의 New 버튼을 클릭합니다.
- 아래 그림에서 보여지는 것과 같이 태그의 속성을 설정합니다.



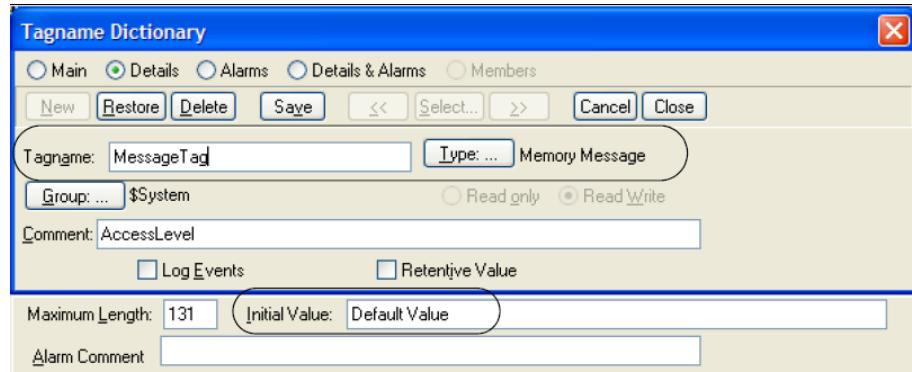
- Save 버튼을 클릭합니다.
 - New 버튼 클릭합니다.
- 태그명은 자동으로 DiscTag2로 변경될 것입니다:



15. Save 버튼을 클릭합니다.

Memory Message 태그 생성

16. 아래의 그림과 같이 이전 단계에서 한 것을 참고로 하여, 속성들을 설정합니다.



17. Save 버튼을 클릭합니다.

18. Close 버튼을 클릭합니다.

Note : 태그 설정값은 Close 버튼을 클릭함으로써 저장이 됩니다.



Module 4

애니메이션 링크

Section 1 – 애니메이션 개체 개요

Section 2 – 애니메이션 개체

Section 3 – 애니메이션 링크

Lab 3 – 애니메이션 링크 설정

Section 4 – SmartSymbol 사용

Lab 4 – SmartSymbol 사용

Section 5 – ArchestraA Symbol 소개

Lab 5 – ArchestraA Symbol 사용하기

Section 1 - 애니메이션 개체 개요

이 단락은 애니메이션 링크 형식들과 같은 기본적인 애니메이션 링크 특징들은 물론 어떻게 태그 브라우저와 태그명 **dot field** 들을 접근 할지를 확인하고 정의할 것입니다.

소개

그래픽 객체들이나 심볼들은 Runtime 시 애니메이션 링크들의 사용에 의해 움직임 효과를 가져옵니다.

애니메이션 링크들은 태그 값 또는 문법 안의 입력을 적용 하기 위하여, 객체나 심볼 모양을 변화 시킵니다..

예를 들어 펌프가 꺼지면 빨간 색이 되고 켜지면 초록색이 되는 펌프 심볼을 만들 수 있습니다. 또한 이 심볼을 마우스로 클릭하면 커짐과 깨짐을 작동 할 수 있는 터치버튼으로 만들 수도 있습니다 .이런 것들과 또 다른 많은 애니메이션 효과는 선택한 개체나 심볼에 적절한 애니메이션 링크를 부여하실 수 있습니다.

InTouch 에서는 기본적인 두 가지 애니메이션 링크를 제공 : Touch Link 와 Display Link

- | Touch Links: 사용자 입력을 허용합니다. Value slider들과 pushbutton 들은 Touch Link 들의 예입니다
- | Display Links: 사용자에게 출력을 허용합니다. Fill Color, Location, Blink link 들은 Display Link 들의 예입니다

대부분의 애니메이션 링크들은 다음과 같은 공통 특징들을 공유합니다.

- | Object Type 대화상자
- | 공통 색상 팔лет트
- | Tag Browser로 빠른 접근
- | Tagname .Fields로 빠른 접근
- | Key 입력 지원 (Touch Links만 가능)
- | 태그명 또는 Expression 입력상자에서 오른쪽 마우스 클릭 지원
(선택된 표현이 적용 되어 지는 메뉴와 명령들을 보여줍니다)

Object Type 대화상자

Object Type 대화상자는 animation link selection dialog box 위에 나타납니다. 이것은 보통 모든 링크들에 만들어집니다. 이것은 애니메이션 링크 연결을 위해 선택된 객체 형식의 설명을 보여줍니다.



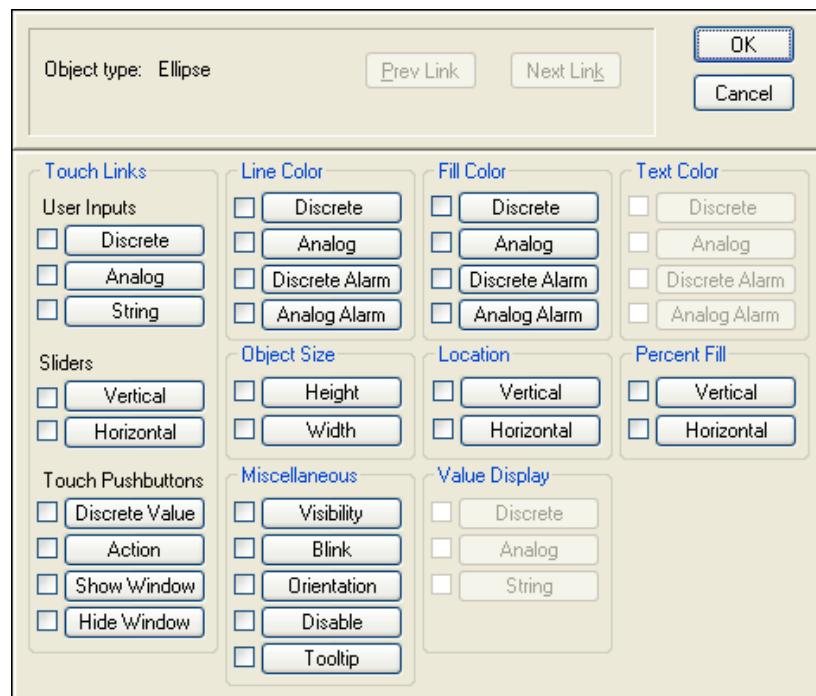
만일 다중 링크들이 하나의 객체에 연결되어 있다면 Prev Link 와 Next Link 는 객체에 연결된 각 링크를 위한 링크 대화 상자를 통해 quickly scroll forward 또는 backwards 가 허용됩니다

Note: 링크들은 그들이 부착된 원래 객체에 의해 저장된다.

애니메이션 링크 선택 대화상자

다중 링크는 여러분의 객체들이나 심볼들, 어떠한 화면 애니메이션 효과를 적용한 개체를 위해 정의 될 수 있습니다.

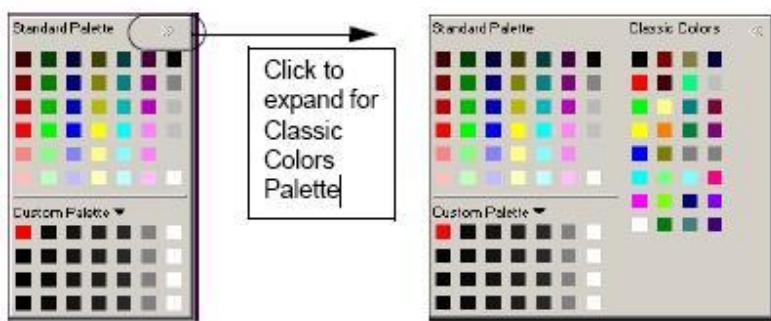
- a. 타원 객체를 스크래치 윈도우에 추가합니다.
- b. 스크래치 윈도우에서 타원 객체를 더블 클릭합니다. 애니메이션 링크 선택 대화상자가 나타납니다. 이것은 모든 객체와 그래픽 애니메이션 형식의 설정에 사용됩니다.



- c. 대화상자를 종료하기 위해 Cancel를 클릭합니다.(나중에 랩에서 사용 할 것입니다.)
- d. Cancel 클릭의 결과로 타원 객체는 수정 없이 유지됩니다.

Color Links 적용

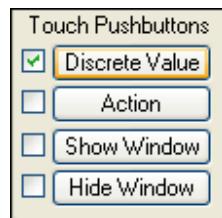
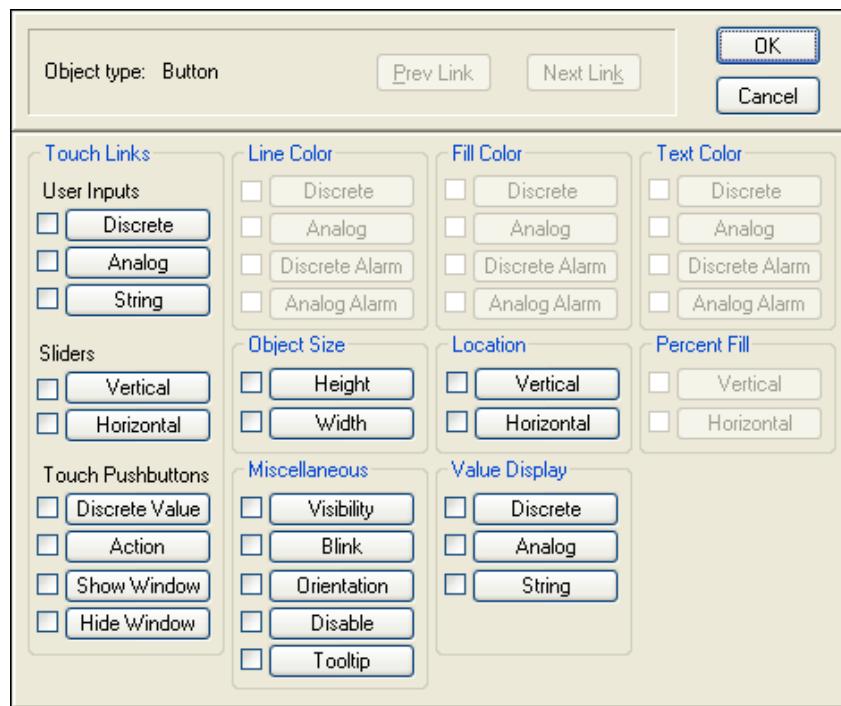
색은 선의 동적 속성, 사각형, 라운드 된 사각형, 타원, 다각선, 다각형, 문자에 적용 될 수 있습니다. Color Palette 는 태그명의 값이나 태그명의 알람 상태, 그 중의 특징에 연결된 색을 선택하는데에 사용됩니다 또한 사용자 정의 Palette 를 만들 수 있습니다. Color 링크가 객체나 심볼에 부착되어 있을 때 Color Palette 를 열기 위해 링크 대화상자 안의 색을 클릭합니다. Palette 안의 원하는 색을 클릭합니다. Palette 를 닫으면 Color 상자 안에 선택된 색이 나타날 것입니다.



Tag Browser 접근

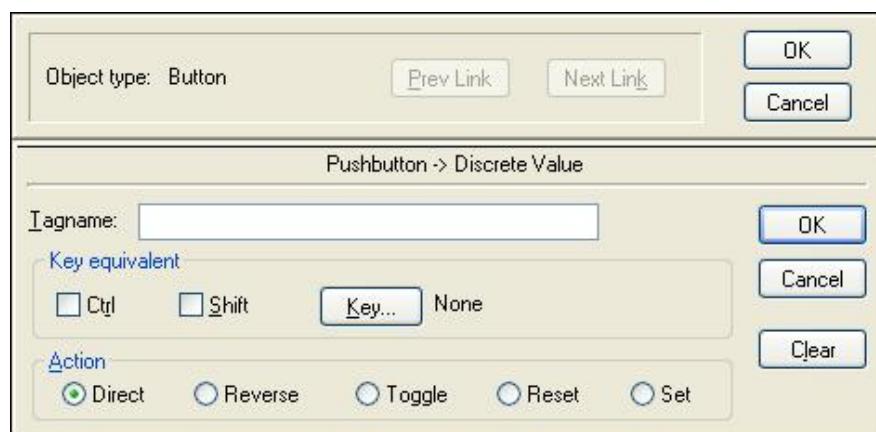
애니메이션 링크를 생성할 때, 어플리케이션에 정의 되어 있는 모든 태그들을 빠르게 볼 수 있는 Tag Browser 를 액세스 합니다. Tag Browser 에 지정되어있는 태그를 선택하게 되며, 자동으로 Tagname 이나 Expression box 안에 쓰여지게 됩니다.

- a. 스크래치 윈도우 안에 버튼 객체를 더블클릭 합니다.



- b. Touch Pushbuttons/Discrete Value 버튼을 클릭합니다.

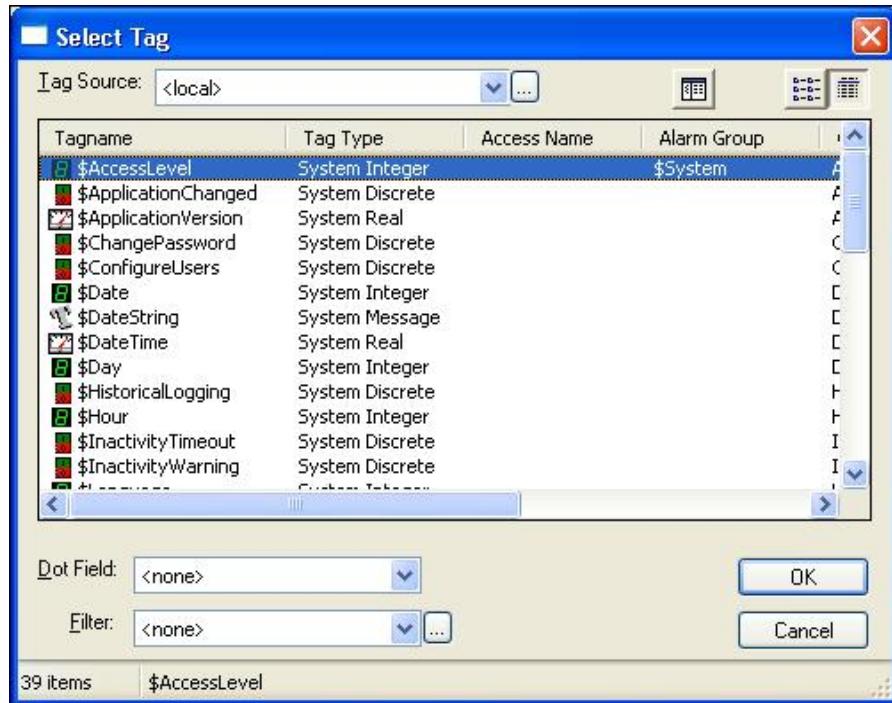
Object Type 대화상자가 나타납니다.



- c. Tagname 필드 안의 텍스트를 제거하기 위해 Clear 버튼을 클릭합니다.

- d. Tagname 필드의 공란을 더블클릭 합니다.

Tag Browser 가 나타납니다.



- e. 원하는 태그를 선택하고 OK버튼을 클릭합니다. 태그는 Tagname 또는 Expression 상자안에 삽입되었을 것입니다.

Note : 이 애니메이션, Discrete Value 가 선택되기 위해 discrete 태그를 요구 합니다.

Cutting, Copying, Pasting, and Clearing 객체 링크들

WindowMaker's link paste buffer 는 객체로 부터 잘라내거나 복사 할 수 있는 임시 저장 지역, 저장 링크들입니다.(버퍼는 오직 최근 잘라내거나 복사 행동만을 위한 링크를 저장합니다) 이 연결들은 어떠한 객체나 심볼에 붙여질 수 있는 link paste buffer 안에 저장됩니다. 붙여진 링크가 객체에 식별 값을 가지고 있지 않다면(e.g., text 객체위의 line color link), 그 연결은 붙여진 것이 아닙니다.

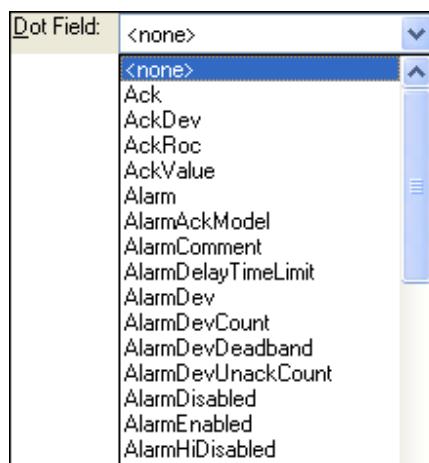
- 객체를 선택합니다.
- WindowMaker로부터 Edit/Links를 클릭하고 Cut, Copy, Paste 또는 Clear를 선택합니다.

Accessing Tagname Dot Fields

Dot Field는 태그의 이름, 값에서 더 나아가 태그에 입력된 특정 값을 포함합니다.

.field와 선택된 태그명을 사용하기 위해 **Dot Field** 드롭다운 리스트를 열고 .field를 선택하고 OK를 클릭합니다. Dot Field 리스트는 애니메이션 링크 설정 또는 스크립트를 수정하는 동안 tag browser를 통해 접근됩니다.

아래의 그림은 **Dot Field**를 보여줍니다 : tag browser 안의 드롭다운 리스트



태그의 데이터베이스 정의를 보여 주기 위해, **Tagname** 안의 태그명을 위치하거나 **Object Type** 대화상자의 **Expression** 상자를 위치에서 더블 클릭합니다.

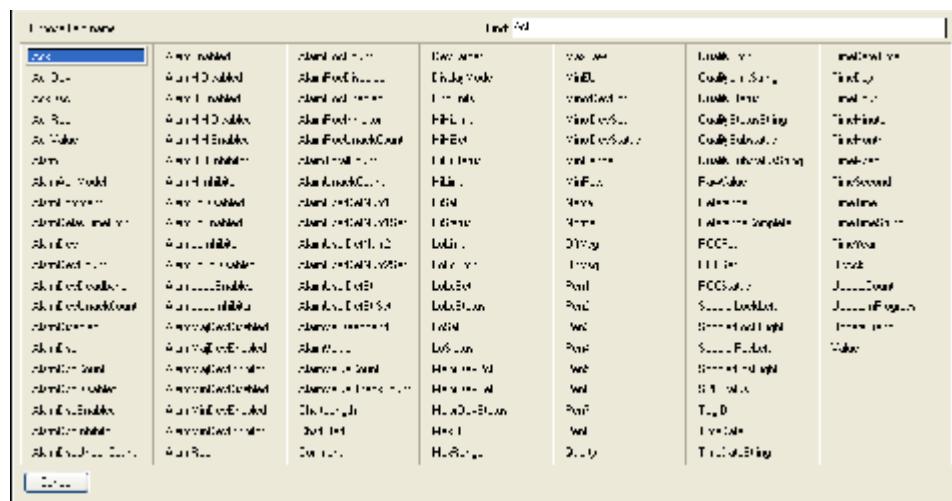
Tagname Dictionary 대화상자는 태그명의 정의를 보여 줄 것입니다.

Accessing Tagname Dot Fields Through the Choose Field Name

- Scratch** 원도우에서 **타원** 객체를 더블클릭합니다. 애니메이션 링크 선택 대화상자가 나타납니다.
- Miscellaneous / Visibility** 버튼을 클릭합니다. **Object type** 대화상자가 나타납니다.
- Expression** 필드안의 **\$AccessLevel**에 위치시킵니다.(마침표 포함)



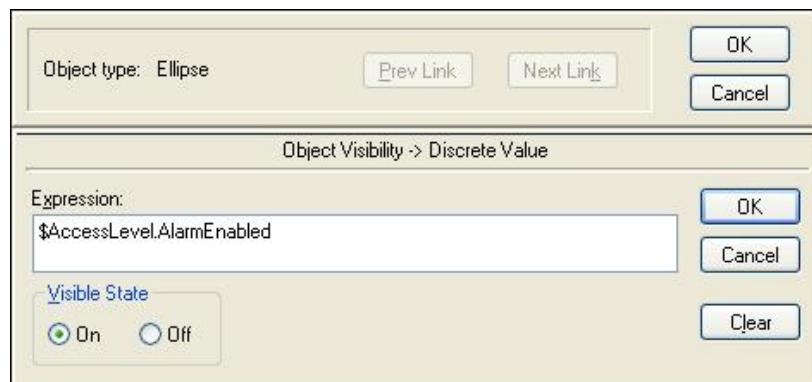
- d. Expression안의 오른쪽(.)를 더블클릭합니다. : choose filed name... 대화상자가 나타납니다.:



이 대화상자는 모든 tagname.fields를 보여줍니다.

모든 tag들의 모든 .fields가 명확한 것은 아닙니다.

- e. AlarmEnabled .field를 선택합니다.



대화상자가 닫힐 것이고 선택된 .field는 자동적으로 Tagname 또는 Expression 입력

상자에 추가되었습니다.

- f. Object type 대화상자를 종료하기 위해 상단의 Cancel 버튼을 클릭 합니다.
- g. Cancel 클릭의 결과와 같이, 타원 객체는 수정 없이 유지 됩니다.
(이 부분에서는 수정없는것이 요구됩니다. 다음 랙동안 바꾸는 것들을 실습할 것입니다.)

Section 2 - 애니메이션 객체

이 단락은 InTouch 애니메이션 링크 옵션들을 설명합니다. 또한 window 가져오기와 animations across windows 공유를 설명합니다.

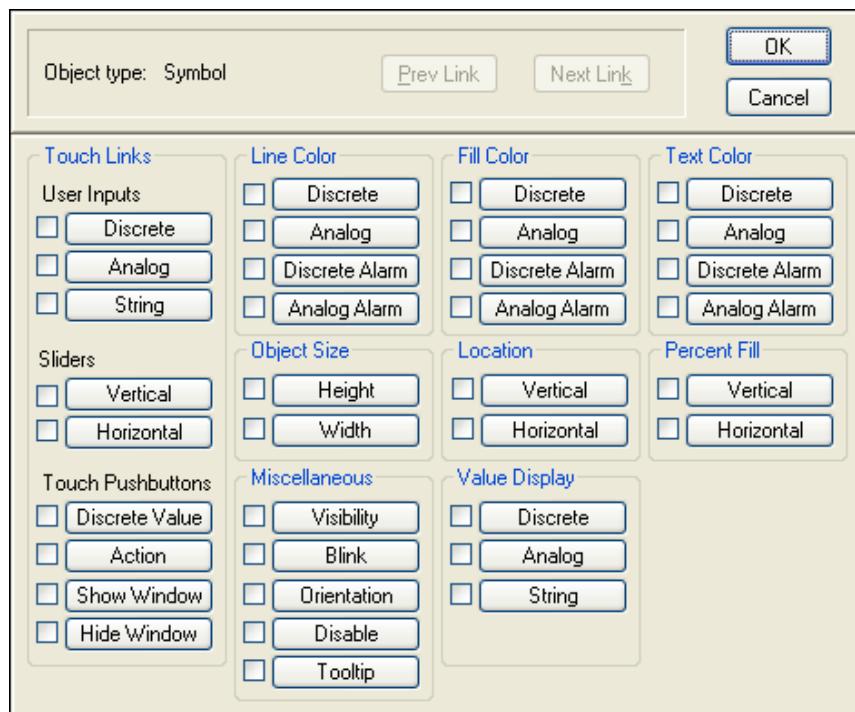
소개

아래의 정보는 Intouch 애니메이션 옵션들을 설명하고 있습니다. 교사가 다양한 애니메이션 링크 옵션들을 설명할 것입니다.

이 단락의 결론에서 교육을 받는 동안 pre-configured InTouch window 를 가져 올 것입니다. 이 window는 다음 랙들에서 animation links across windows 를 설정하고 공유하는데 사용됩니다.

The Animation Links Selector Dialog Box 사용

애니메이션 링크들은 Animation Links Selection dialog box 로 부터 설정됩니다.



- | 링크 버튼들은 선택된 객체 링크를 연결합니다.
- | 링크가 선택된 객체에 적용할 수 없으면 버튼은 사용할 수 없습니다.

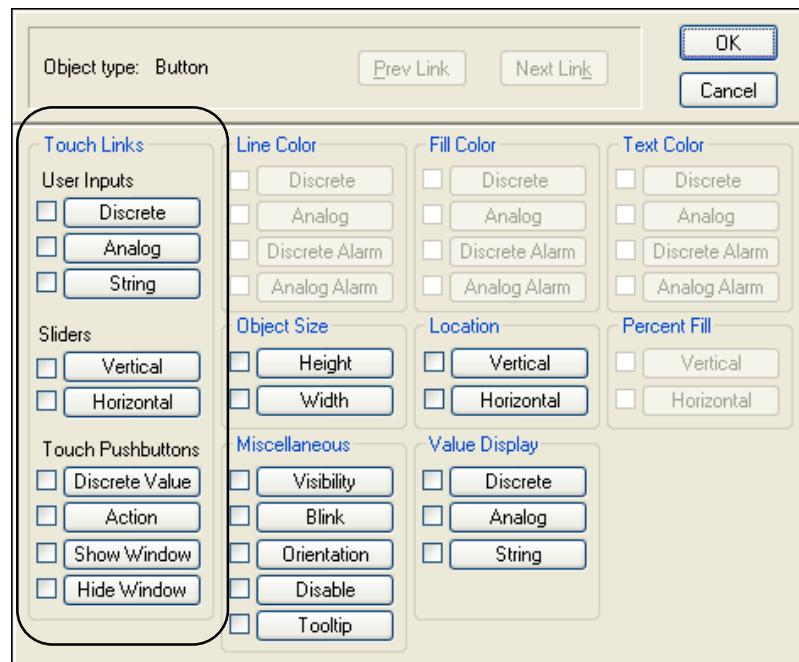
- | 설정된 링크안에서 링크 체크 박스는 결과를 나타내지 않습니다.
- | 링크를 선택하기 위해 링크명 버튼을 클릭하고 **Detail Definition** 대화상자가 열립니다.
- | 선택된 객체로 부터 설정된 링크를 제거하기 위해 링크 체크박스의 체크를 해제합니다.
- | 이 체크박스는 링크이름버튼을 클릭하고 링크설정을 허용 할때 자동적으로 선택됩니다.

Note: 버튼을 클릭하지 않는다면 링크기본정의를 수정할 수 없을 것입니다.

- | 객체에 대한 모든 링크들을 수락하기 위해 **OK**버튼을 클릭하고 대화상자를 닫습니다.
- | 애니메이션 링크들이 생성될때 애니메이션 링크의 **Tagname**안에 **tagname0** 들어가거나 **Tagname Dictionary**안에 **Expression field**가 벌써 정의되어있을 것입니다.

그것이 정의 되지 않으면 태그를 정의하기 위해 메세지 박스가 나타나 태그정의를 요청할 것입니다. **Yes**를 클릭하면 **Tagname Dictionary** 가 나타날 것이고 태그는 정의될 수 있습니다.

Touch Links



Touch Link 들은 Runtime 시에 그들의 "touch-sensitive"(i.e. active)를 만들기 위한 객체들이나 심볼들을 사용합니다. Touch Links 들은 사용자의 입력을 허용합니다. 이 입력의 예로는 뱀브를 여닫기, 새 경보 장치 넣기, 복잡한 논리 스크립트 실행하기 그리고 텍스트 문자열 사용하기 등이 있습니다. Touch Link 들은 Runtime 시에 쉽게 확인됩니다. 커서가 지나칠 때 음영이 touch-sensitive object 를 둘러쌓습니다. TAB 키는 touch-sensitive object에서

Touch-sensitive object 로 이동하기 위해 사용될 수 있습니다.

touch link 객체나 심볼이 각각 다른 상단의 놓여지는 text object 를 포함한다면, 상단의 text object 는 데이터 값을 보여주기 위해 사용될 것입니다.

클릭과 객체 터치(터치스크린을 사용할 때), 객체가 활성화되어 할당된 키를 누르거나 엔터키를 누름으로 touch-sensitive pushbutton 를 활성화 시킵니다.

Note: 객체나 심볼이 텍스트 필드들을 가지고 있는 링크들(3-D button 들의 예외를 가진)을 위해 사용되면 텍스트 객체가 WindowViewer 에 나타날 때 모든 속성들은 일반적으로 텍스트(정렬, 스타일, 폰트)에 설정되고 적용됩니다.
텍스트 필드가 입력 값에 사용될 때, 태그명 수정 대화상자의 각각의 태그 상세 지역의 Input Only 옵션을 활성화하지 않는다면 출력 값 또한 보여질 것입니다.

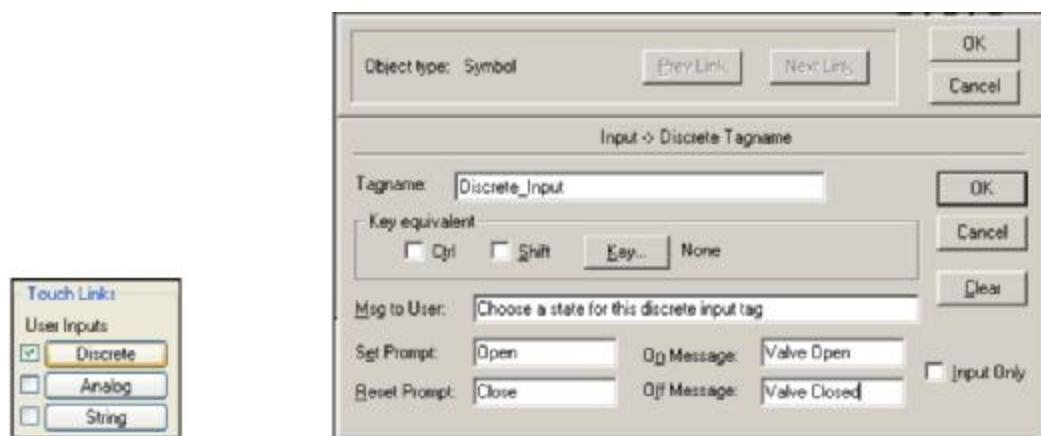
User Inputs

이 연결들로 사용자의 입력을 시스템이 받아들일 수 있는 터치에 반응하는 객체를 만듭니다. 이 상태를 수정하기 위해 사용하는 푸쉬버튼, 아날로그 값 또는 보안 로그온 등이 예입니다.

User Inputs	설명
Discrete	이상태그의 값을 제어하기 위해 사용됩니다. 연결이 활성화되면, 다이얼로그박스가 나타나 여러 가지 선택을 하게 됩니다
Analog	아날로그(정수 또는 실수) 태그명의 값을 입력하기 위해 사용하는 객체를 만들기 위해 사용하는 값입니다. 이 연결이 활성화되면 입력 상자가 나타나고 키보드에서 값을 입력할 수 있습니다. 옵션에 따라 스크린상에 키보드가 나타나게도 할 수 있습니다.

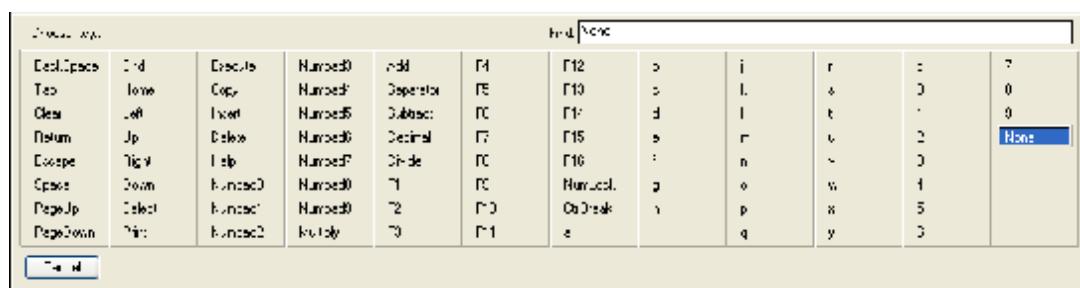
String	<p>이 연결은 문자열 메시지를 입력할 수 있는 객체를 만들기 위해 사용합니다. 이 연결이 활성화되면 메시지 값을 입력할 수 있게 입력 상자가 나타납니다.</p>
---------------	--

User Inputs – Discrete



Tagname: 이산 유형의 태그명을 Tagname 필드 안에 입력합니다.

Key equivalent area: 이 연결에 대응할 키를 할당합니다.



Note: CTRL 또는 SHIFT를 선택하던지 대응하는 키를 누를 때 그 키들을 유지하는 연산이 필요합니다.

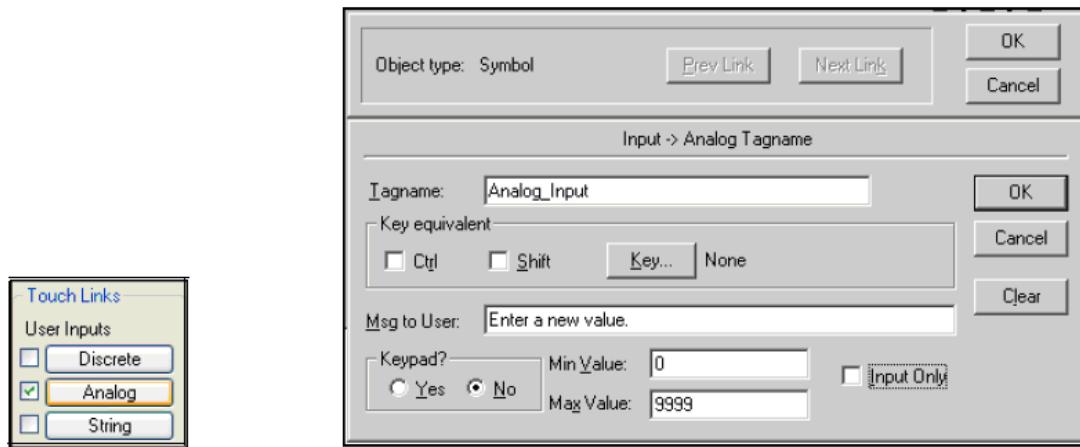
Msg to user: 입력 링크가 활성화되었을 때 입력 대화상자에 나타날 메시지를 입력합니다.

Set Prompt and Reset Prompt: 입력 대화 상자에서 이산 값을 **On/Off** 하기 위해 사용하는 버튼 위에 나타날 메시지를 각각 필드에 입력합니다.

On Message and Off Message: 객체와 연관된 텍스트 필드가 있으면 그 필드에 나타날 On 또는 Off의 메시지를 각각 필드에 입력합니다.

Input Only: 객체와 연관된 텍스트 필드에 온/오프 메시지가 나타나지 못하게 합니다. 이 옵션은 텍스트 필드를 수반하는 모든 객체에 적용 가능합니다.

User Input – Analog



Note: 텍스트 필드가 이 링크에 사용되면, 그것은 정확한 순서대로 아날로그(정수 또는 실수) 출력값을 표현을 적절하게 형식화 해야 합니다.

Tagname: 아날로그 유형의 태그명을 필드에 입력합니다.

Key equivalent area: 이 연결에 대응할 키를 할당합니다.

Msg to User: 입력 연결이 활성화 될 때 나타나는 입력 디아일로그 박스에 나타날 메시지를 필드에 입력합니다.

Keypad? Area: YES: 문자열의 새로운 값 입력을 위해 숫자 키패드를 보여줍니다..

Note: 터치스크린을 사용하고 사용자가 키보드 입력을 못한다면, KEYPAD에서 “YES”를 선택해야 합니다.

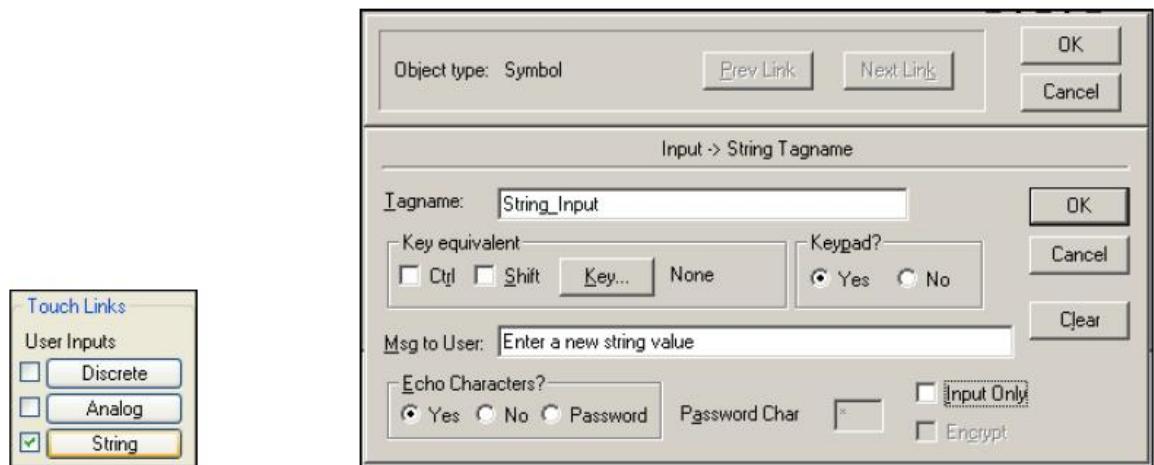
Min Value: 태그가 사용할 최저 허용치를 입력합니다.

Max Value: 태그가 사용할 최고 허용치를 입력합니다.

Min/Max 값은 실제 태그값안에 입력을 할지라도 작동하는 값 범위를 제한합니다.

Input Only: 객체와 관련된 문자 필드안에 입력을 허용하지 않습니다. 이 옵션은 3 dimensional 버튼과 같은 객체와 관련 있는 문자 필드를 가진 객체에 적용 됩니다.

User Inputs – String



Tagname: 메시지 유형 tagname을 입력합니다.

Key equivalent area:: 연결에 대응할 키를 할당합니다.

Keypad? area: Yes: 새로운 문자열 입력을 위해 키보드를 보여줍니다.

Msg to User: 연결이 활성화 될 때 키패드가 보여지면 사용자에게 나타날 메시지를 필드에 입력합니다.

Echo Characters? 필드의 Yes를 선택하면 입력 문자열이 입력된대로 화면에 나타나고 No를 선택하면 암호와 같이 보안이 필요한 데이터를 나타나지 않게 할 수 있습니다.

Password Char: 원하는 형태의 숨김 문자를 사용 할 수 있습니다..

Input Only: 입력 값이 객체에 연관된 텍스트 필드에 나타나지 못하게 합니다 문자 필드를 가진 모든 객체에 적용됩니다.

Note: WindowViewer가 실행되었고, 태그 연결을 객체에 정의할때 초기값 상자안에 입력한 텍스트가 보여질것입니다.

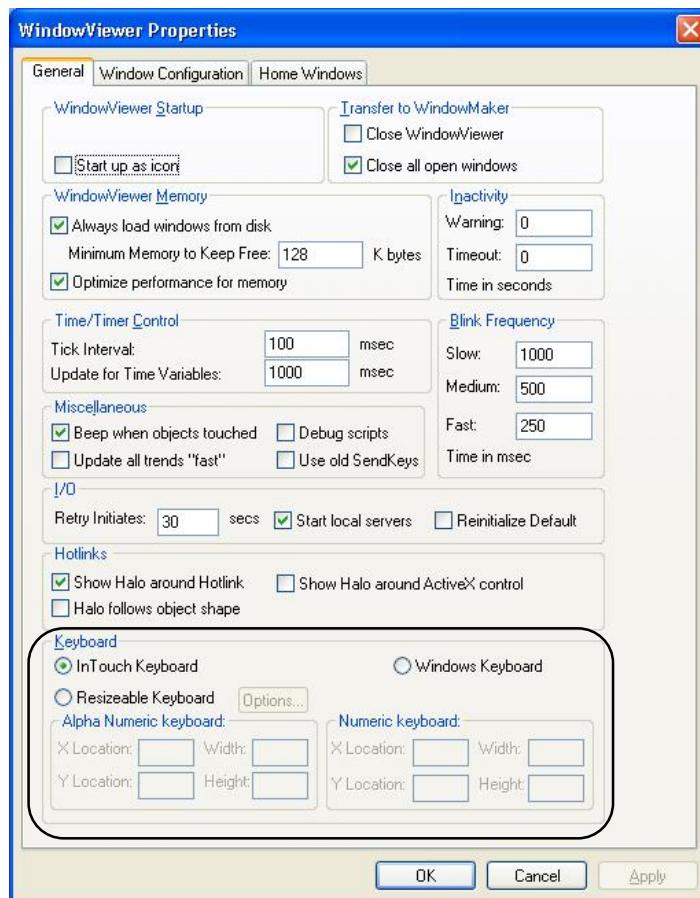
Encrypt: 로그온 대화상자에 입력을 위한 태그를 위해 체크 않아야 합니다. 태그가 암호화가 되었고 보안을 위해 암호를 사용했다면 그것은 작동하지 않을 것입니다

Tip: 추가적인 보안을 위해 `StringCompareEncrypted()`함수를 이용하여, 암호화된 문자열들을 비교할수 있습니다.

On-Screen Keyboards (and Keypads)

InTouch에서 세 가지 다른 키보드들을 지원합니다. : 기존의 InTouch 키보드, 폰트 설정을 지원하는 새로운 Wonderware resizeable 키보드, 마이크로소프트가 지원하는 Windows keyboard.

on-screen keyboard들은 WindowViewer 속성 대화 상자의 General tab에서 설정됩니다.

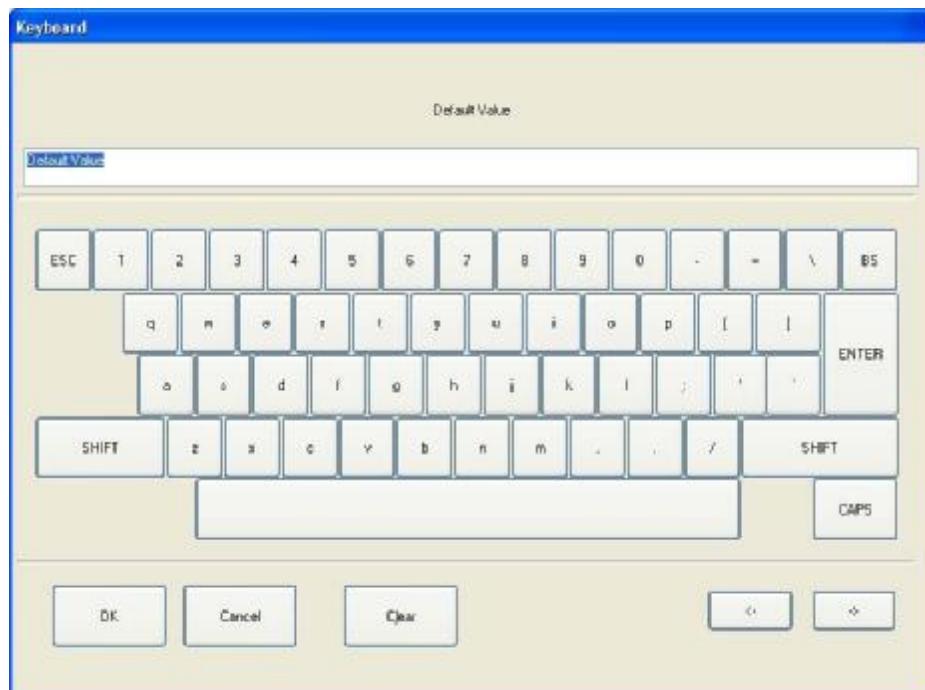
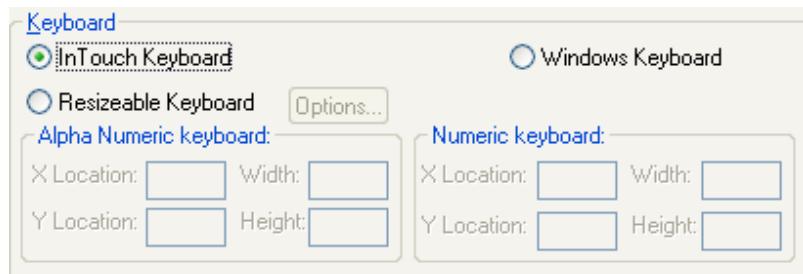


Note: 사용자 입력 아날로그 링크들을 위해 키보드를 선택할 뿐만 아니라 숫자 키패드도 적용합니다. 예를 들어 새로운 Wonderware resizeable keyboard를 선택하면 그 키패드는 크기 조정할 수 있을 것입니다. Windows keyboard를 선택하면 그것이 기본 Windows keyboard에 포함되어 있기 때문에 키패드가 나뉘짐 없이 사용될 것입니다.

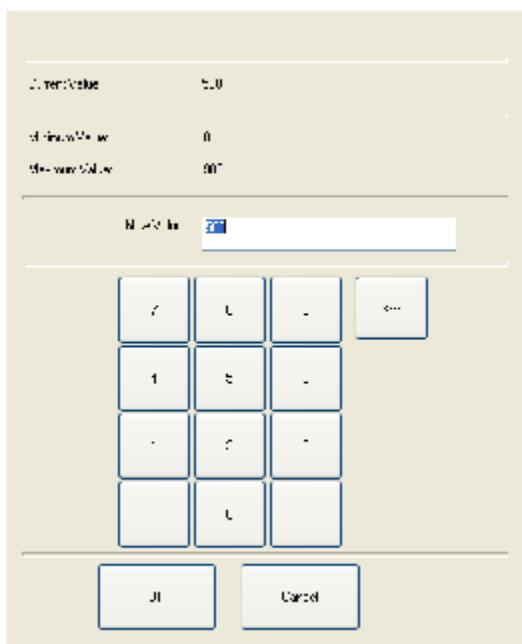
스크린 키보드를 사용하기 위해 아래와 같은 단계를 따릅니다 :

- a. Special / Configure / WindowViewer / Generaltab에서 키보드 설정을 합니다.
- b. 원도우에서 문자 객체를 생성합니다.
- c. Touch Link / User Inputs / String의 애니메이션 링크를 할당합니다.
- d. Tagname and Keypad? Yes에서 애니메이션과 메모리 메세지를 설정합니다.
- e. 실행합니다.
- f. 기본 문자 객체를 클릭합니다. 키보드가 나타납니다

InTouch Keyboard

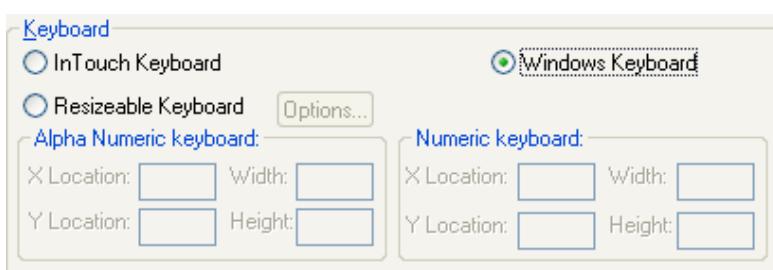


InTouch Keypad



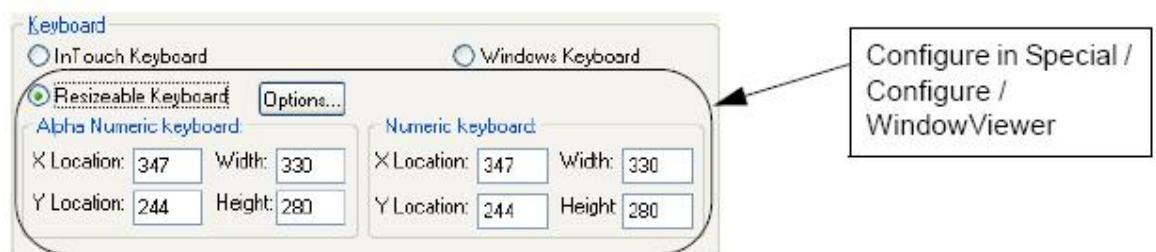
Windows Keyboard

Windows keyboard 사용의 이점은 국제적인 설정을 지원하는 것입니다. 그러나 크기조정이 되지 않아서 아마 터치스크린에 너무 작을지도 모르고 분리된 숫자키패드를 지원하지 않습니다.

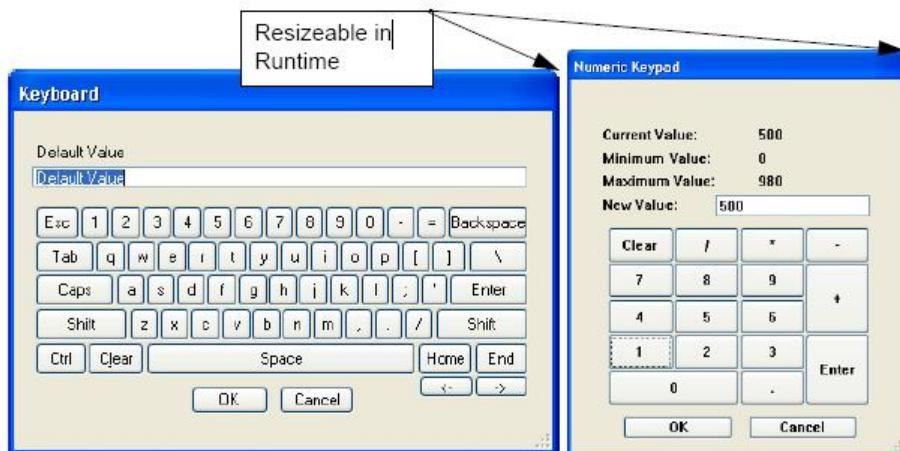




Resizeable Keyboard and Keypad



Resizeable Keyboard 설정이 키보드가 사용 될 폰트를 정의 할 수 있는 **Font** 대화상자를 열 때 **Options** 버튼을 클릭합니다.



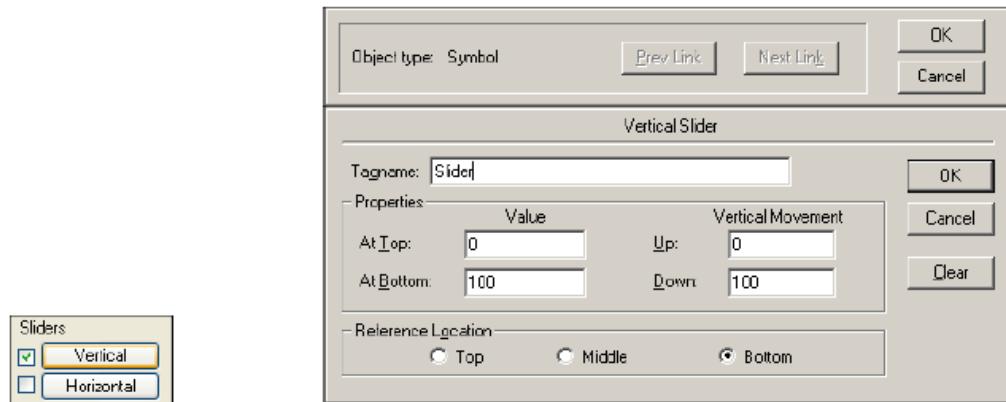
Sliders

이 연결은 화면에서 마우스나 다른 포인팅 장비((e.g., a finger on a touch screen)로 움직일 수 있는 객체나 심볼을 만들기 위해 사용합니다.

객체나 심볼이 이동할 때 거기에 연결된 태그의 값이 변화합니다. 이 연결은 시스템 설정 변수를 위한 장치를 생성하기 위해 슬라이드 기능을 제공합니다.

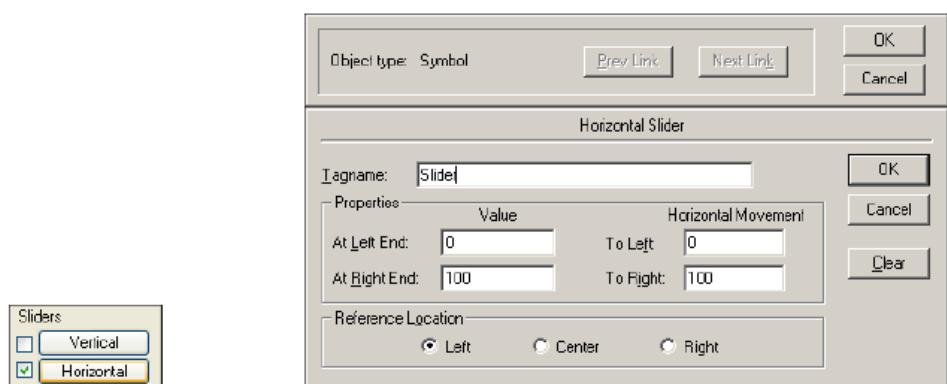
객체는 horizontal slider touch link, vertical slider touch link 또는 두 가지 모두를 가질 수 있습니다. 한 객체에 두 연결을 같이 사용함으로 두 아날로그 태그의 값을 동시에 바꿀 수 있습니다.

Vertical and Horizontal Sliders



At Top / At Bottom: 슬라이드가 가장 먼곳에 있을 경우의 **tagname** 값을 입력합니다.

Up / Down: 슬라이드가 움직일 수 있는 픽셀의 숫자를 입력합니다.



At Left End / At Right End: 슬라이드가 왼쪽과 오른쪽의 가장 먼곳에 있을 때의 **tagname** 값을 입력합니다.

To Left / To Right: 슬라이드가 움직일 수 있는 픽셀 숫자를 입력합니다.

Tagname: 아날로그(정수 또는 실수)태그명을 입력합니다.

Reference Location: 커서로 잡아서 움직일 포인터 심볼의 위치를 필드에서 선택합니다.

Tip: 객체를 이동시키기 위한 픽셀의 숫자 수치를 나타내기 위해 **Ruler**를 클릭합니다.

Touch Pushbuttons

Touch Pushbutton 이 연결은 마우스나 터치스크린을 클릭할때 연산들을 즉시 수행하는 객체 연결을 만들기 위해 사용됩니다.

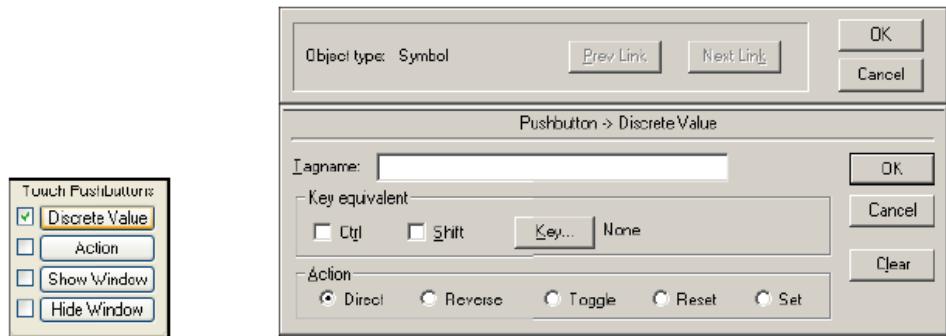
이 연산들은 이사나 값 수정, 액션 스크립트 적용, 윈도우 숨기기와 보여주기 등이 있습니다.

터치스크린 링크들의 4가지 유형이 있습니다.

사용자 입력	설명
Discrete Value	이산 태그명의 상태를 조정하는 푸쉬버튼안에 객체나 심볼을 만들기 위해 사용. 버튼의 유형들은 직접 설정, 재설정, 전환, 일시적 온(직접) 또는 일시적 오프(반전)등입니다.
Action	이 연결은 객체, 심볼 또는 버튼에게 13가지 다른 스크립트 제어를 입력 할 수 있는 연결입니다. 스크립트 태그를 특정한 값으로 설정하기 윈도우를 보여 주거나 숨기기 그리고 다른 어플리케이션을 시작하고 제어하기, 함수 실행하기 등에 사용합니다.
Show Window	이 연결은 객체나 심볼을 클릭함으로 하나 이상의 윈도우가 나타나게 하는 푸쉬버튼을 만들기 위해 사용합니다.

Hide Window	<p>이 연결은 객체나 심볼을 클릭함으로 하나 이상의 윈도우가 닫히게 하는 푸쉬버튼을 만들기 위해 사용합니다.</p>
--------------------	---

Touch Pushbuttons – Discrete Value



Tagname: 이산 태그명을 입력합니다.

Key equivalent area: 애니메이션 링크가 활성화 되게 키보드에 특별한 키를 할당할 수 있습니다. 객체와 링크가 보이고 선택되었을 때 사용합니다.. 객체가 Visibility 또는 Disable 연결이라면 객체가 보이지 않거나 활성화 되지 않을 때 작동하지 않습니다.

다중 윈도우에서 같은 키로 정의 할 수 있습니다. 그러나 이 정의는 가장 최근 열린 윈도우에서 작동할 것입니다.

Overlay 윈도우의 경우에 윈도우 상단에서 작동 할 것입니다.

Note: 활성화된 윈도우에서 객체나 action pushbutton이 Key Action 스크립트에 사용된 같은 키가 할당되었다면, 활성화된 윈도우에서 대응 key가 Key Action 스크립트의 실행 보다 앞서 수행 될 것입니다.

Key: 연결에 지정할 키를 할당합니다.

Action area:

- **Direct:** 푸쉬버튼이 클릭되어지고 그대로 있으면 값을 **1(True,On,Yes)**로 설정하고 그 버튼이 놓아지면 자동적으로 **0(False,Off,No)**으로 재설정됩니다.
- **Reverse:** 푸쉬버튼이 클릭되어지고 그대로 있으면 값을 **0(False,Off,No)**으로 설정하고 그 버튼이 놓아지면 자동적으로 **1(True,On,Yes)**로 재설정됩니다.
- **Toggle:** 클릭될 때 이산 태그명 상태를 바꿉니다. 만약 버튼이 클릭될 때 태그명이

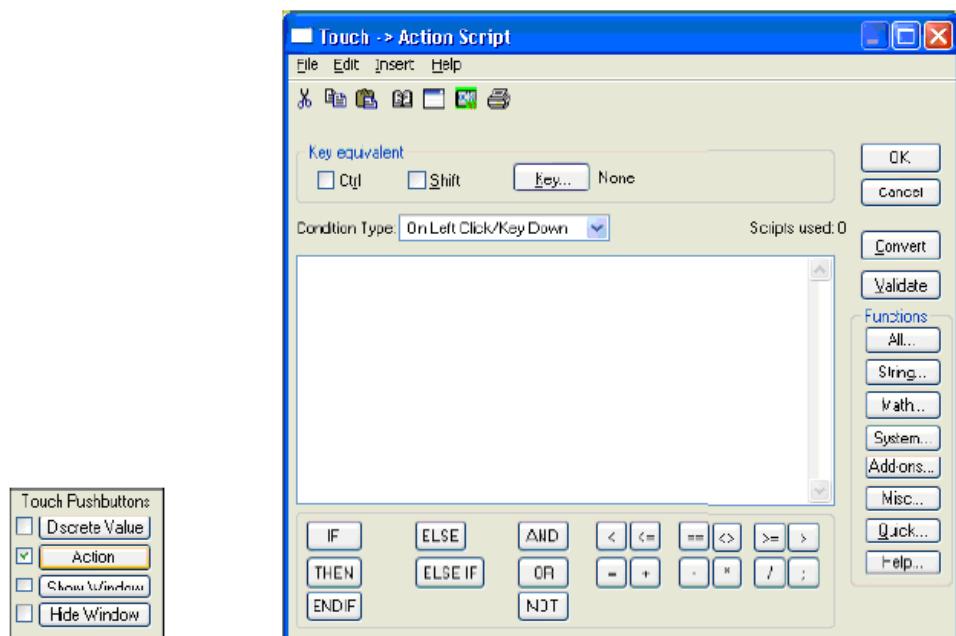
1이면 그것을 0으로 재설정합니다.

¥ Reset: 푸쉬버튼이 클릭될 때 값을 0(False,Off,No)으로 합니다.

¥ Set: 푸쉬버튼이 클릭될 때 값을 1(True,On,Yes)로 합니다.

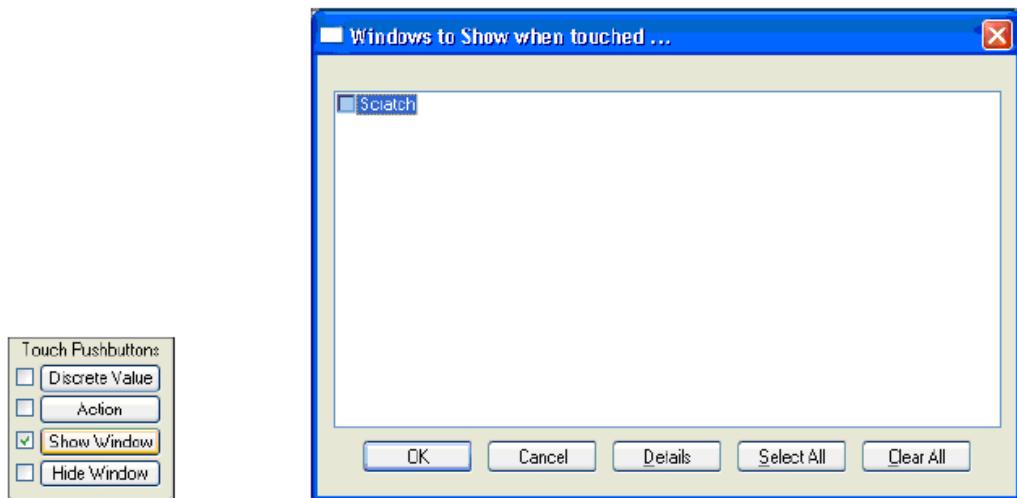
Touch Pushbuttons – Action

액션 스크립트들은 다음 챕터에서 다루겠습니다. 이것은 Touch Pushbutton – Action을 할당할때 나타나는 액션 스크립트 예제입니다.



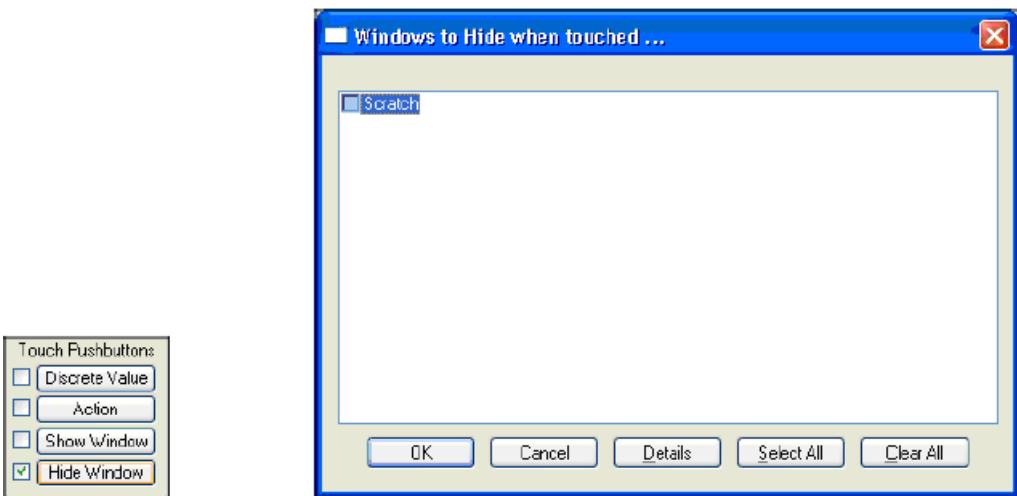
Touch Pushbuttons – Show Window

Touch Pushbutton 애플리케이션부분에서 Touch Pushbutton - Show Window 애니메이션을 할당하면 대화상자가 모든 윈도우 리스트를 보여줄 것입니다.



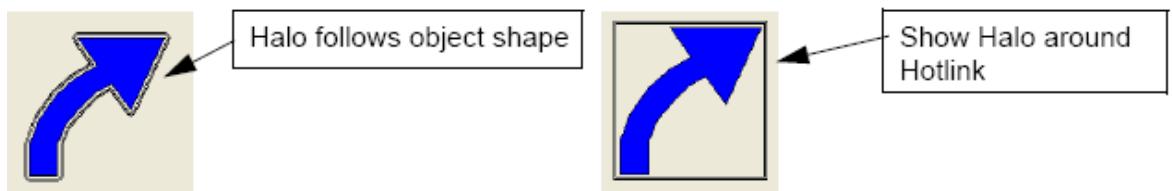
Touch Pushbuttons – Hide Window

Touch Pushbutton 애플리케이션부분에서 Touch Pushbutton - Hide Window 애니메이션을 할당하면 대화상자가 모든 윈도우 리스트를 보여줄 것입니다.



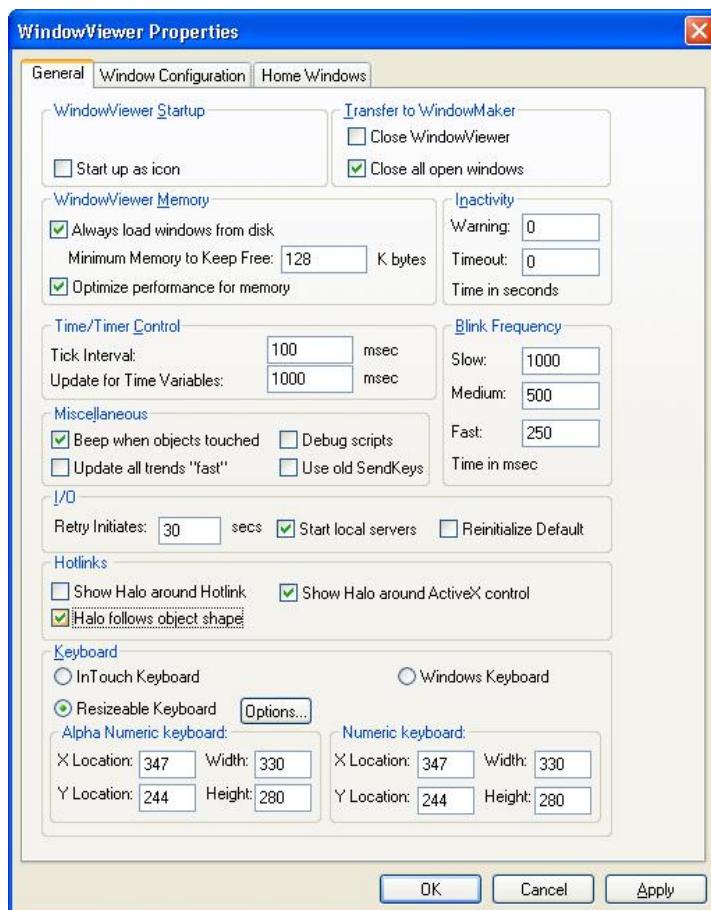
Appearance of Hotlinks in Runtime

Hotlink 동작은 런타임시 활성화 되거나 비활성화 되게 하고 ActiveX 컨트롤들을 보여 주거나 다음과 같은 Hotlink 형태의 객체를 보여주기 위해 음영주변에 Hotlink을 허용합니다.

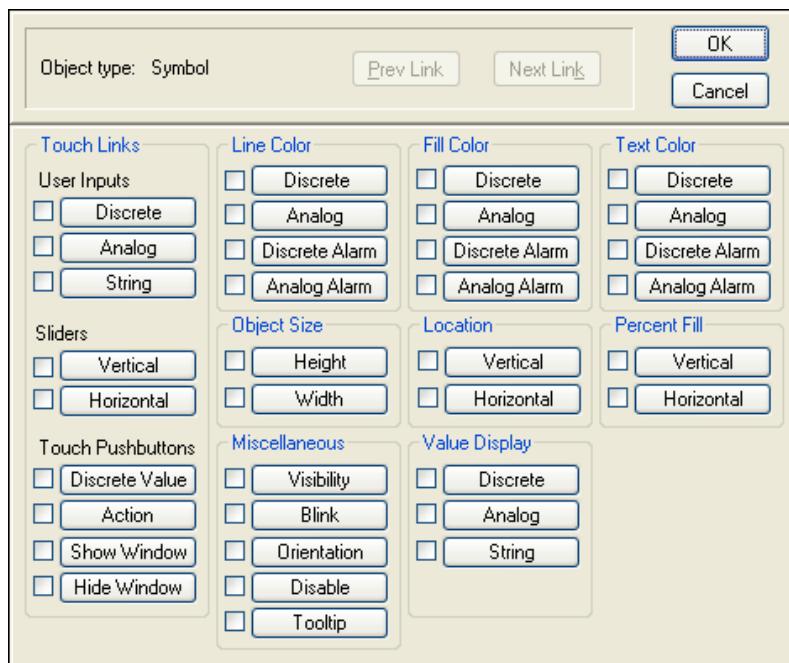


Note: 셀 또는 심볼 객체일지라도 음영은 객체의 형태를 따르지 않습니다.

hotlink링크 동작은 WindowViewer Properties 대화상자의 General 탭에서 설정합니다.



Display Links



이 연결은 사용자에게 화면 출력을 제공합니다. **display link**들의 8가지 유형이 위의 그림 안에 있습니다.

Line, Fill, and Text Color

Color 연결은 객체의 Line Color, Fill Color에 움직임을 주고 Text Color 속성에 움직임을 주기 위해 사용하는 연결입니다.

Note: Line이나 Text Color연결에는 반드시 단색을 사용 하셔야 합니다. 섞인 색을 선택하면 WindowMaker는 solid에 가까운 색을 선택할 것입니다. 섞인 색을 피하기 위해 비디오 카드는 최소 2MB어야 되며 색상모드는 32K나 65K(하이칼라나 트루칼라) 같은 256color이상이어야 합니다.

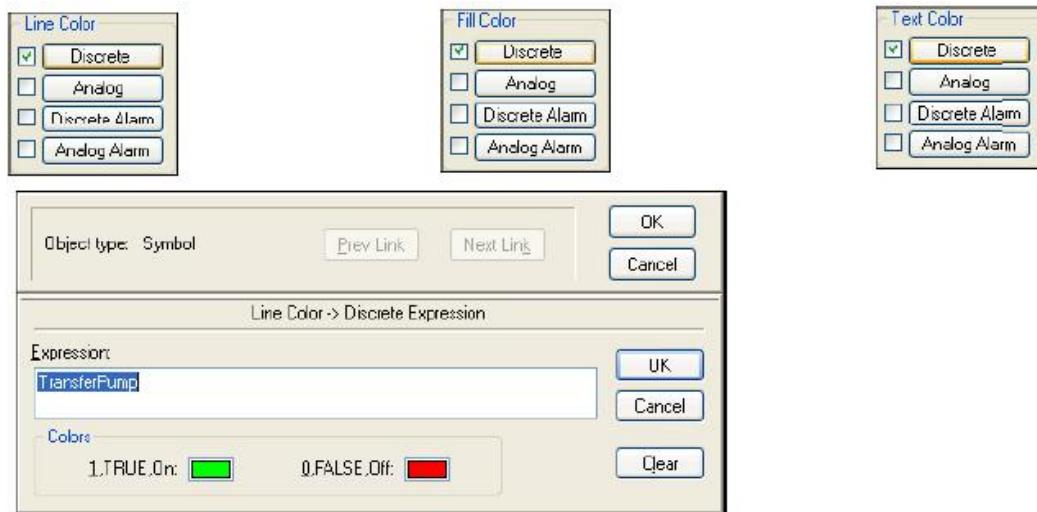
색에 관한 속성들을 이들의 Color 연결을 정의해 줌으로 태그값의 변화에 따라 동적으로 변화하게 만들 수 있습니다. 이 Color 속성은 이산 식, 아날로그 식, 이산 경보상태 또는 아날로그 경보 상태의 값들에 연결될 수도 있습니다.

선, 채우기, 텍스트 칼라연결에 4가지 형태가 있습니다.

사용자 입력	설명
Discrete	객체나 심볼의 Fill, Line 그리고 Text Color 속성들은 이산식의 값에 연결합니다.
Analog	객체나 심볼의 선 채우기, 그리고 Text Color는 아날로그 태그명의 값(정수 혹은 실수)이나 아날로그 식에 연결될 수 있습니다. 특별한 9개의 분기점에 의해 10개의 값 범위가 정의됩니다. 10개의 다른 색은 값 범위 변화처럼 보여지게 선택 할 수 있습니다.
Discrete Alarm	객체의 텍스트, 선, 객체의 색 채우기는 태그, 알람 그룹 또는 변수 그룹의 알람 상태에 연결 될 수 있습니다. 이 색 연결은 두 가지 색 선택을 사용할 수 있습니다. 하나는 보통 상태이고 하나는 태그의 알람 상태입니다. 이 연결은 알람과 이산태그 둘다 사용할 수 있습니다. 아날로그 태그를 같이 사용하면 이것은 태그의 어떠한 알람 조건에도 반응합니다.
Analog Alarm	객체의 문자, 선, 색채우기는 아날로그 태그 알람 그룹 또는 변수 그룹의 알람상태에 연결 될 수 있습니다. 보통 상태에 특별한 색 설정, 태그에 각 알람 조건을 위해 분리된 색을 허용합니다.

Discrete Line / Fill / Text Color

Note: 모든 선/채우기/텍스트 색상 연결은 같은 방법으로 설정되어집니다.



Expression: 이산 태그명이나 참 또는 거짓을 결정할 수 있는 식을 필드에 입력합니다.

표현식은 **1023**문자로 구성할 수 있습니다. 더 많은 표현식이 필요하면 QuickFunction을 만들어 그것을 표현식으로 호출 할 수 있습니다

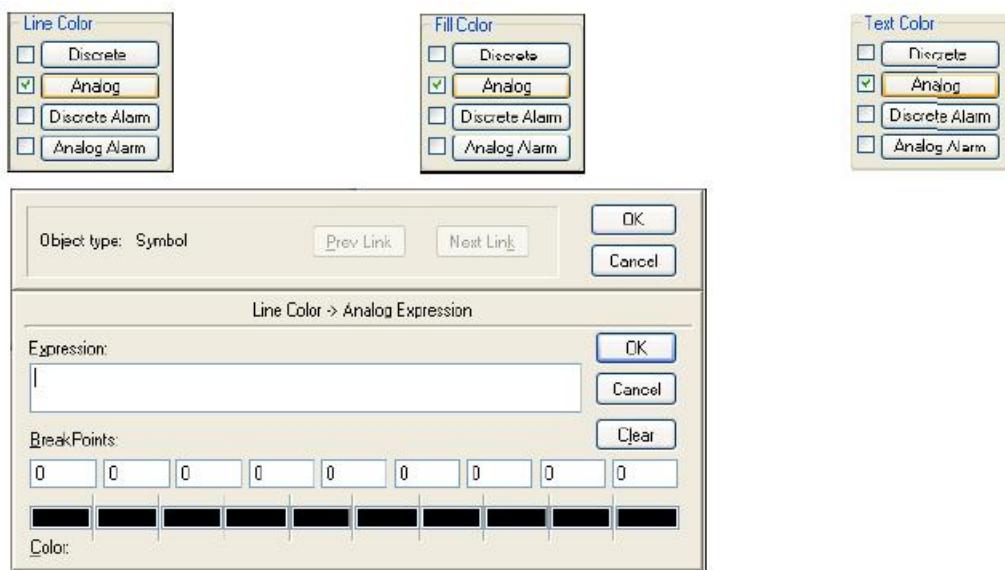
이산 표현들 또한 아날로그 태그명들을 포함할수 있다.

예를 들어, **TankLevel >= 75**. TankLevel태그의 값이 75보다 크거나 같을 때 객체의 색채우기는 변합니다.

선택된 문자를 적용하기 위한 명령을 보기 위해 표현 필드를 오른쪽 클릭합니다.

Colors area: 각 태그 상태에 대한 **Palette**로 부터 각 색상자를 클릭하고 색을 선택합니다.

Analog Line / Fill / Text Color



Expression: 필드에 아날로그 값을 계산해 넣어 주는 식이나 정수 또는 실수 태그명을 입력합니다.

Break Points: 각 필드에 색이 변할 분기점에 주어질 값들을 입력합니다.(실수도 사용할 수 있습니다.)

Color: 각 범위에서 사용될 색을 선택하기 위해 각 색 상자를 클릭합니다.

Discrete Alarm Line / Fill / Text Color



Tagname: 알람 상태와 관련된 객체 이산 태그명을 입력합니다.

Colors area: 각 상태에 사용할 색을 선택하기 위해 색 상자를 클릭합니다.

Analog Alarm Line / Fill / Text Color

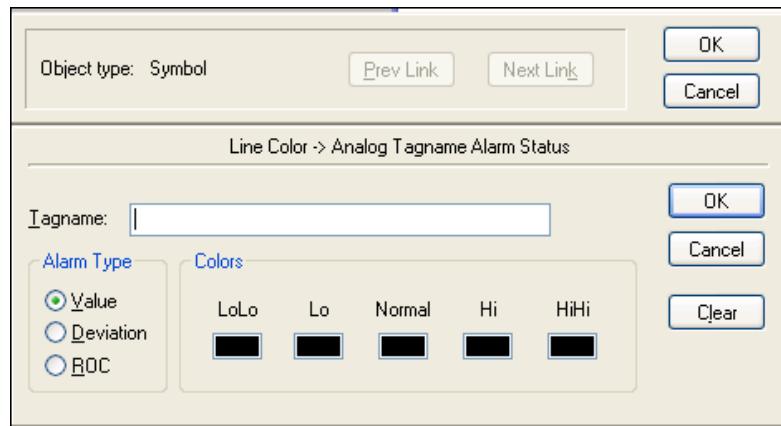


Tagname: 알람 상태와 관련된 객체 아날로그 태그명을 입력합니다

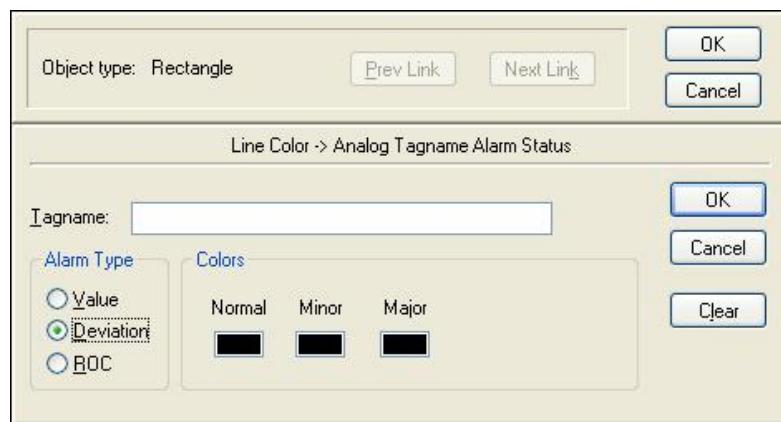
Alarm Type area: 객체와 관련된 알람 유형을 선택합니다.

3가지 상호 배타적인 아날로그 색 연결이 가능함

- I **Value Alarm :** 정의되어 있는 값 경보의 상태에 따라 다섯 가지의 다른 색을 선택할 수 있습니다.



- I Deviation : 정의되어 있는 편차 경보의 상태에 따라 세 가지의 다른 색을 선택할 수 있습니다.



ROC (Rate-of-Change) 정의되어 있는 변경율 경보의 상태에 따라 두 가지의 다른 색을 선택 할 수 있습니다.



Object Size Links

아날로그(정수 혹은 실수)태그명이나 아날로그 식의 값에 따라 객체의 높이 및 넓이를 변화시키기 위해 사용하는 연결입니다. 크기 연결은 연결에 “anchor”을 설정함으로 객체의 높이 및/혹은 넓이를 확대하는 방향을 조정할 수 있습니다. 높이 연결과 넓이 연결 둘 다 같은 객체에 동시에 붙일 수 있습니다.

Note: 높이와 넓이 연결은 같은 방법으로 만들어집니다.

Height or Width Link

Expression: 아날로그 태그명이나 아날로그 값을 계산해 주는 아날로그 식을 입력합니다.

Value at Max Height / Width: 객체가 최대 높이를 가질 때의 태그명이나 식의 값을 입력합니다.

Value at Min Height / Width: 객체가 최저 높이를 가질 때의 태그명이나 식의 값을 입력합니다.

Max % Height / Width: 태그명이나 식의 값이 **Max Height Value** 필드에 설정된 값을 가질 때 객체의 높이를 보여 주기 위해 퍼센트를 **0**부터 **100**사이의 값으로 입력합니다.

Min % Height / Width: 태그명이나 식의 값이 **Min Height Value** 필드에 설정된 값을 가질 때 객체의 높이를 보여 주기 위해 퍼센트를 **0**부터 **100**사이의 값으로 입력합니다.

Note: 높이 수치 퍼센트는 객체의 실제 그려진 크기 퍼센테이지처럼 표현됩니다.

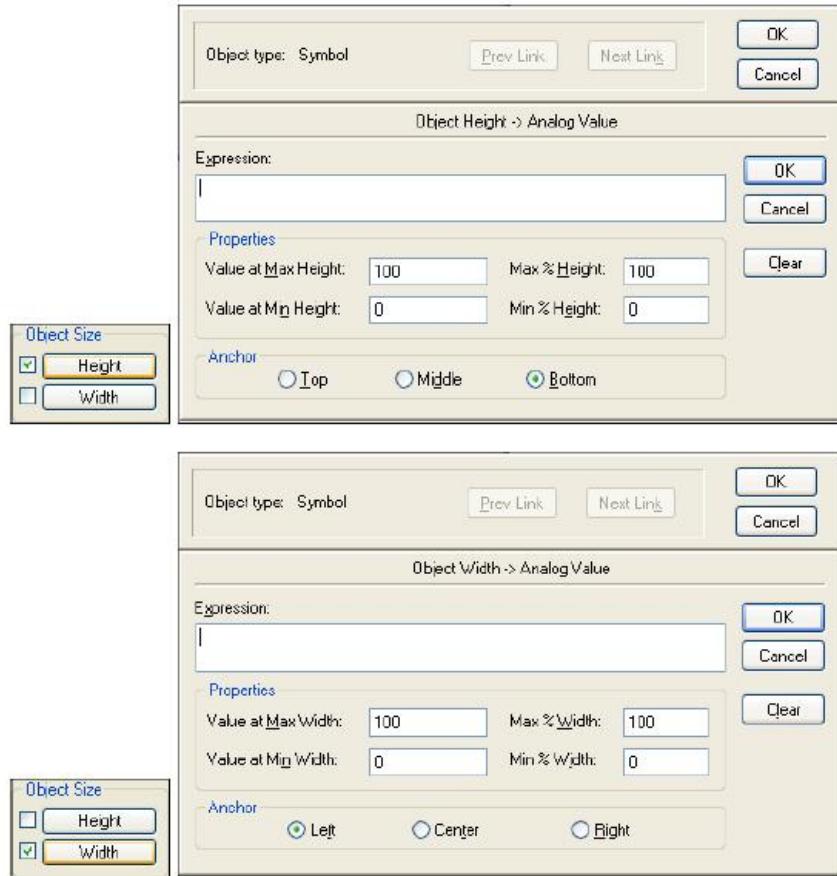
Anchor: 객체의 크기가 커지게 될 시작 점

Location

Location연결은 식의 값의 변화에 따라 객체를 자동적으로 수평, 수직 혹은 두가지 모두의 방향으로 움직이기 위해 사용합니다.

Note: Horizontal과 Vertical Location 연결은 같은 방법으로 만들어 집니다.

Vertical or Horizontal



Expression: 아날로그태그명(정수 또는 실수)이나 아날로그 값을 계산 해주는 아날로그식을 입력합니다.

At Top / Left End: 객체가 왼쪽끝이나 최상단에 있을때의 태그명의 값을 입력합니다.

At Bottom / Right End: 객체가 오른쪽끝이나 최하단에 있을때의 태그명의 값을 입력합니다.

Up / To Left: 객체가 현재위치에서 위나 왼쪽으로 움직일 수 있는 픽셀 수를 입력합니다.

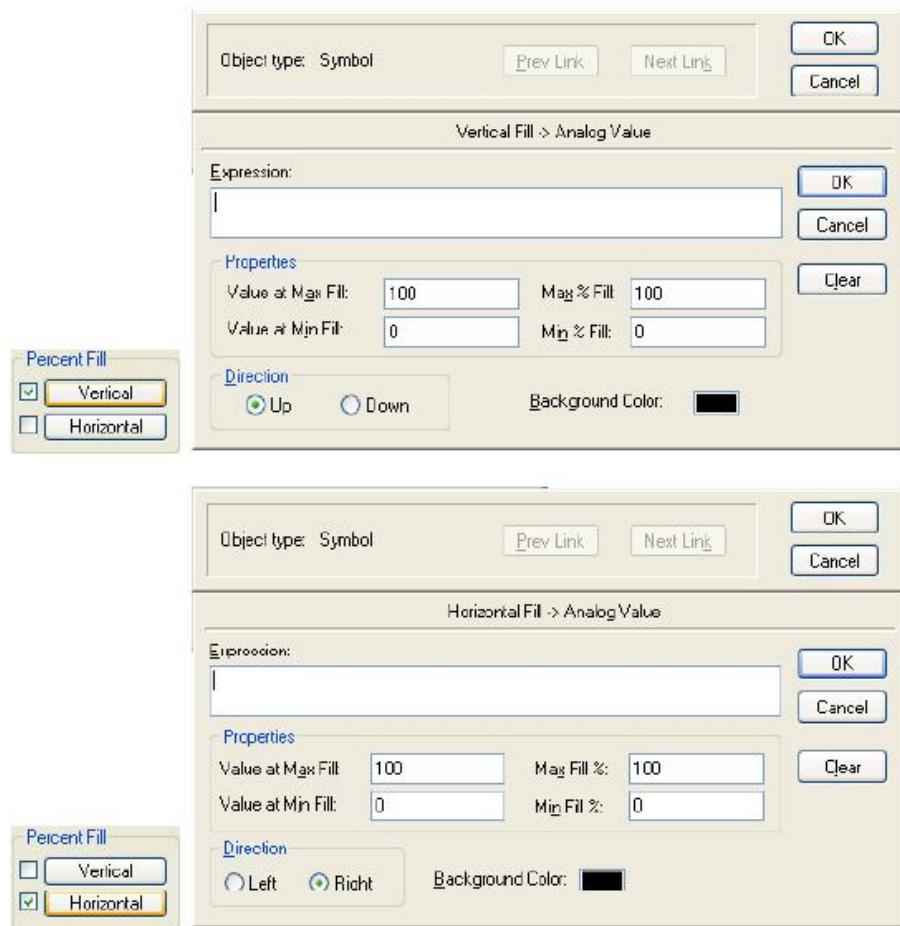
Down / To Right: 객체가 현재위치에서 아래나 오른쪽으로 움직일 수 있는 픽셀 수를 입력합니다.

Percent Fill Links

이 연결로 아날로그 태그명이나 아날로그 값을 계산해 주는 식의 값에 따라 칠해진 모양(또는 칠해진 모양을 포함하는 심볼)의 채워진 정도를 변하게 할 수 있습니다. 예를 들면 어떤 용기 안에 채워진 액체의 정도를 표시하게 할 수 있습니다. 객체나 심볼은 Horizontal Fill 연결, Vertical fill 연결 또는 둘 다 가질 수 있습니다.

Note: Vertical 과 Horizontal Percent Fill 연결은 같은 방법으로 만들어집니다.

Vertical or Horizontal



Expression: 아날로그 태그명이나 식을 입력합니다.

Value at Max Fill: 채우기 정도가 최대에 달했을 때의 식의 값을 입력합니다.

Value at Min Fill: 채우기 정도가 최소에 달했을 때의 식의 값을 입력합니다.

Max % Fill: 표현식이 설정된 값에 도달했을 때 객체가 실제로 채워지는 퍼센트를 가리키는 값(**0- 100**)을 입력합니다.

Min % Fill: 표현식이 설정된 값에 도달했을 때 객체가 실제로 채워지는 퍼센트를 가리키는 값(**0- 100**)을 입력합니다.

Direction: 하단에서 상단/상단에서 하단/왼쪽에서 오른쪽/오른쪽에서 왼쪽으로 부터 채워지고 움직이기 시작한 방향

Background Color: 객체의 채워지지 않은 부분의 색입니다. 실제 Fill Color는 객체가 만들어질 때 도구 상자에서 선택해진 색입니다. 만약 Vertical Percent Fill와 Horizontal Percent Fill 연결이 둘 다 객체에 연결되어 있으면 둘 중의 하나 연결 정의 대화 상자에서 마지막으로 선택해진 색이 배경색이 됩니다.

Miscellaneous Links

다섯가지 Miscellaneous 연결들이 있습니다.

MISC Link	설명
Visibility	식의 값이 참인지 거짓인지에 따라 객체의 가시성을 관리하는 연결입니다.
Blink	이산 태그명이나 식의 값에 따라 객체가 깜빡이게 해주는 연결입니다.
Orientation	아날로그의 값에 따라 객체의 방향을 자동적으로 회전시키는 연결입니다.
Disable	이산 태그명이나 식의 값에 따라 자동적으로 객체를 활성화 할 수 있는 기능을 사용 불가능하게 합니다.

Tooltip	런타임 상태에서 커서가 특정 객체위에 올라가 있을 때 툴팁을 나타내는 기능입니다. 툴팁내용은 문자열을 직접 입력하여 사용할 수도 있고 메모리 태그를 이용하여 툴팁 내용을 변경 시킬 수도 있습니다.

Visibility



Expression: 이산 태그명이나 식을 입력합니다.

Visible State

On: 식이 값이 참일때 객체가 나타납니다.

Off: 식의 값이 거짓일때 객체가 사라집니다.

Blink

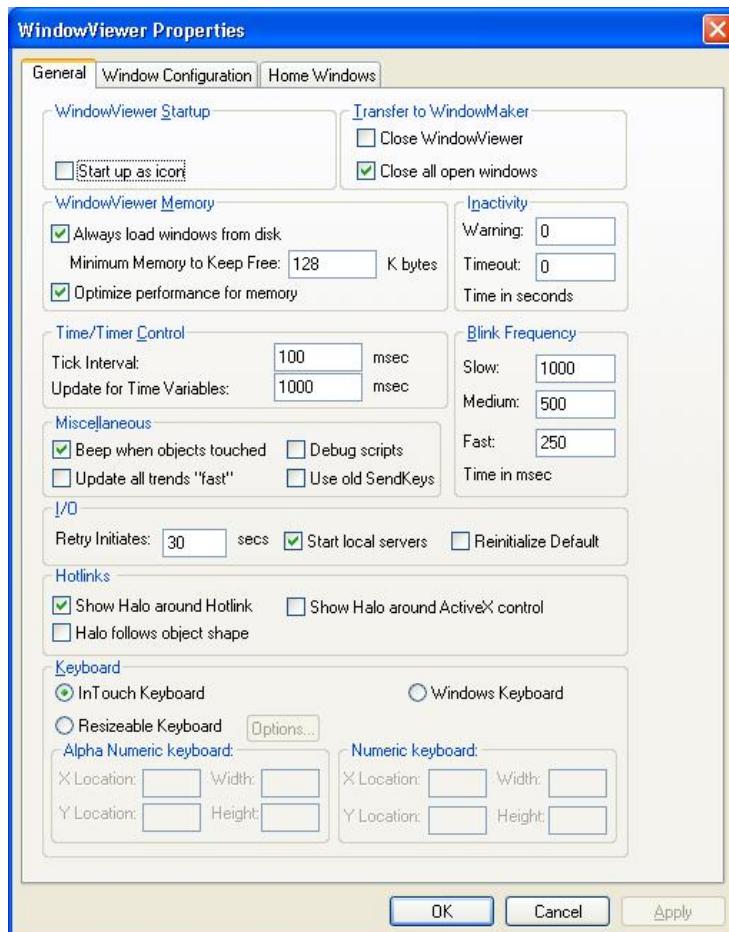


Expression - Blink When: 객체가 깜박일 때를 관리하는 이산 태그명이나 식을 입력합니다.

Blinked Attributes area:

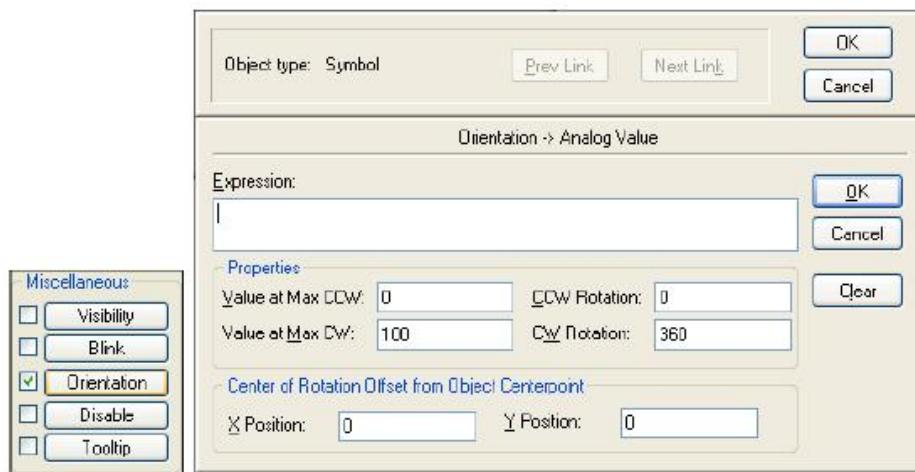
- | **Blink Invisible:** 객체/심볼이 윈도우에서 사라졌다 나타나는 것을 반복합니다.
- | **Blink visible with these attributes:** 라디오 버튼을 선택하면 객체나 심볼이 윈도우에 나타나 있는 채로 색의 속성을 변화시켜 깜박임 효과를 얻게 해줍니다.
- | **Text Color, Line Color and Fill Color:** 색을 선택합니다. 객체가 깜빡일 때의 채워지는 색을 선택합니다.

Blink Speed: 천천히, 중간, 빠르기 속도는 Special /Configure / WindowViewer 명령 선택에 의해 설정 할 수 있습니다. WindowViewer Properties 대화 상자 General 탭 필드는 Blink Frequency 그룹을 가지고 있습니다.



Note: 어떤 변경은 **global** 설정을 만들고 어플리케이션 전체에 걸쳐 모든 깜박임 연결의 깜박임 속도에 영향을 줍니다.

Orientation



Expression: 태그명 또는 식을 입력합니다.

Value at Max CCW: 객체가 최대 반 시계 방향으로 회전하게끔 하는 식의 값을 입력합니다.

Value at Max CW: 객체가 최대 시계 방향으로 회전하게끔 하는 식의 값을 입력합니다.

CCW Rotation: 최대 반 시계 방향을 위한 식의 입력된 값에 응하여 객체가 반 시계 방향으로 회전할 도수를 입력합니다.

CW Rotation: 최대 시계 방향을 위한 식의 입력된 값에 응하여 객체가 시계 방향으로 회전할 도수를 입력합니다.

객체는 그들이 WindowMaker에 그려진 원래 위치에 근거해서 시계방향으로 회전되어 지거나 반시계 방향으로 회전되어 집니다

Tip: 간단히 360도로 최대 반 시계방향으로 회전하게 값을 설정하거나, 0도로 최대 시계방향을 회전하게 값을 설정하거나, 360도로 반 시계방향으로 회전하거나, 0도로 시계 방향으로 회전하게 하기위해 객체에 특별한 각도를 줍니다. 그리고 표현식 상자안에 각도 값을 입력합니다.태그명 없이 이 표현식은 결코 변경되지 않으며 객체는 항상 90도위치에 고정되어 있을 것입니다.

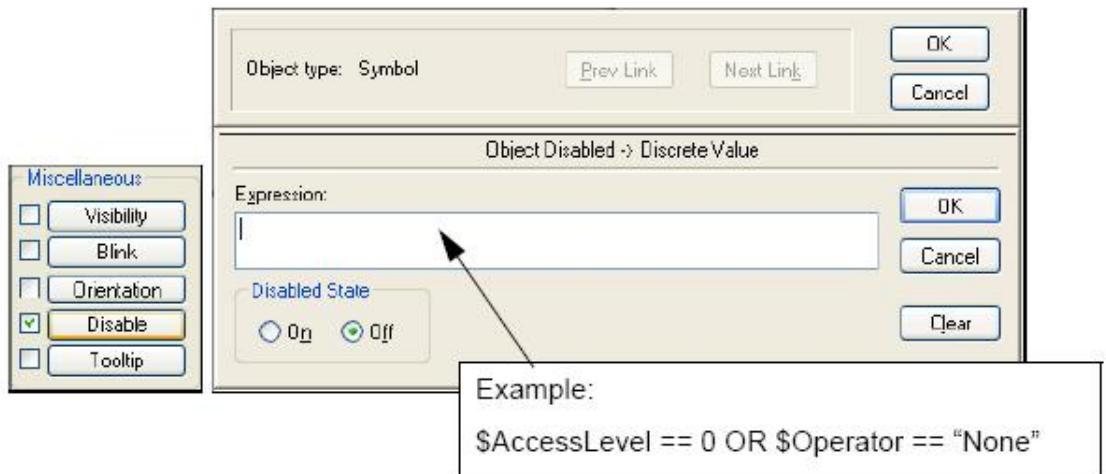
Note: 문자 객체 방위는 90도 증가분 안에서 WindowMaker에 설정될 수 있으나, WindowViewer에서 태그 값은 회전되지 않습니다.

X Position: 객체의 중심점에서 회전 중심점까지 사이에 수평으로 이동할 픽셀 수를 입력합니다. 양수값은 오른쪽으로 움직임을 가리킵니다. **orientation** 연결은 객체나 심볼의 중심점을 회전의 중심점으로 사용합니다.

Y Position: 객체의 중심점에서 회전 중심점까지 수직으로 이동할 픽셀 수를 입력합니다. 양수값은 가운데 아래로 움직임을 가리킵니다. 음수값은 가운데 위로 움직임을 가리킵니다.

Disable

어플리케이션에 보안을 적용 시킬 때 이 연결은 매우 유용합니다. 예를 들어 로그온한 오퍼레이터의 액세스 레벨이나 이름에 따라 어떤 객체의 사용을 불가능하게 할 수 있습니다. 이를 사용할 때, **Disable** 상태가 **True**일 동안 **Intouch**의 어떠한 연결도 허용되지 않을 것입니다.



Expression: 이산 태그명이나 식을 입력합니다.

위의 그림에서 보여준 표현식 사용에서, 어떤 로그온도 없다면, 객체나 버튼은 임의의 조작으로부터 보호됩니다.

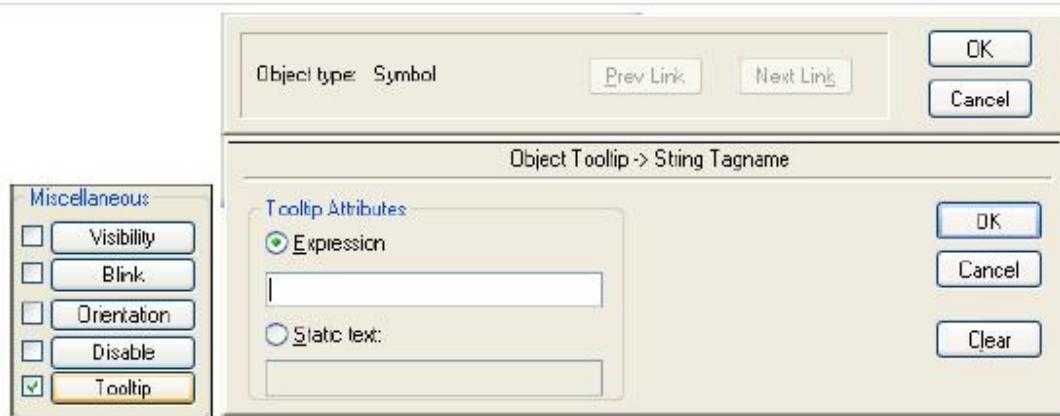
Disabled State: 이산태그나 표현식이 참일 때 객체를 활성화하게 하는 기능을 사용하게 할 것인지 불가능하게 할 것인지 선택합니다.

객체의 **Disabled State**가 **On**이라는 의미는 객체나 버튼의 **touch** 기능이 꺼지고 표현식이 참일 때까지 클릭을 할 수 없음을 의미합니다.

객체의 **Disabled State**가 **Off**이라는 의미는 객체나 버튼의 **touch** 기능이 꺼지고 표현식이

거짓일 때까지 클릭을 할 수 없음을 의미합니다.

Tooltip



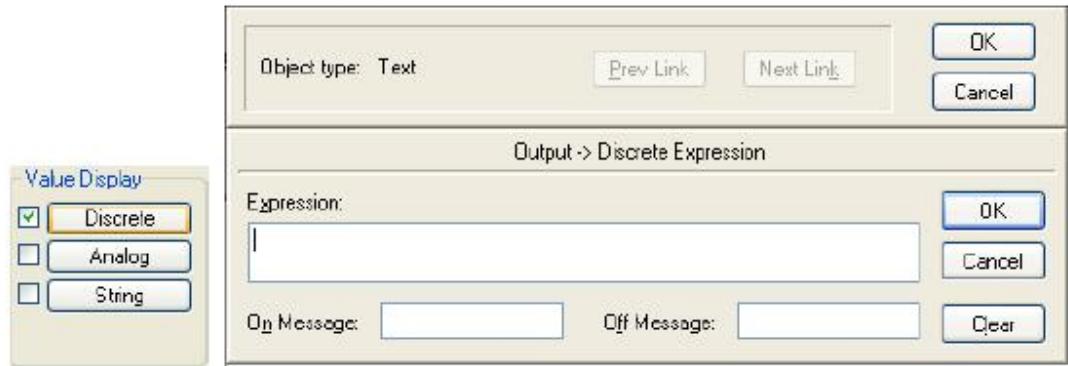
툴팁 애니메이션은 런타임 상태에서 커서가 특정 객체위에 올라가 있을 때 툴팁을 나타내는 기능입니다. 툴팁내용은 문자열을 직접 입력하여 사용할 수도 있고 메모리 태그를 이용하여 툴팁 내용을 변경 시킬 수도 있습니다.

Value Display Links

이 연결은 이산, 아날로그 또는 문자열 태그명의 값을 디스플레이 해주는 텍스트 객체를 사용할 수 있게 해줍니다.

Value Display Type	설명
Discrete	이 연결은 이산 식의 값을 따라 사용자가 정해 주는 메시지를 텍스트 필드에 표시해 주는 연결입니다.
Analog	이 연결은 아날로그 식의 값을 텍스트 객체에 나타내기 위해 사용하는 연결입니다.
String	이 연결은 텍스트 객체에 문자열 식의 값을 표시해 주기 위해 사용합니다.

Value Display – Discrete



Expression: 이산식을 입력합니다.

On Message: 이산식의 값이 **1**(True,On,Yes)일 때 디스플레이 될 메시지를 입력합니다.

Off Message: 이산식의 값이 **0**(False,Off,No)일 때 디스플레이 될 메시지를 입력합니다.

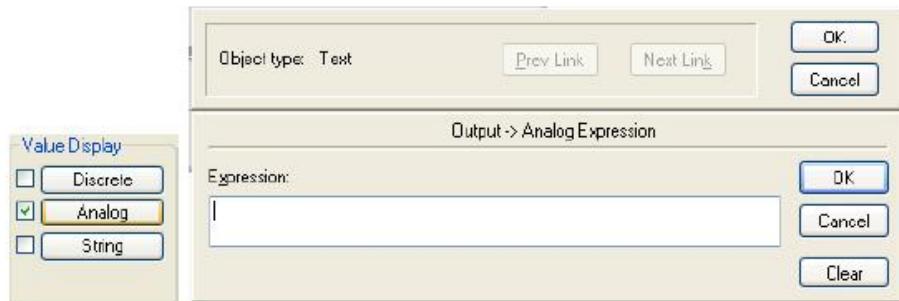
이 메시지는 원래 문자객체에 사용되는 폰트, 크기, 색, 정렬 위치와 객체에 연결된 속성 설정안에 보여 질것입니다. 원래 필드의 내용들은 **Runtime**시에 영향 없이 보여집니다.

또한 Value Display Output -> String Expression 연결을 이산태그에 켜짐과 꺼짐을 보여 주기위해 연결할수 있습니다. 연결을 위해 아래와 같은 식을 입력했을 것입니다.

`DText (Pump, Pump.OnMsg, Pump.OffMsg);`

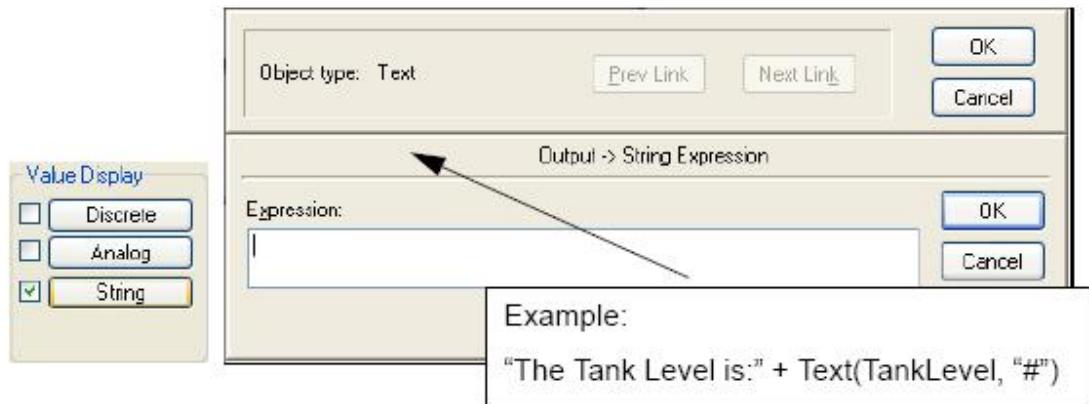
표현식에서 .OnMsg 와 .OffMsg 문자열은 이산 태그 Pump.에 InTouch **Tagname dictionary** 으로 부터 추출될 것입니다.

Analog



Expression: 태그명이나 식을 입력합니다. 또한 표현식에서 이산형 태그를 사용할 수 있습니다. 그것은 간단히 **1** 또는 **0**으로 보여 질 것입니다.

String



Expression: 태그명이나 식을 입력합니다.

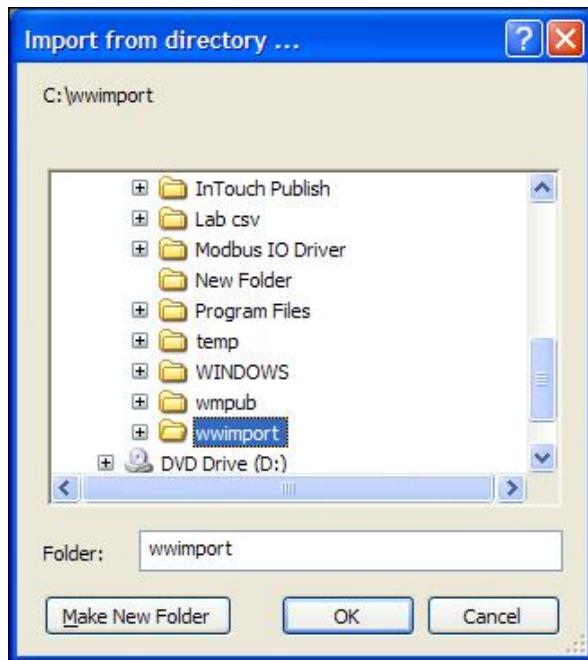
위에 예제에서, **Text()**스크립트는 **TankLevel** 정수태그의 값을 문자형으로 변환하기 위해 사용합니다.

윈도우 읽어오기 내보내기

어떤 InTouch 어플리케이션으로부터 **import** 한 윈도우들은 현재 어플리케이션으로 저장할 수 있습니다. 또한 **import** 한 윈도우들은 **remote tagname references** 생성에 빠르고 쉬운 방법을 제공합니다. 이것은 어떠한 이전에 생성된 윈도우들, 객체들, 윈도우 스크립트의 재사용을 가능하게 합니다. InTouch 어플리케이션으로부터 다른 곳으로 윈도우를 이동할 때 반드시 **File / Import** 명령을 사용해야 합니다.

Note: 어떤 다른 방법으로 InTouch 윈도우 파일 복사를 시도하려 한다면, 어플리케이션 **Tagname dictionary**에 오류가 생길지도 모릅니다.

- a. 어떠한 윈도우들이라도 닫습니다.
- b. WindowMaker 메뉴에 **File / Import**을 클릭합니다. **Import from directory** 대화상자가 나타납니다.



- c. Ok를 클릭합니다.

Note: published application ArchestrA graphics 사용으로 부터 **import** 라면,
그래픽들은 보여지지 않을 것이며, System Management console 로그 뷰어에서 에러
메세지가 보여질 것입니다.

Application Data Import Options 대화상자가 나타납니다.

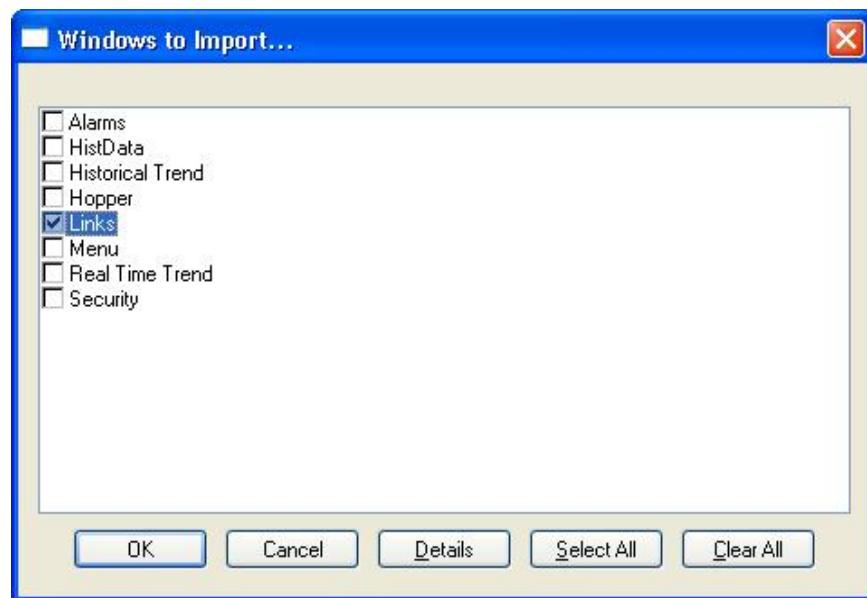
- d. Import Options 부분으로 부터 Windows를 선택합니다. 각 **importing** 옵션은 상세한
import 리스트들을 가지고 있습니다.



e. 원도우즈 옵션에서 Select버튼을 클릭합니다.

Windows to Import...윈도우가 나타납니다.

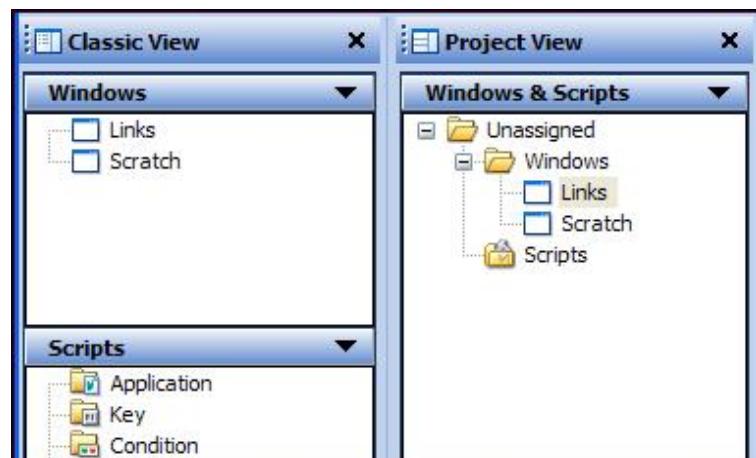
f. Links 윈도우를 선택합니다.(Links윈도우는 랩에서 사용했습니다.)



g. Ok를 클릭합니다. 어플리케이션 import 대화상자가 다시 나타납니다.

h. Import를 클릭합니다.

연결 윈도우는 성공적으로 InTouch 어플리케이션에서 가져오게 됩니다.



윈도우 내보내기

윈도우 **export**는 라이브러리 어플리케이션을 만들어 유지하거나 한 윈도우를 비슷한 윈도우를 사용하는 다른 어플리케이션으로 이동할 때 매우 유용합니다. 윈도우를 이동시켰을 때 윈도우 파일은 선택된 디렉토리에 소속된 인덱스 파일과 같이 복사 됩니다. 게다가 이 디렉토리는 다른 어플리케이션의 **import** 윈도우와 같이 소스처럼 사용 될 수 있습니다.

Note: 어떤 다른방법으로 Intouch 윈도우 파일 복사를 시도하려 한다면, 어플리케이션 태그명 사전에 오류가 생길지 모릅니다.

현재 어플리케이션의 모든 윈도우들을 닫습니다.

- File / Export Window를 클릭합니다. Export to directory... 대화상자가 나타납니다.



- export**를 원하는 윈도우가 있는 어플리케이션 디렉토리를 선택합니다.

Windows to Export 대화상자가 보여지고, 어플리케이션에서 윈도우 선택이 가능합니다.

- Export**를 위해 윈도우를 선택하고 Ok버튼을 클릭합니다.

태그명 변환

태그 변환은 일반적으로 Intouch **importing** 작동의 한부분이기 때문에 아래의 정보들은 포함됩니다.

Import되는 윈도우는 단지 그래픽들의 구성들을 가져오고 태그들을 포함하지 않으며 그런 변환이 필요하지 않는다.

Note: 태그 변환은 **Lab 6, “Using InTouch QuickScripts”**에서 설명합니다.

어플리케이션의 구성을 **import** 혹은 **export**할 때 그 윈도우와 관련된 모든 태그명은 윈도우와 함께 전송되지만 새로운 어플리케이션의 데이터베이스에 추가되지는 않습니다. 대신에 그것들은 자동적으로 플레이스홀더 태그명으로 표시됩니다.

Note: 원도우가 **import** 되었을 때 태그명 사전에 태그가 정의되어 있지 않으면, 그들이 변환할 수 있기 전에 신속히 그들이 정의될 것입니다. 이 경우는 **Ok**를 클릭한 경우입니다. 태그명 사전 대화상자는 나타나 각 태그명을 정의할 수 있습니다.

플레이스홀더 **?d:, ?i:, ?r:, ?m:, ?g:, ?h:,** 그리고 **?t:** 등의 태그명들은 태그명이 원래 정의된 타입을 나타냅니다.

D Discrete type

I Integer type

R Real type

M Message type

G Group type

H Hist Trend type

T Tag ID type

Substituting Tagnames

객체가 복제 되었을 때 링크나 애니메이션, 스크립트와 같은 설정은 원래의 것과 똑같이 복제됩니다. 그러나 복제된 객체에 다른 태그의 사용이 필요하다면, 태그명을 반드시 변경해야 합니다.

WindowMaker에서 이러한 기능은 **substituting a tagname**이라고 불려집니다. 객체에 설정되는 태그명은 단독으로나 그룹으로나 언제든지 변경 될 수 있습니다.

- a. 변경하기를 원하는 태그명 객체를 선택합니다.
- b. **Ctrl+E (Special / Substitute Tags...)**를 누릅니다.
- c. **New Name box**에서 새로운 태그명을 입력하고 **Ok**를 클릭합니다. 선택된 객체와 관련된 태그명이 자동적으로 변경될 것입니다.

Substitute Tagnames... 1 of 3

Current Name:	Required Type	New Name:
DiscTag1	Discrete	DiscTag1
DiscTag2	Discrete	DiscTag2
RealTag	Analog	RealTag

Section 3 - Animation Links - Quick Reference

아래의 표는 모든 애니메이션 링크의 기본적인 기능을 정리 합니다.

Touch Links

User Input	설명
Discrete	이산 태그명의 값을 조정하기 위해 사용합니다. 연결이 활성화 되면 대화 상자가 나타나 값을 선택 할 수 있습니다.
Analog	아날로그(정수 또는 실수)태그명의 값을 입력하기 위해 사용하는 객체를 만들기 위해 사용하는 연결입니다. 이 연결이 활성화 되면 입력 상자가 나타나고 값을 입력할 수 있습니다. 옵션에 따라 스크린상에 키보드가 나타나게 할 수도 있습니다.
String	이 연결은 문자열 메시지를 입력할수 있는 객체를 만들기 위해 사용합니다. 연결이 활성화 되면 메시지 값을 입력할수 있게 입력 상자가 나타납니다.

Slider Touch Link	설명
Vertical / Horizontal	이 연결은 화면에서 마우스로 움직일수 있는 객체나 심볼을 만들기 위해 사용합니다. 객체나 심볼을 움직이는 동안 거기에 연결된 태그의 값이 변화합니다. 시스템의 설정값에 대한 디바이스를 생성 할 수 있습니다.

Touch Pushbuttons	설명
Discrete Value	이 연결은 객체나 심볼을 이산 태그의 상태를 제어하는 버튼으로 만들기 위해 사용합니다. 이산 값 수정, 액션 스크립트 적용, 그리고 윈도우 명령들을 숨기기와 보여주기 등이 있습니다.
Action	이 연결은 객체, 심볼 또는 버튼에게 스크립트 제어를 입력 할 수 있는 애니메이션입니다. (Module 5, “InTouch QuickScripts,” page 4-11.) 스크립트는 태그를 특정한 값으로 설정하기 윈도우를 보여 주거나 숨기기 그리고 다른 어플리케이션을 시작하고 제어하기, 함수 실행하기 등에 사용합니다.
Show Window	이 연결은 객체나 심볼을 클릭함으로 하나 이상의 윈도우가 나타나게 하는 pushbutton을 만들기 위해 사용합니다
Hide Window	이 연결은 객체나 심볼을 클릭함으로 하나 이상의 윈도우가 사라지게 하는 pushbutton을 만들기 위해 사용합니다.

Display Links

Color Link	설명
Discrete	객체나 심볼의 채우기, 선 그리고 문자 색 속성들은 이산식의 값에 연결합니다.
Analog	객체나 심볼의 선, 채우기, 그리고 문자 색은 아날로그 태그명의 값(정수 혹은 실수)이나 아날로그 식에 연결될 수 있습니다. 특별한 9개의 분기점에 의해 10개의 값 범위가 정의됩니다. 10개의 다른 색은

	값 범위 변화처럼 보여지게 선택 할 수 있습니다.
Discrete Alarm	<p>객체의 문자,선, 색 채우기는 태그, 알람그룹 또는 변수 그룹의 알람 상태에 연결 될 수 있습니다.</p> <p>이 색 연결은 두 가지 색 선택을 사용할 수 있습니다. 하나는 보통 상태이고 하나는 태그의 알람 상태입니다.</p> <p>이 연결은 알람과 이산태그 둘다 사용할 수 있습니다. 아날로그 태그를 같이 사용하면 이것은 태그의 어떠한 알람 조건에도 반응합니다.</p>
Analog Alarm	<p>객체의 문자,선, 색 채우기는 아날로그 태그 알람 그룹 또는 변수 그룹의 알람상태에 연결 될 수 있습니다. 이것은 보통 상태에 특별한 색 설정, 태그에 각 알람 조건을 분리된 색을 허용합니다.</p>

Object Size	설명
Analog(Integer or Real) Tagname, Analog expression	<p>아날로그(정수 혹은 실수)태그명이나 아날로그 식의 값에 따라 객체의 높이 및/혹은 넓이를 변화시키기 위해 사용하는 연결입니다. 크기 연결은 연결에 “anchor” 을 설정함으로 객체의 높이 및/혹은 넓이를 확대하는 방향을 조정할 수 있습니다. 높이 연결과 넓이 연결 둘 다 같은 객체에 동시에 붙일 수 있습니다. 높이와 넓이 연결은 같은 방법으로 만들어집니다.</p>

Location Links	설명
Analog	객체를 자동적으로 아날로그 태그명의 값이나 식에 반응해서 수평, 수직 혹은 두가지 모두의 방향으로 만들기 위해 사용됩니다.

Percent Fill	설명
Horiz/Vert Fill	이 연결로 아날로그 태그명이나 아날로그 값을 계산해 주는 식의 값에 따라 칠해진 모양(또는 칠해진 모양을 포함하는 심볼)의 채워진 정도를 변하게 할 수 있습니다. 예를 들면 어떤 용기 안에 채워진 액체의 정도를 표시하게 할 수 있습니다. 개체나 심볼은 Horizontal Fill 연결, Vertical fill 연결 또는 둘 다 가질 수 있습니다.
MISC Link	설명
Visibility	식의 값이 참인지 거짓인지에 따라 객체의 가시성을 관리하는 연결입니다.
Blink	이산 태그명이나 식의 값에 따라 객체가 깜박이게 해주는 연결입니다.
Orientation	아날로그의 값에 따라 객체의 방향을 자동적으로 회전시키는 연결입니다.
Disable	이산 태그명이나 식의 값에 따라 자동적으로 객체를 활동하게 할 수 있는 기능을 사용 불가능하게 합니다.
Tooltip	Runtime 상태에서 커서가 특정 객체위에 올라가 있을 때 툴팁을 나타내는 기능입니다. 툴팁 내용은 문자열을 직접 입력하여 사용할 수도 있고 메모리 태그를 이용하여 툴팁 내용을 변경 시킬 수도 있습니다.

Value Display	설명
Discrete	문자 객체에 사용자 정의 메세지를 On 또는 Off을 보여주기 위해 태그값이나 이산식을 사용합니다.
Analog	문자 객체에 아날로그 식이나 태그값을 보여줍니다.
String	문자 객체에 문자열 식이나 태그값을 보여줍니다.

Lab 3 - Configuring Animation Links

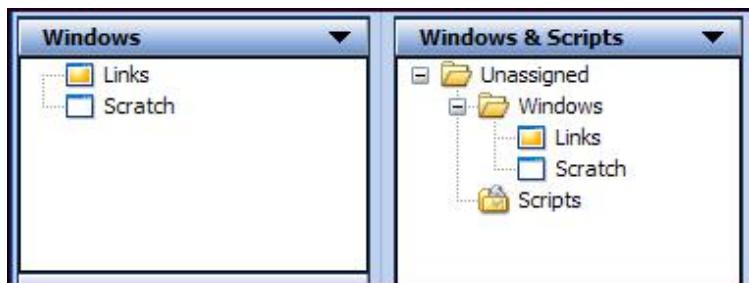
소개

이 랩은 어떤 이벤트의 상태에 근거해 발생하는 특별한 설정 행동과 애니메이션 링크들을 만들기 위해 Animation Selection 대화상자를 사용할 것입니다.

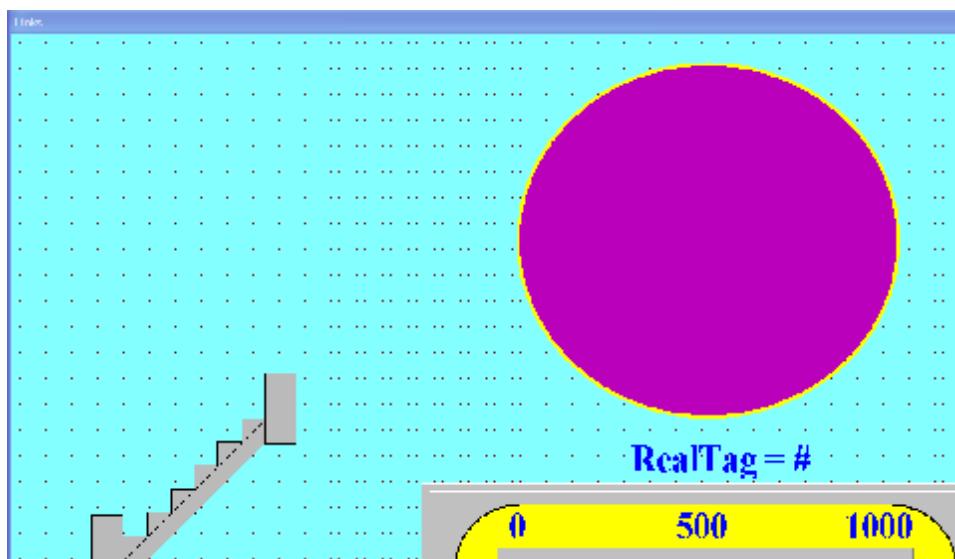
Detailed Lab

Links 윈도우를 열기

1. WindowMaker의 Application Explorer 트리구조에서 이용 가능한 윈도우들의 목록을 펼치기 위해 +를 클릭합니다.



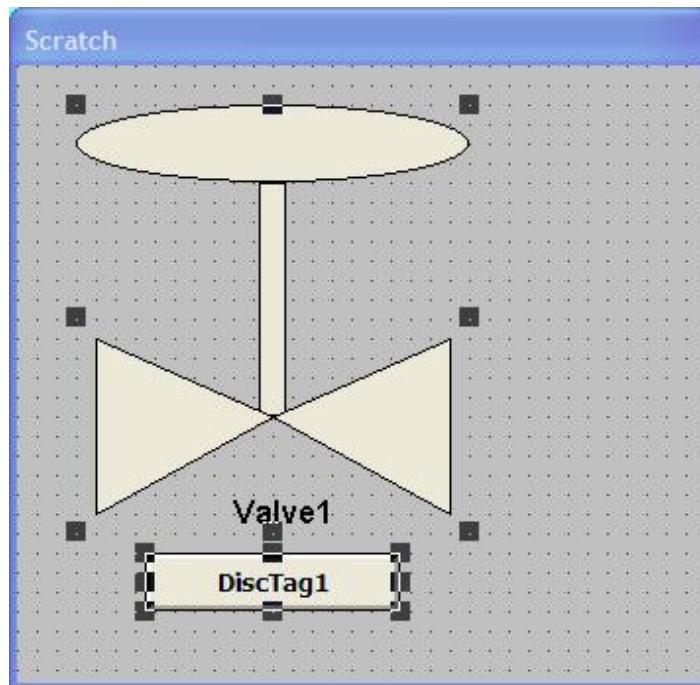
2. 새롭게 **import**한 윈도우를 열기 위해 **Links**를 더블 클릭합니다.
Links 윈도우는 아래와 그림과 같이 비슷하게 보여질 것입니다.



Note: 이 랩의 완료를 위해 어떠한 셀들이나 심볼들도 조각내지 않습니다.

그래픽들과 윈도우들 사이의 공유

3. Scratch 윈도우를 엽니다.

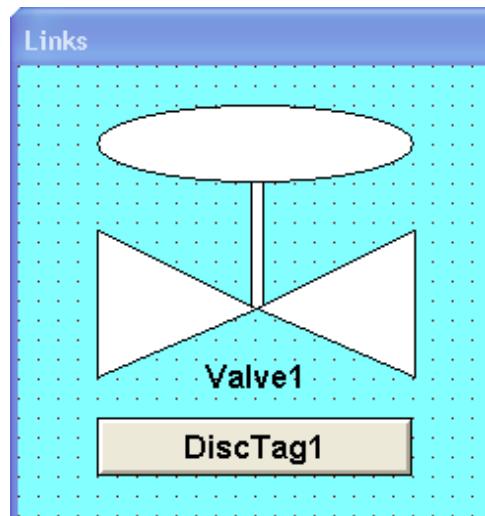


4. Valve1심볼과 그것의 버튼을 선택하고 둘다 복사합니다.

Note: 필요한 경우 Valve 객체들과 문자 객체를 심볼로 변환합니다. 이것은 복사/붙이기 동작 전이나 후에 할 수 있습니다

5. Scratch 윈도우를 닫습니다.
6. Links 윈도우안에 Valve와 그것의 버튼을 붙입니다. 포인터는 이것처럼 보일 것입니다.

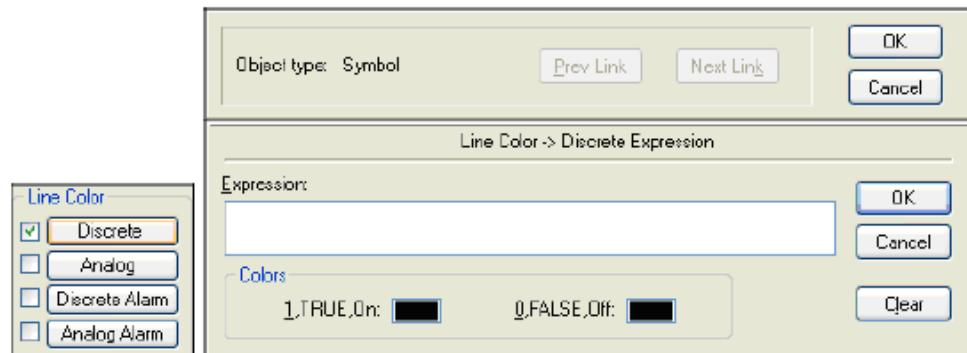
7. valve를 놓기 위해 Links 윈도우의 상단 왼쪽 부분을 한번 클릭합니다.



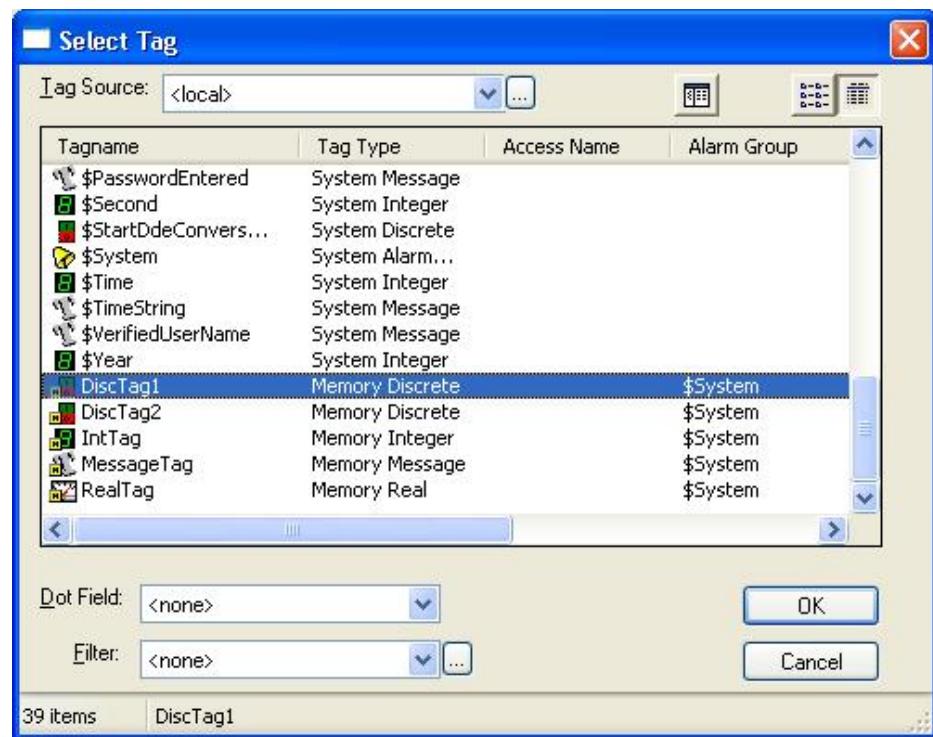
8. 위치와 크기를 적용합니다.

Valve1 심볼에 움직임 주기

9. Valve1심볼을 더블클릭 합니다.
10. Line Color : Discrete 버튼을 클릭합니다.
객체 유형: 심볼 대화상자가 나타납니다.



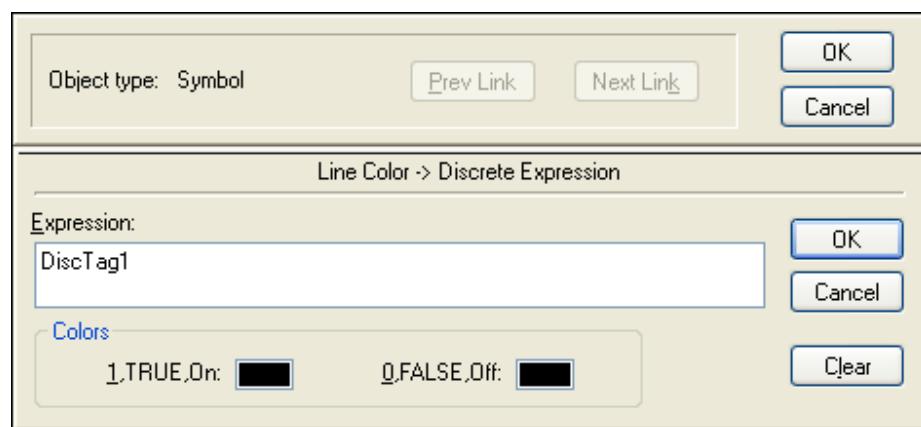
11. 표현식 필드를 더블 클릭합니다.
Tag browser 대화상자가 나타납니다.



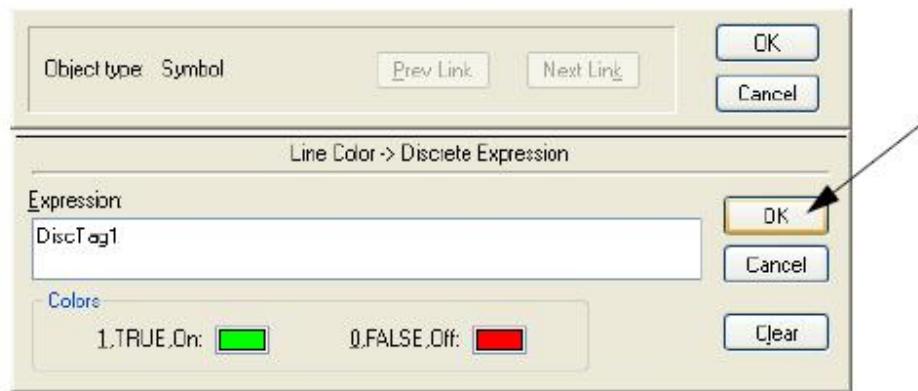
12. DiscTag1 태그를 선택합니다.

13. OK를 클릭합니다.

표현식 필드안에 태그명이 나타납니다.



14. 1, TRUE, On: 색상자를 클릭하고 color palette로 녹색으로 설정 합니다.



15. 0, FALSE, Off: 색상자를 클릭하고 color palette로 빨간색을 설정합니다.

16. 하단의 OK버튼을 클릭합니다.



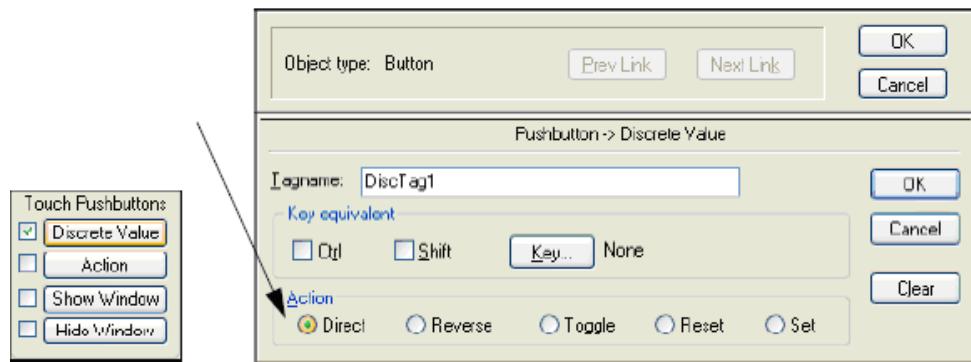
17. 심볼의 Fill Color와 Text Color를 이전 단계에서 사용한 DiscTag1 태그에 연결합니다..

18. 각 애니메이션 링크가 종료될 때 OK를 클릭하고 3개 모두 할당될 때 상단의 OK를 클릭합니다.

DiscTag1 버튼 움직임 주기

19. DiscTag1버튼을 더블 클릭합니다.

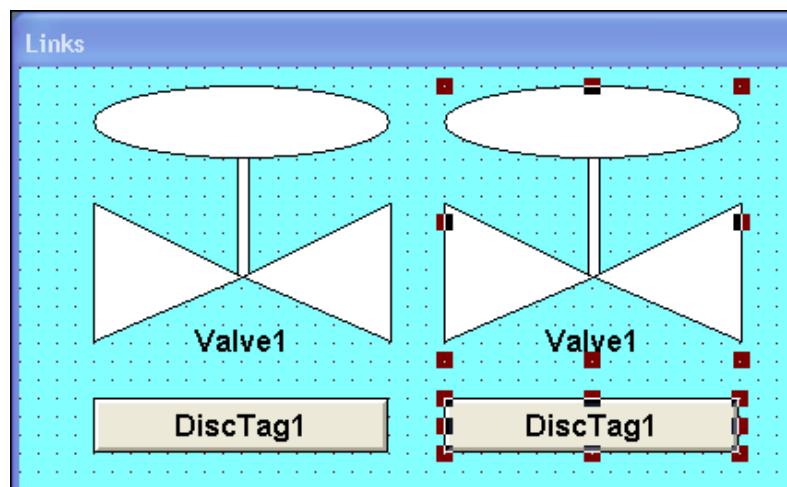
20. Touch Link / Touch Pushbutton / Discrete Value의 Direct action을 DiscTag1버튼에 정의합니다.



21. DiscTag1에 할당합니다.
22. OK버튼을 클릭합니다.

밸브와 버튼에 움직임 주기

23. Valve1 심볼과 DiscTag1버튼을 선택합니다.
24. 복사 버튼 클릭에 의해 valve 심볼과 버튼을 복사합니다.



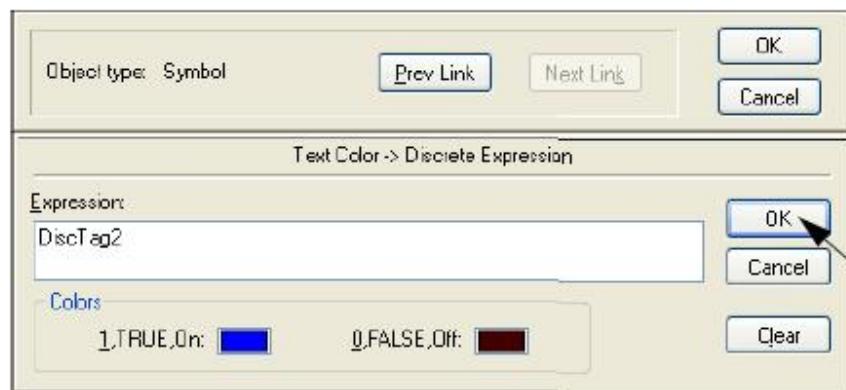
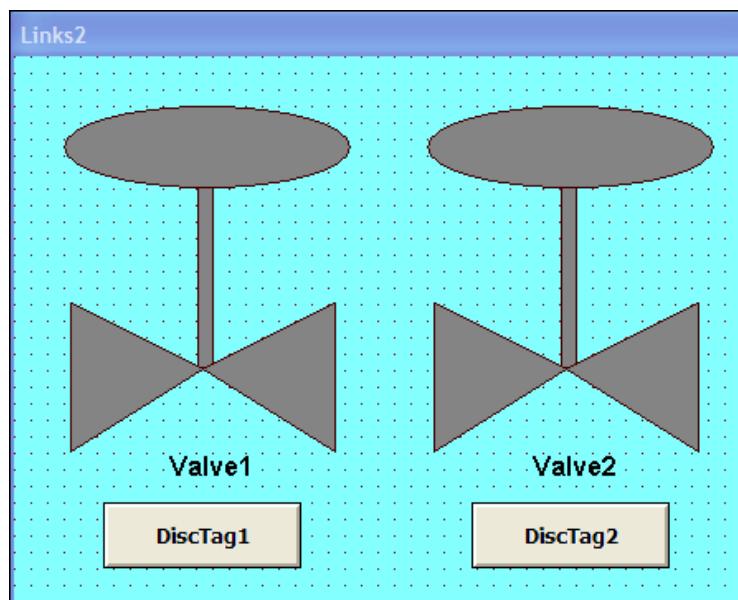
25. Valves 옆에 나란히 놓습니다.
26. 두번째 valves와 그것의 버튼을 선택하고 Special / Substitute Strings (CTRL+L)을 선택합니다.



27. Valve2의 문자와 DiscTag2의 버튼 라벨을 변경합니다.

28. OK를 클릭합니다.

밸브와 버튼은 아래와 같은 그림으로 나타납니다.



29. Valve2와 Line Color, Fill Color와 DiscTag2에 Text Color 을 더블 클릭합니다.

1,TRUE, On 과 0,FALSE,Off 를 사용하기 위해 푸른색과 검정색을 사용합니다.

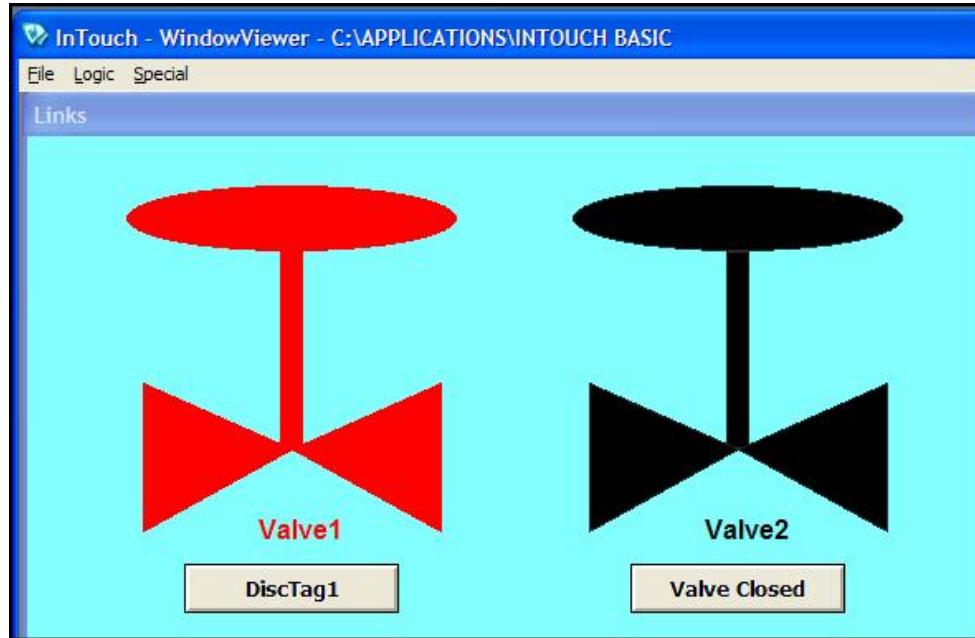
Note: 새로운 객체들안에 Valve1 심볼과 DiscTag1 은 모든 애니메이션 링크들을 유지하고 있기 때문에 Tag Editor 대화 상자안에의 하단 OK를 클릭하면 Animation Links Selector 대화 상자로 돌아 오고, 다른 태그 애니메이션 링크들도 쉽게 선택 할 수 있습니다. 애니메이션 링크들이 완료 되었을 때 대화 상자를 닫고 링크들을 저장하기 위해 대화 상장의 상단 부분에 있는 OK를 클릭합니다.



30. 프롬프트와 메세지를 위와 같이 설정하고 OK를 클릭합니다.



31. 화면 오른쪽 상단코너의 Runtime를 한번 클릭하여 Runtime으로 전환합니다.



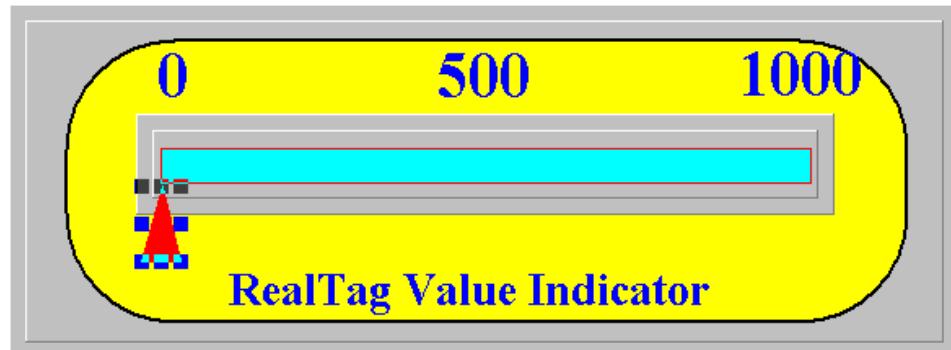
32. DiscTag pushbutton들을 클릭합니다. 그들의 On/Off 상태에 따라 밸브들은 변경되었을 것입니다.



33. Development를 한번 클릭하여 개발모드로 돌아갑니다.

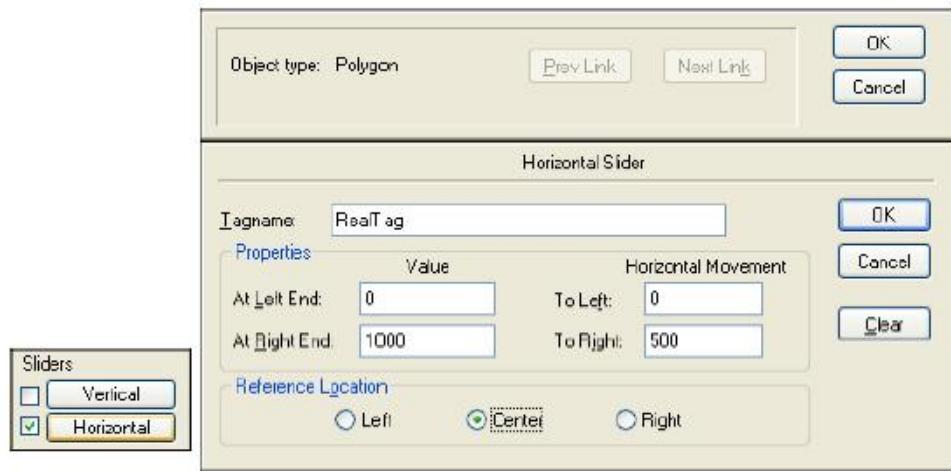
슬라이드에 움직임 주기

34. 포인터 객체를 더블 클릭합니다.



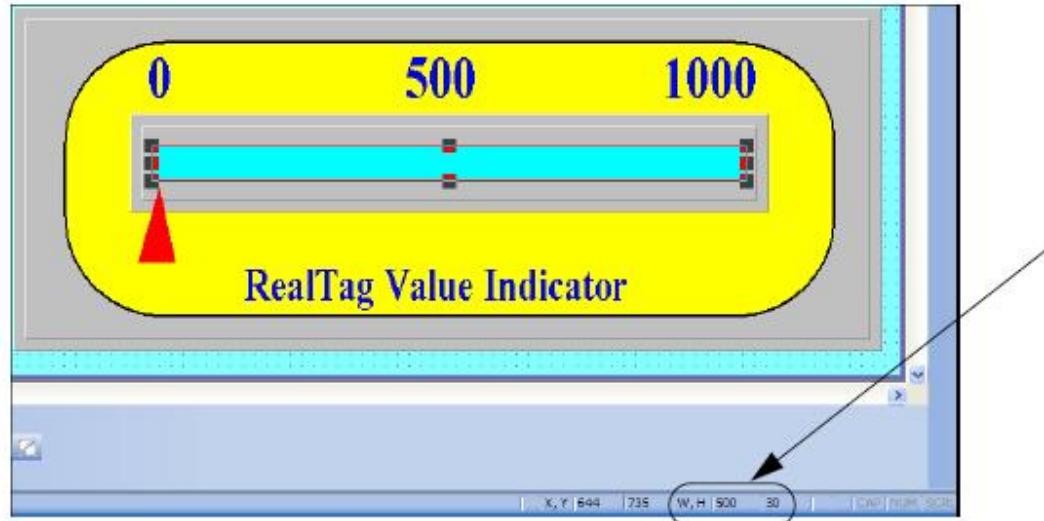
35. Animation Selection 대화상자의 Sliders 지역의 Horizontal버튼을 클릭합니다.

36. 아래 그림과 같이 슬라이드의 속성들을 설정합니다.



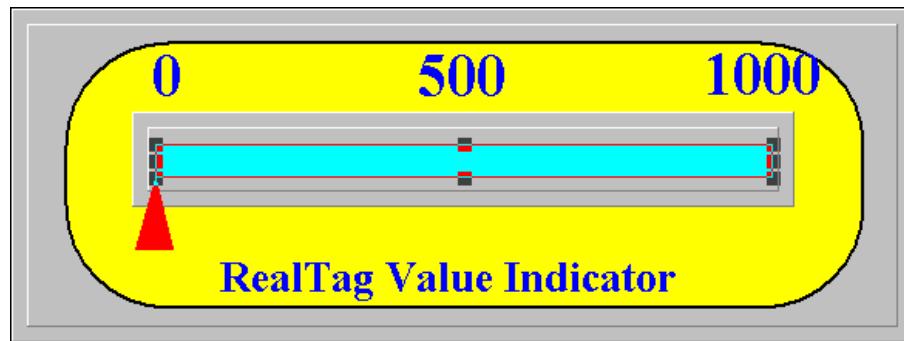
37. OK를 클릭합니다.

38. To Right / Horizontal Movement 확인하기 위해서 indicator bar를 선택합니다.
WindowMaker 윈도우의 하단에 있는 바의 넓이가 나타납니다.

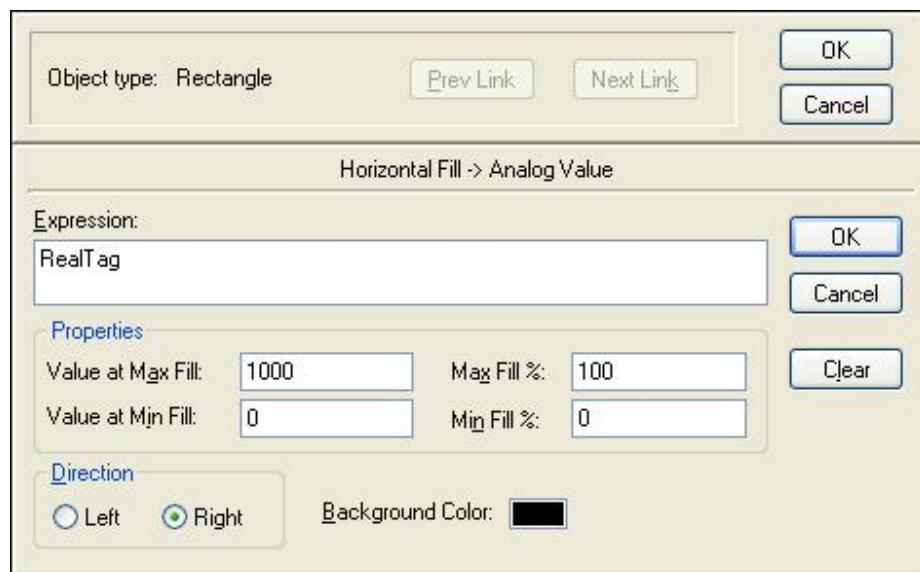


Percent Fill Link 추가

39. 슬라이드 객체위의 푸른색 Value Indicator바를 더블-클릭합니다.



40. Percent Fill / Horizontal을 선택합니다.



41. 아래그림과 같이 하기 위해 Horizontal Percent Fill 연결을 정의합니다.(배경색은 선택입니다.)

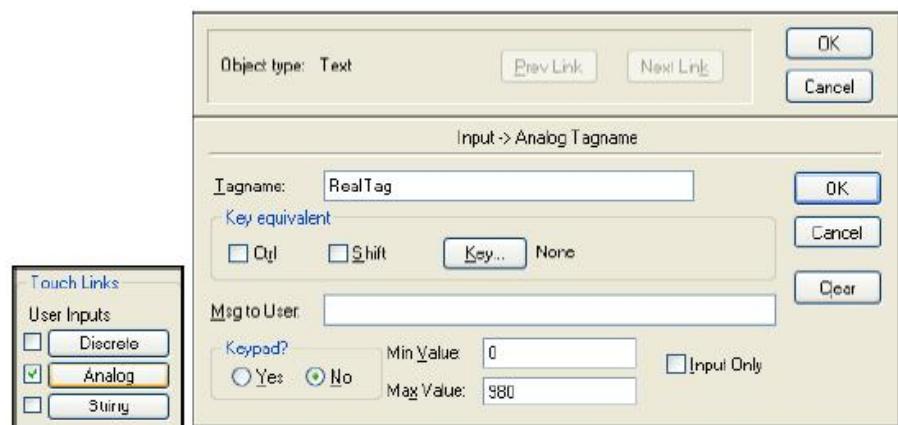
42. 상단의 OK버튼을 클릭합니다.

이산태그 텍스트 필드 움직임 주기

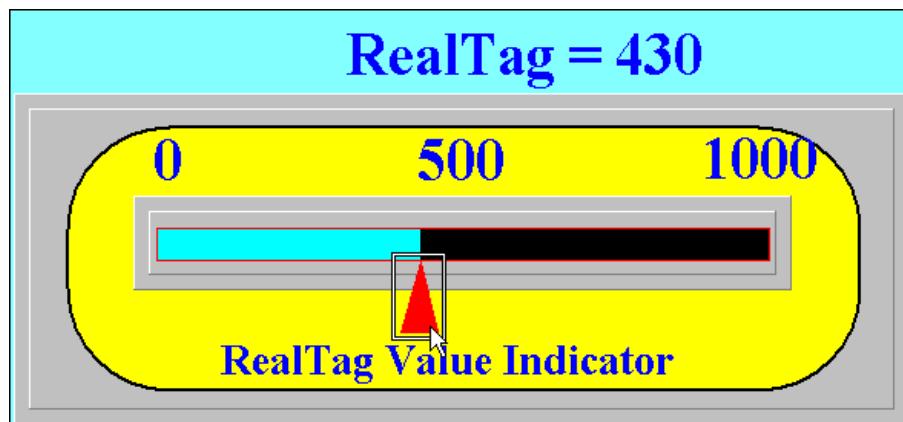
43. RealTag = # 텍스트 필드를 더블 클릭합니다.

44. Animation Selection 대화상자의 Links / User Inputs 선택안에 아날로그 버튼을 클릭합니다.

45. 아래 그림과 같이 연결을 설정 합니다.



46. OK를 클릭합니다.
47. **Runtime**으로 전환합니다.
48. 오른쪽 마우스 버튼을 클릭하고 유지하여 포인터를 움직입니다.
Indicator bar에 속한 값들은 태그에 보여진 값들과 일치됩니다.



49. Value display를 클릭합니다.
프레임은 활성화 되고 입력 필드가 나타납니다.

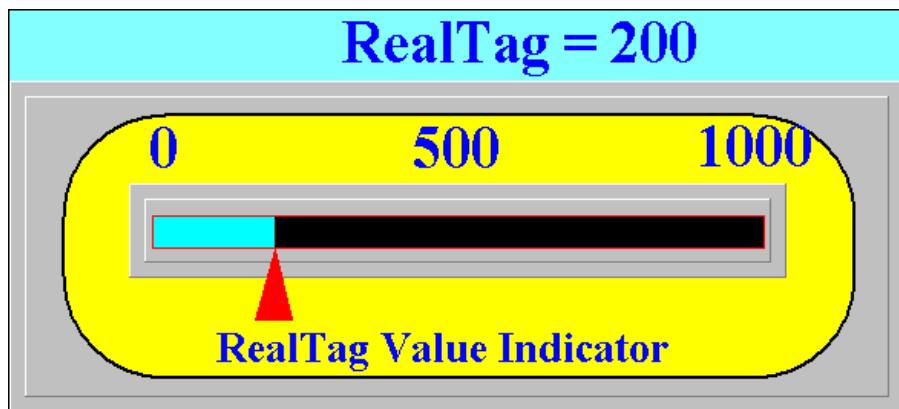


50. 한번 보여진 값과 다른 값을 입력하고 **Enter**를 누릅니다.(예로 값은 200으로 설정됩니다.)



Value indicator 가 이동했을것입니다. indicator bar 위에 percent fill 연결 작업을 보았을 것입니다.

RealTag = 지금 **200** 으로 보입니다.



51. 개발모드로 전환합니다.

타원 객체 연결들 설정

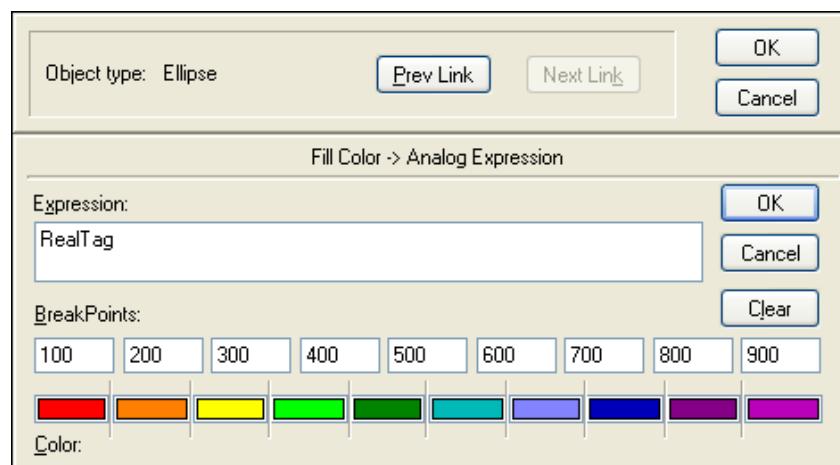
52. 타원 객체를 더블 클릭하고 아래와 같은 링크 속성들을 할당합니다.

Link Selection	Object Size - Height
Expression	RealTag
Value at Max Height	1000
Value at Min Height	0
Max % Height	100
Min % Height	0
Anchor	Middle

Link Selection	Object Size - Width
Expression	RealTag
Value at Max Width	1000
Value at Min Width	0
Max % Width	100
Min % Width	0
Anchor	Center

Link Selection	Fill Color - Analog
Expression	RealTag
Break Points	100, 200, 300, 400, 500, 600, 700, 800, 900
Color	9 colors of your choice

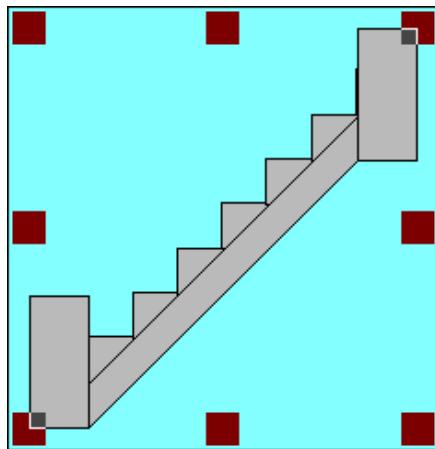
아래와 같은 Fill Color : Analog 설정이 보입니다.



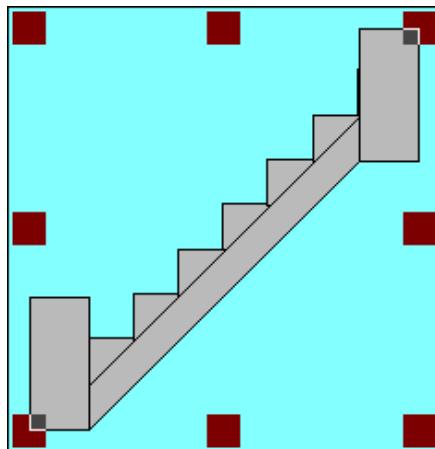
53. **Runtime** 모드로 전환합니다.
54. 포인터 뒤와 앞을 가로질러 슬라이드를 이동합니다. 포인터의 이동에 따라 원이 커지고 오그라들고 색이 변합니다.
55. 개발모드로 전환합니다.

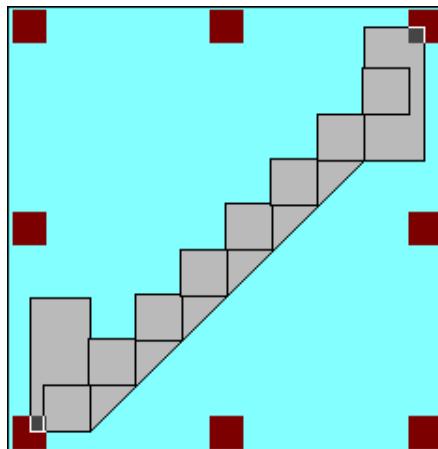
Escalator 객체 연결들 설정

56. Escalator 객체를 선택합니다.



57. Send to Back 버튼을 클릭합니다. 이 이미지는 여러 개의 객체들을 포함합니다.





58. 객체 단계를 선택합니다.(다중 입방)
59. 객체 단계를 아래와 같은 연결 속성으로 설정합니다.

Link Selection	Location - Vertical
Expression	\$Second MOD 10
At Top Value	10
At Bottom	0
Up	25
Down	0

Link Selection	Location - Horizontal
Expression	\$Second MOD 10
At Left End	0
At Right End	10
To Left	0
To Right	25

60. Railing 객체를 선택합니다.
61. Bring to Front 버튼을 클릭합니다.
62. **Runtime** 모드로 전환하고 움직이기 위해 나타난 단계를 지켜봅니다.

Section 4 - SmartSymbols 사용

소개

SmartSymbol은 인터치의 그래픽 요소들을 재사용이 가능한 템플릿으로 변환시켜주는 객체지향 기반의 기술입니다. SmartSymbol의 속성을 변화 시키면 어플리케이션 전체에 걸쳐 심지어 멀티 네트워크 PC 노드들도 자동적으로 템플릿으로 변경 시켜줍니다. . 결국 엔진지어들은 개발, 수정, 확인 그리고 HMI 어플리케이션 재확인 시간이 줄어듭니다.

SmartSymbol은 SmartSymbol Manager라는 기능을 이용하여 생성, 변경 및 삭제가 가능하며 이 기능은 ArchesrA 객체들, 로컬 인터치 태그들, remote references를 통한 인터치 태그들과 연동이 가능하게 됩니다. 이를 심볼 템플릿들은 SmartSymbol Library에 저장되고 인터치 윈도우에서 그들 선택과 그들 취소로 예시할 수 있습니다.

인터치 윈도우안에 어떤 그래픽은 셀로 만들어져 SmartSymbol로 변환 될수 있습니다. 또한, SmartSymbol의 라이브러리는 다른 어플리케이션과 공장, 기관 전체의 그래픽을 표준화 하는 회사들로 export 할 수 있습니다. SmartSymbol instance가 한번 디플로이 될 때 사용자가 instance의 변경이 필요하면 사용자는 SmartSymbol template을 간단히 수정하고 변경을 자동적으로 어플리케이션 전체에 걸쳐 적용시키고 모든 윈도우안의 SmartSymbol instance에 사용됩니다.

이것은 변경을 만들고 업그레이드 하고 빠르고 쉽게 어플리케이션을 수정합니다. 수정 후 확인과 재확인은 SmartSymbol과 함께 단순화됩니다..

SmartSymbol 생성하기

인터치에서 생성된 그래픽은 종류에 관계 없이 셀로 변환 하기만 한다면 모두 SmartSymbol로 변환이 가능합니다. 하지만 이런한 그래픽 요소 중에서도 SmartSymbol로 만들었을 때 가장 활용도가 높게 만드실려면 애니메이션 연결과 태그 정보를 같이 가지게 만드는게 좋습니다. 개발자는 SmartSymbol을 만들때, Application 안에 비슷하거나 동일하게 쓰일수있는 그래픽 개체가 존재하는지에 대해 생각해서 만들고, 또한 다른 Application에서의 재 사용성을 고려하게됩니다.. 두 방법 모두 반드시 그래픽과 시작되어야 합니다.

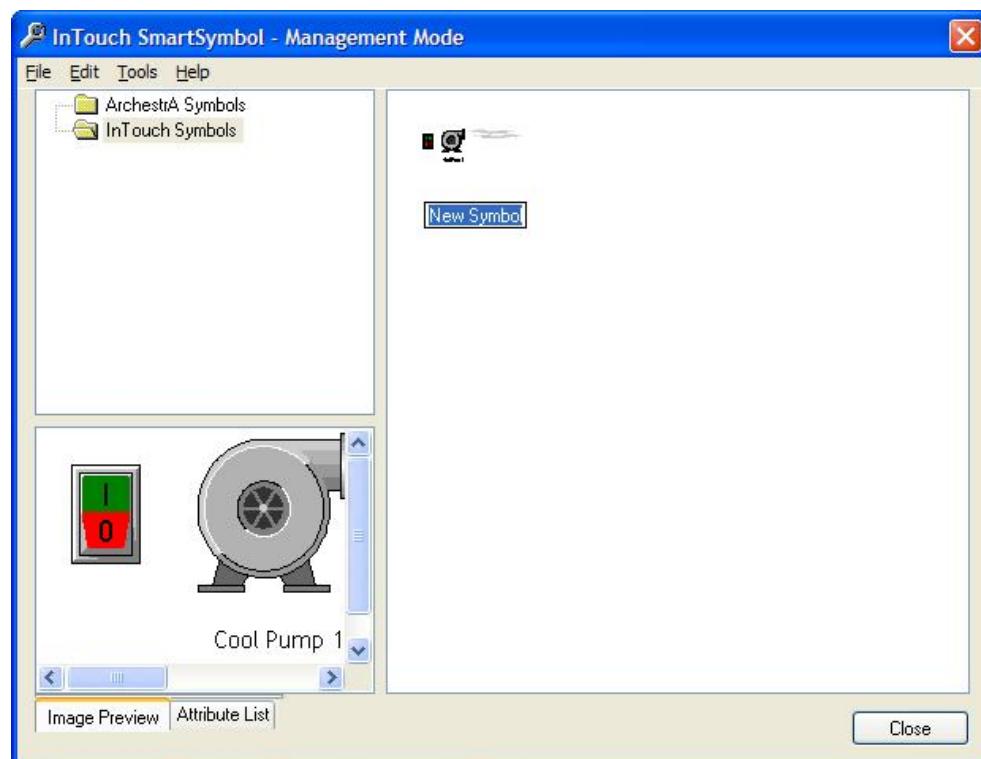
그래픽 설정

- a. 윈도우에 그래픽을 놓거나 생성합니다.
- b. 그래픽에 태그명이나 태그명들을 할당 합니다.
- c. 그래픽에 애니메이션 설정을 합니다.
- d. 전체 그래픽을 설정하고 Make Cell을 선택합니다.

SmartSymbol 생성하기

- a. 그래픽에서 오른쪽 마우스를 클릭하고 SmartSymbol / Generate SmartSymbol을 선택합니다.

새로운 심볼이 NewSymbol이라는 이름을 가지고 SmartSymbol - Management Mode 윈도우안에 나타날 것입니다.

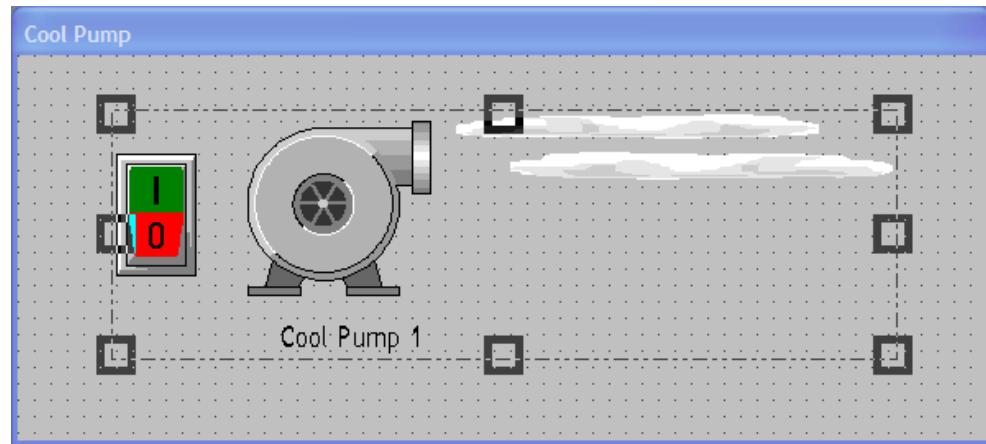


- b. NewSymbol을 클릭하고 이름을 변경합니다.
- c. Close를 클릭합니다.

새로운 SmartSymbol과 원래 셀을 대신하기 위해 WindowMaker 프롬프트창이 나타납니다..

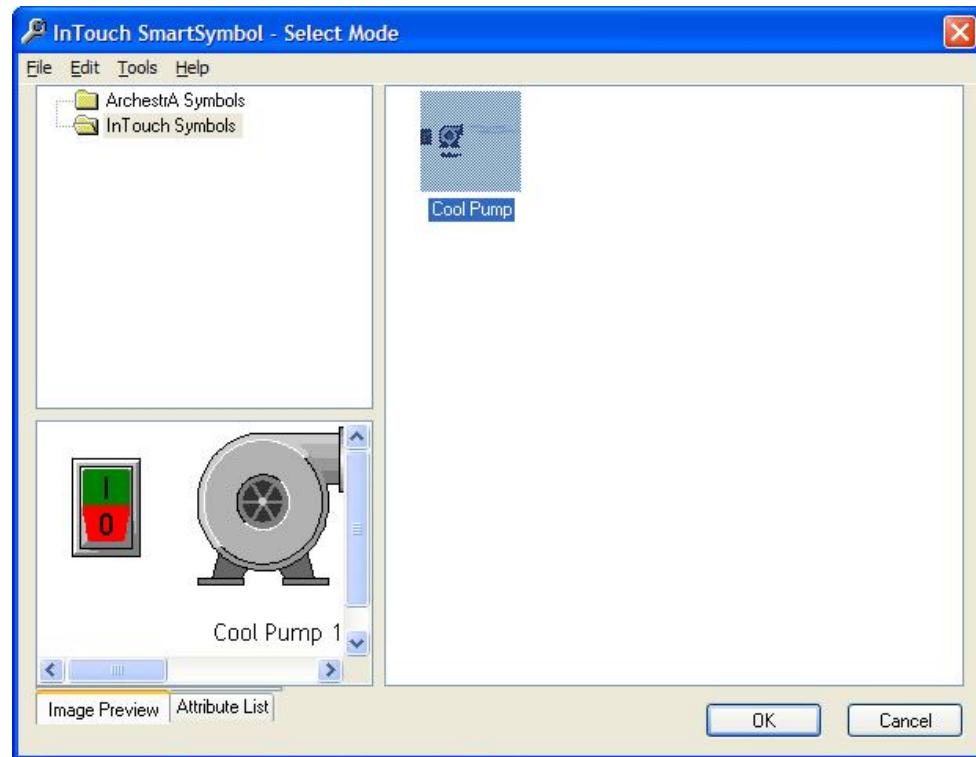
- d. Yes를 클릭합니다.

윈도우에 지금 SmartSymbol 디스플레이들이 sizing handles을 엽니다. 이것은 그것이 SmartSymbol이라는 것을 가리키고 셀이 아니라는 것을 가리킵니다.



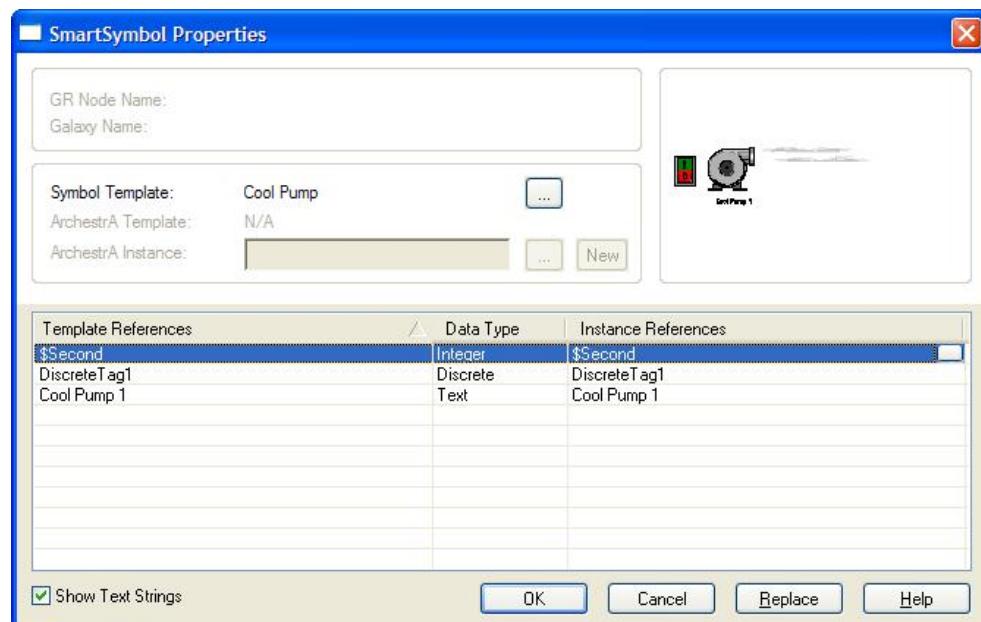
SmartSymbol Wizard

- a. 어플리케이션에 SmartSymbol을 놓기 위해 SmartSymbol Wizard 버튼을 클릭합니다. 커서가 입력 모드로 변경됩니다.
- b. InTouch SmartSymbol - Select Mode 윈도우에서 어플리케이션안에 원하는 SmartSymbol을 놓기 위해 원하는 SmartSymbol을 선택합니다.



c. OK를 클릭합니다.

SmartSymbol Properties 윈도우가 보여집니다.



Template References	Data Type	Instance References
\$Second	Integer	\$Second
DiscreteTag1	Discrete	DiscreteTag1
Cool Pump 1	Text	Cool Pump 1

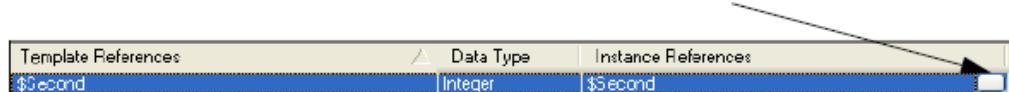
Show Text Strings

OK Cancel Replace Help

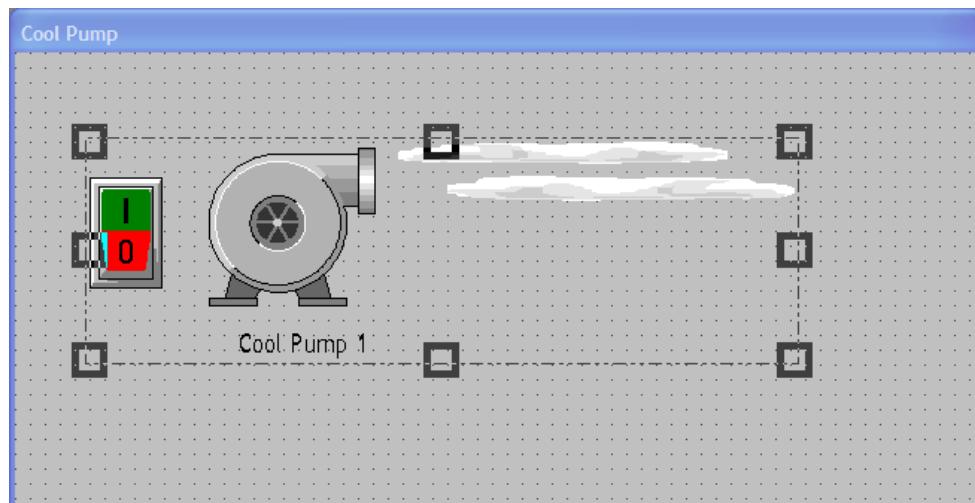
Template References: SmartSymbol 과 관련된 태그명을 생성합니다.(템플릿)

Instance References: SmartSymbol 과 관련된 태그명을 인터치 윈도우에 놓습니다.(인스턴스)

- d. SmartSymbol Properties window에서 Show Text Strings 체크박스를 클릭합니다.
- e. Instance References 칸에서 타원 버튼을 클릭합니다. 이것은 Tagname Browser의 접근할 것입니다.



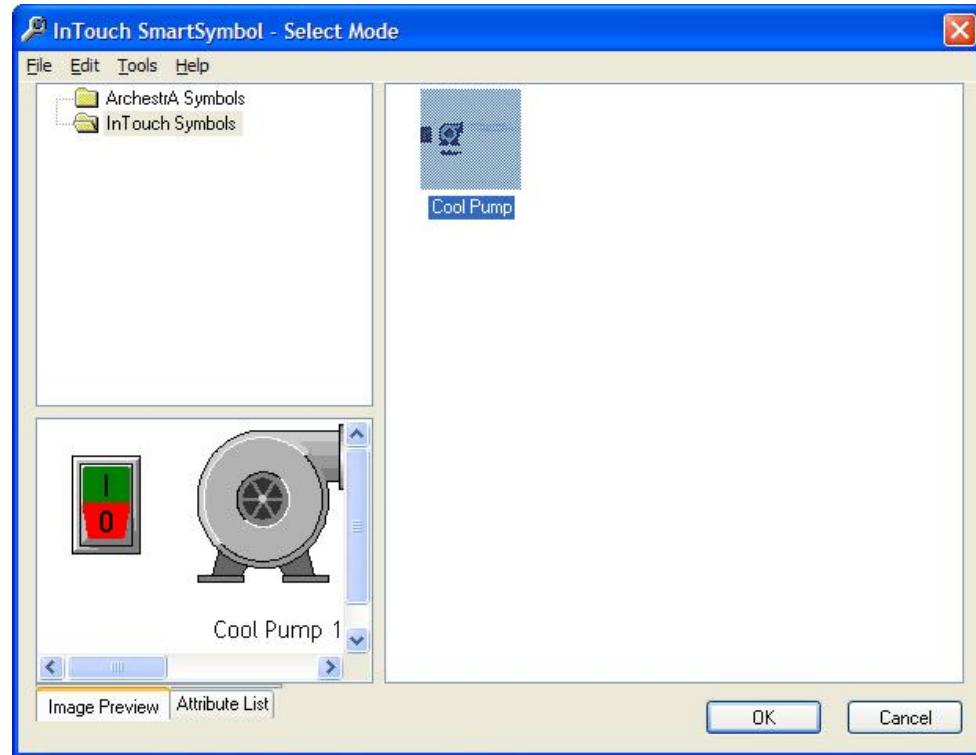
- f. 적당한 태그명을 선택합니다.
- g. 태그명을 선택후 선택한 태그명이 적용되는 것을 확인하기 위해 tab 키를 누릅니다..
- h. OK를 클릭합니다.
- i. 어플리케이션 윈도우안에 SmartSymbol이 나타날 것입니다.



Replace

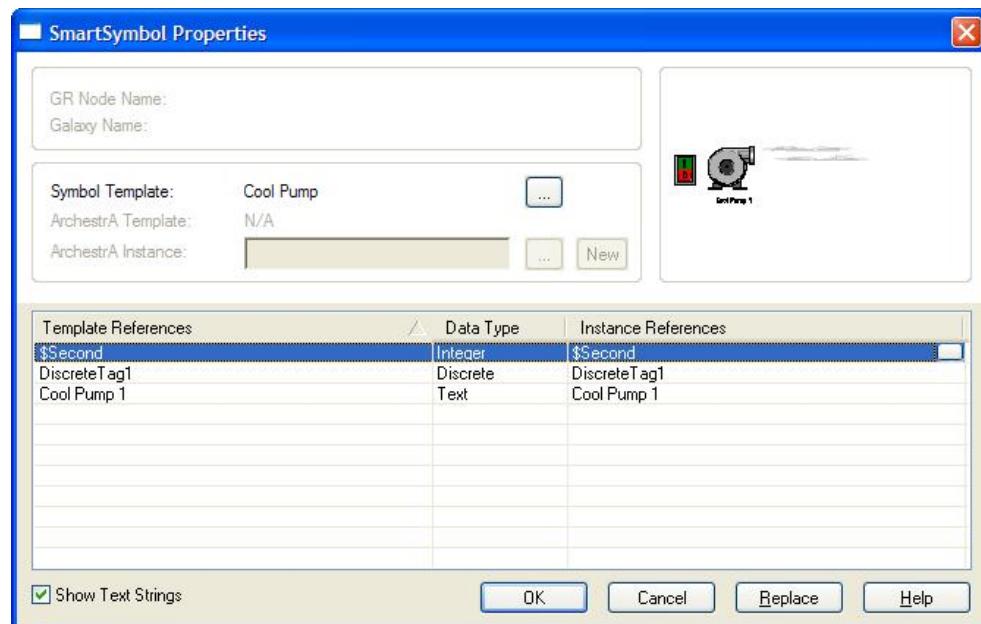
다른 방도로 여러 태그명과 문자열이 같은 Instance Reference 변경을 요구하면, Replace 특성을 사용합니다.

- a. 어플리케이션 윈도우에 SmartSymbol을 놓기위해 SmartSymbol Wizard 버튼을 클릭합니다. 커서가 입력모드로 변경됩니다.

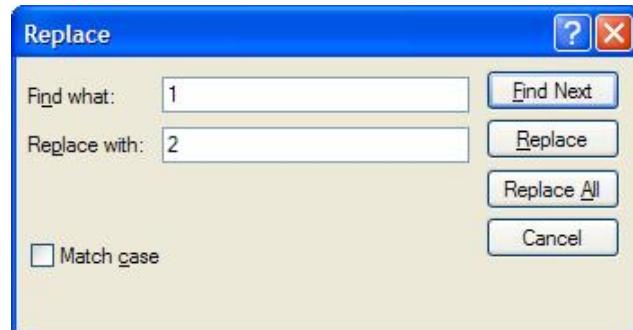


- b. SmartSymbol - Select Mode 윈도우에서 선택된 SmartSymbol이 어플리케이션에 놓아질 것 입니다. OK를 클릭합니다.

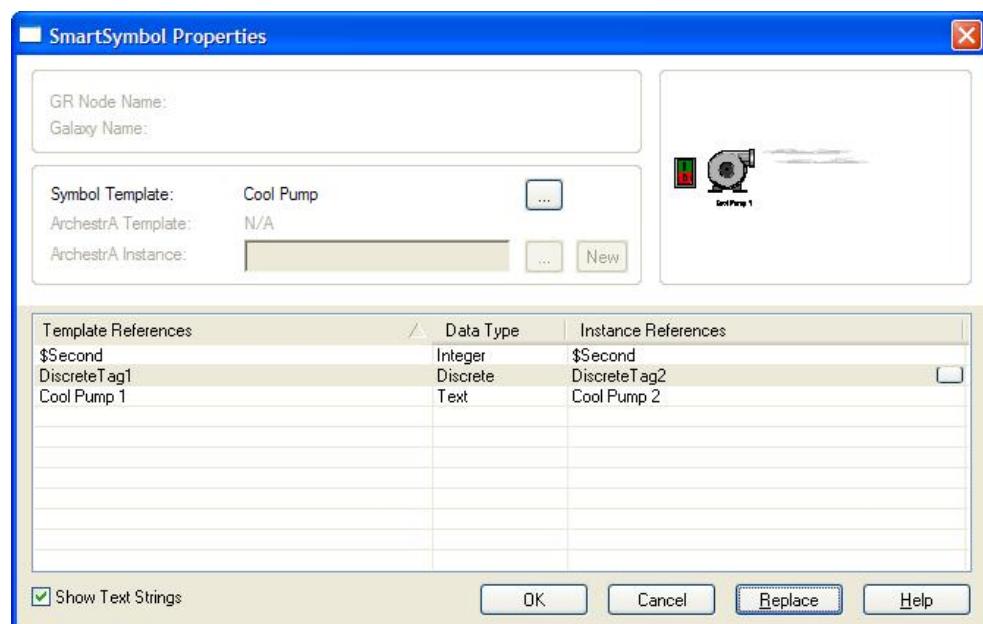
SmartSymbol 속성 대화상자가 나타납니다.



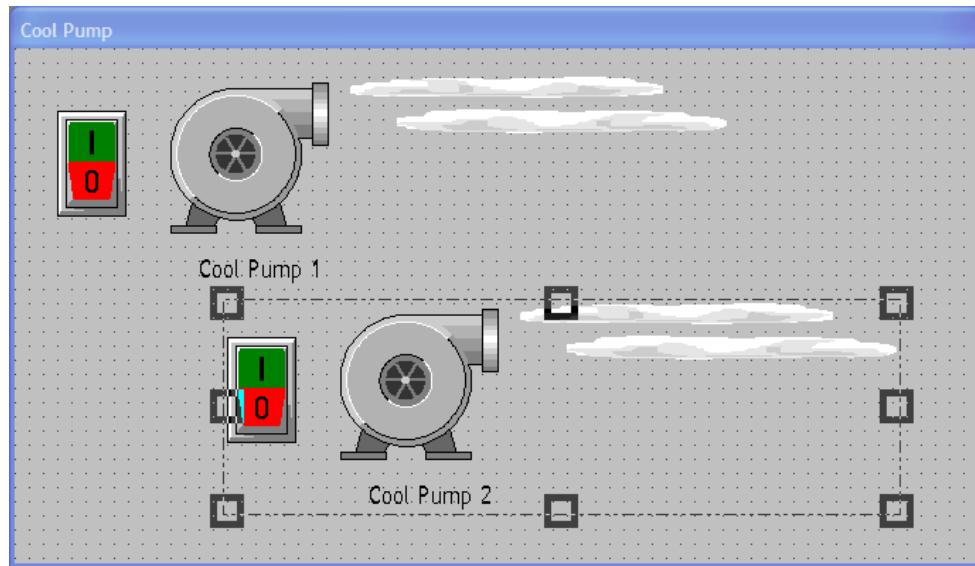
- c. Replace 버튼을 클릭합니다.
Replace 대화상자가 나타납니다.



- d. Find what과 Replace with 필드에 텍스트를 입력합니다.
e. Replace All를 클릭합니다.
f. 끝나면 Cancel를 클릭합니다. 태그명과 문자열 Instance References는 변경됩니다.



- g. OK를 클릭합니다.
SmartSymbol이 업데이트된 태그명과 함께 보여집니다.

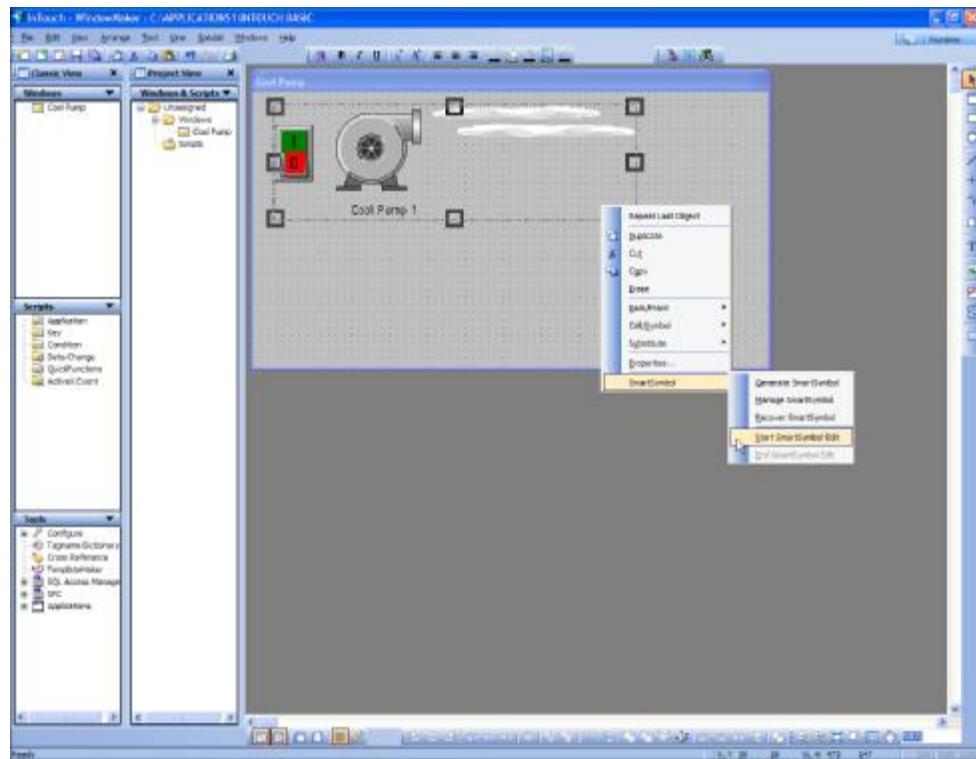


Edit SmartSymbols and Propagate Changes

어플리케이션에 다양한 윈도우들안에 여러 SmartSymbol들을 배치할때 변경또는 추가가 필요할때가 있습니다. 심볼과 수동적인 변경 적용의 인스턴스를 각각 찾기보다 인스턴스를 수정할수 있고 어플리케이션 윈도우를 통해 변경들을 전달 할 수 있습니다.

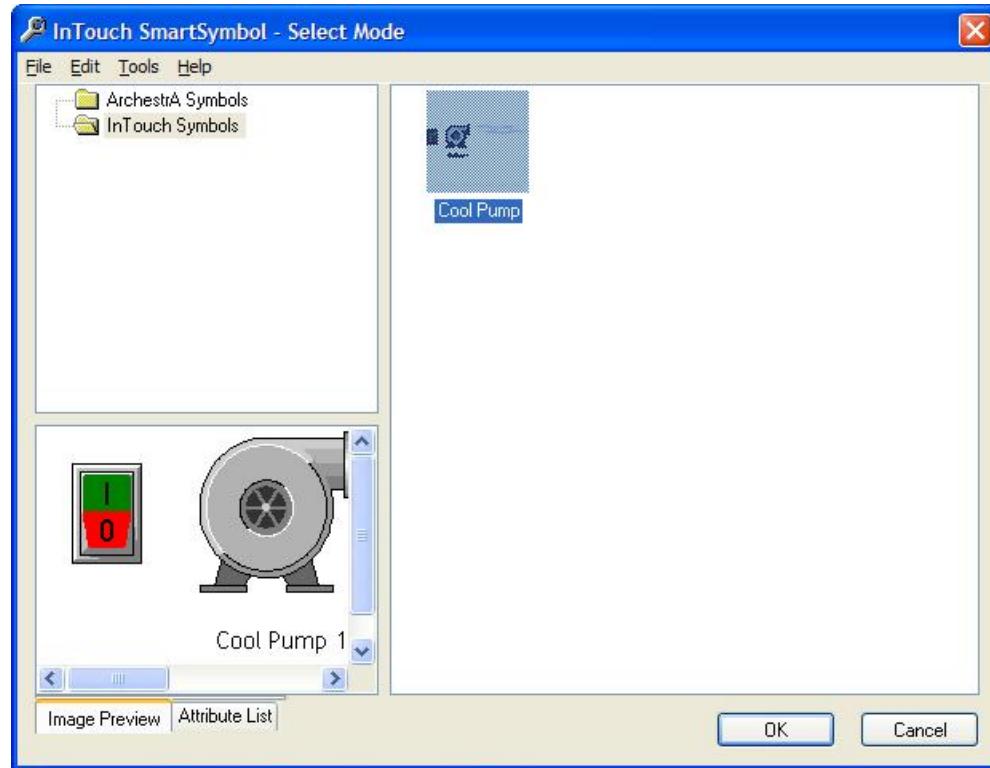
Start SmartSymbol Edit

- 어느 인터치 어플리케이션 윈도우의 빈 부분에서 마우스 오른쪽 버튼을 클릭합니다.

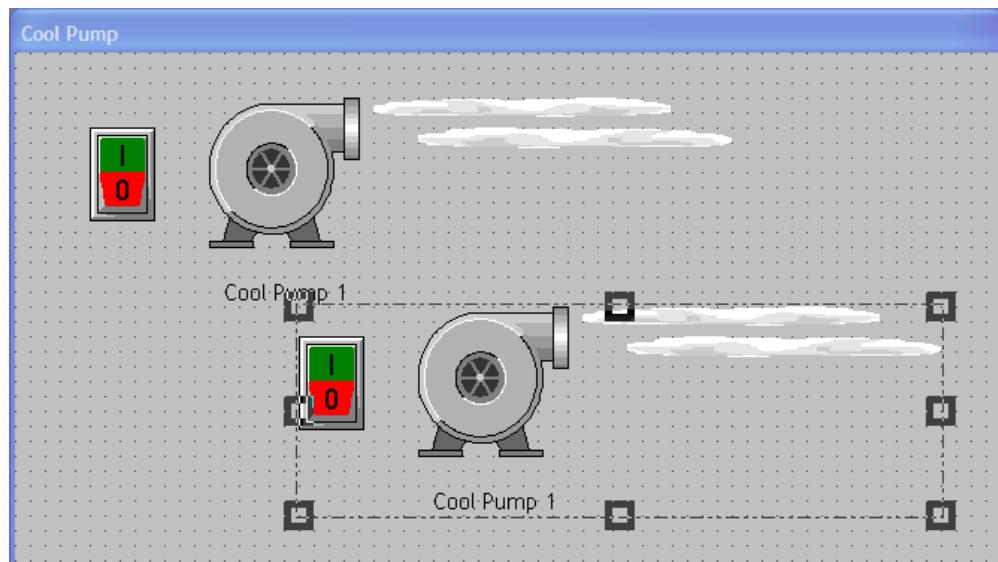


- b. Start SmartSymbol Edit를 선택합니다. (이 예제는 윈도우안에 이미 SmartSymbol를 보여줍니다.. 이 예제를 끝에서 수정모드가 반영되기 위해 그것이 변경되는 것을 볼수 있을 것입니다.

InTouch SmartSymbol : 대화상자 모드를 선택합니다.

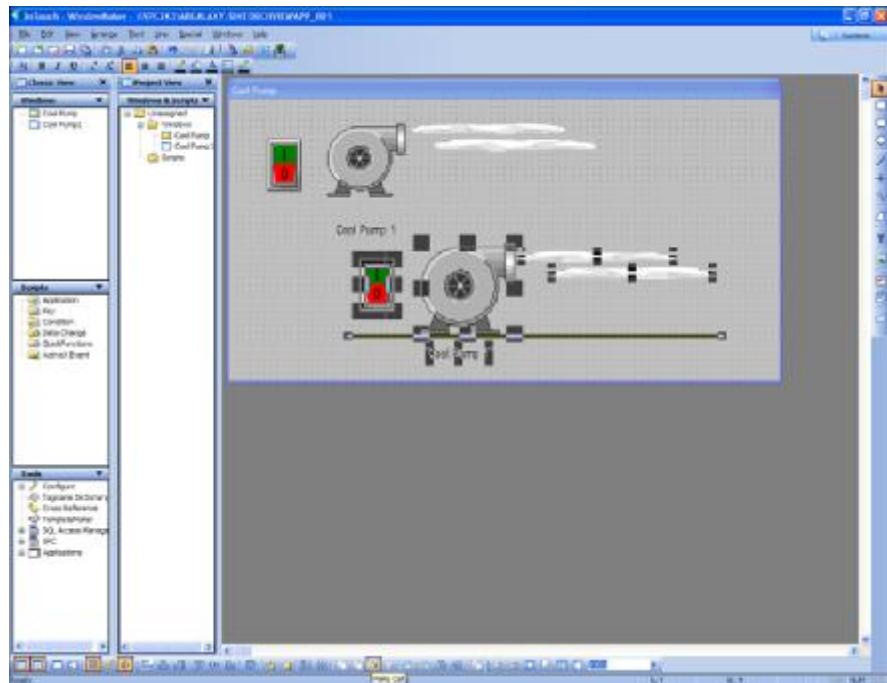


- c. 수정을 원하는 SmartSymbol를 선택합니다.
- d. OK를 클릭합니다.
- e. 수정을 하는 동안 일시적으로 SmartSymbol를 놓기 위해 윈도우의 열린 영역부분위에 커서를 클릭합니다.

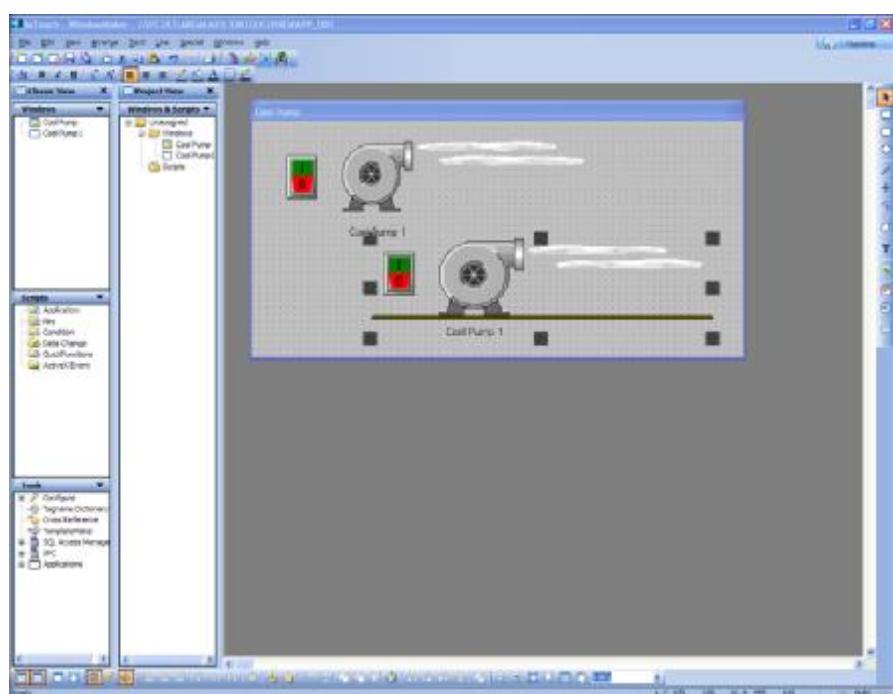


- f. 필요하다면 Break Cell을 클릭합니다.

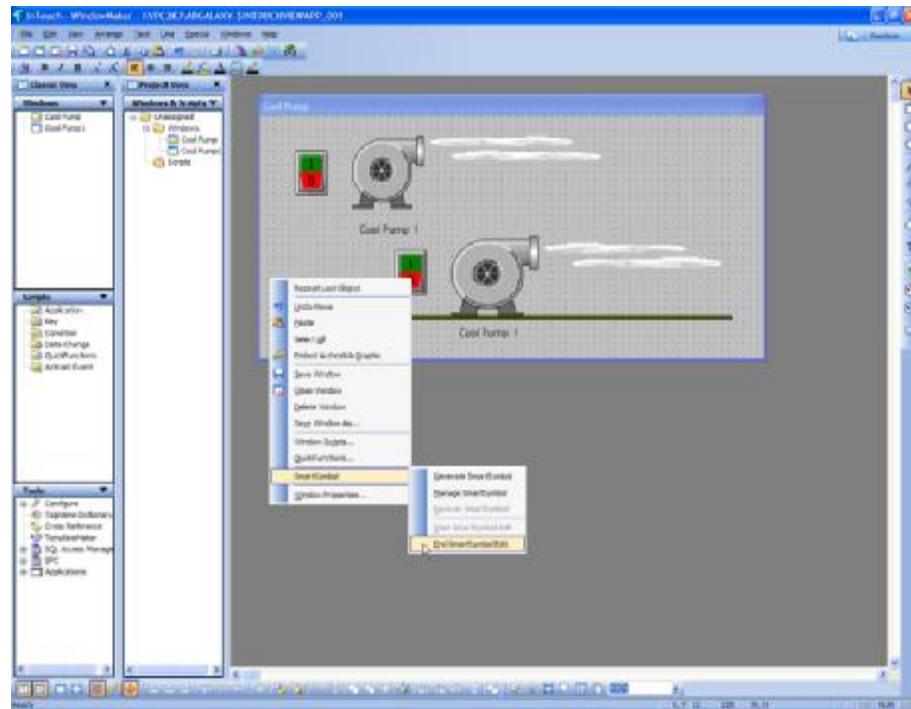
- g. SmartSymbol에 원하는 변경/추가를 생성합니다.



- h. 객체 전체를 선택합니다.
i. 그것을 셀로 만들기 위해 Make Cell 버튼을 클릭합니다.



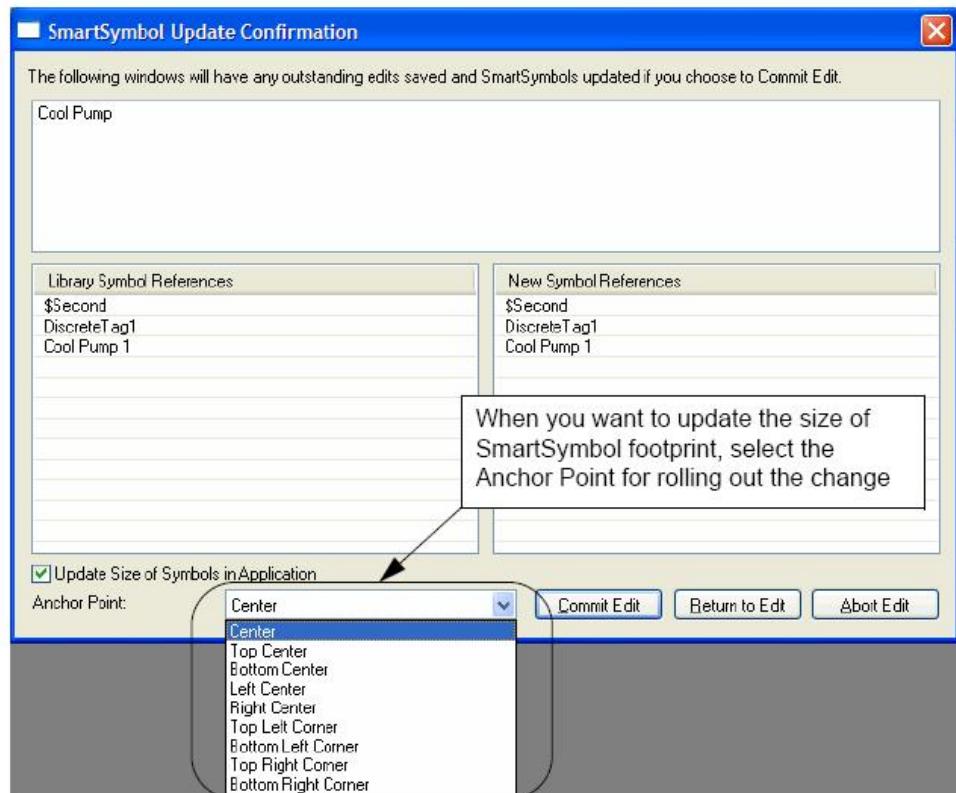
End SmartSymbol Edit



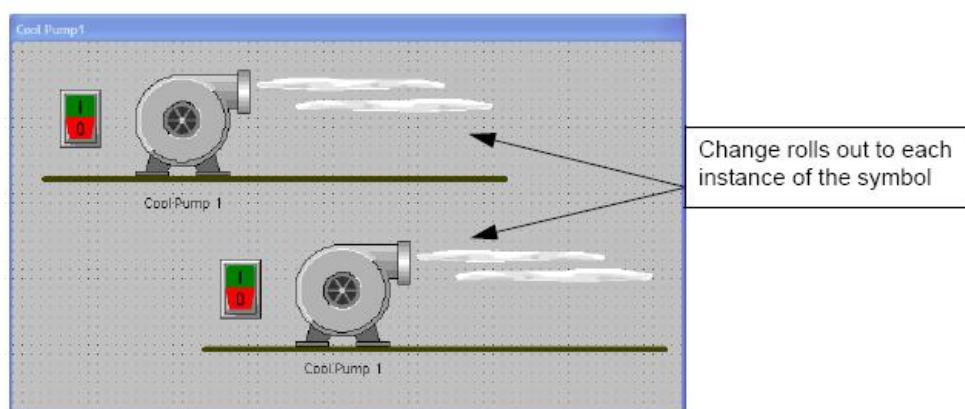
- j. 셀을 오른쪽 버튼 클릭하고 SmartSymbol / End SmartSymbol Edit를 선택합니다.

Commit Edit

- k. 어플리케이션의 모든 원도우들의 목록이 SmartSymbol을 포함하고 있습니다. 이들 원도우들 위의 SmartSymbol에 해당하는 설명이 수정 가능한 것을 보여주기 위해, Commit Edit를 선택합니다.



Note: SmartSymbol 을 수정 할 때,,아래와 같이 유지됩니다. 어플리케이션에서 Update Size of Symbol 을 실행 하면 예시된 SmartSymbol 의 차지양이 영향을 주게 되는 동안 윈도우위에 증가하고 있는 차지량이나 SmartSymbol 의 전체공간을 점유하게 됩니다. Update Size 를 선택하지 않으면 SmartSymbol 은 SmartSymbol 의 원래 크기를 유지할 것이며, 이미지는 자동적으로 원래 공간안에 적용 될 것입니다.



Manage SmartSymbols

SmartSymbol Manager 는 다양한 목적으로 사용 하실 수 있습니다.

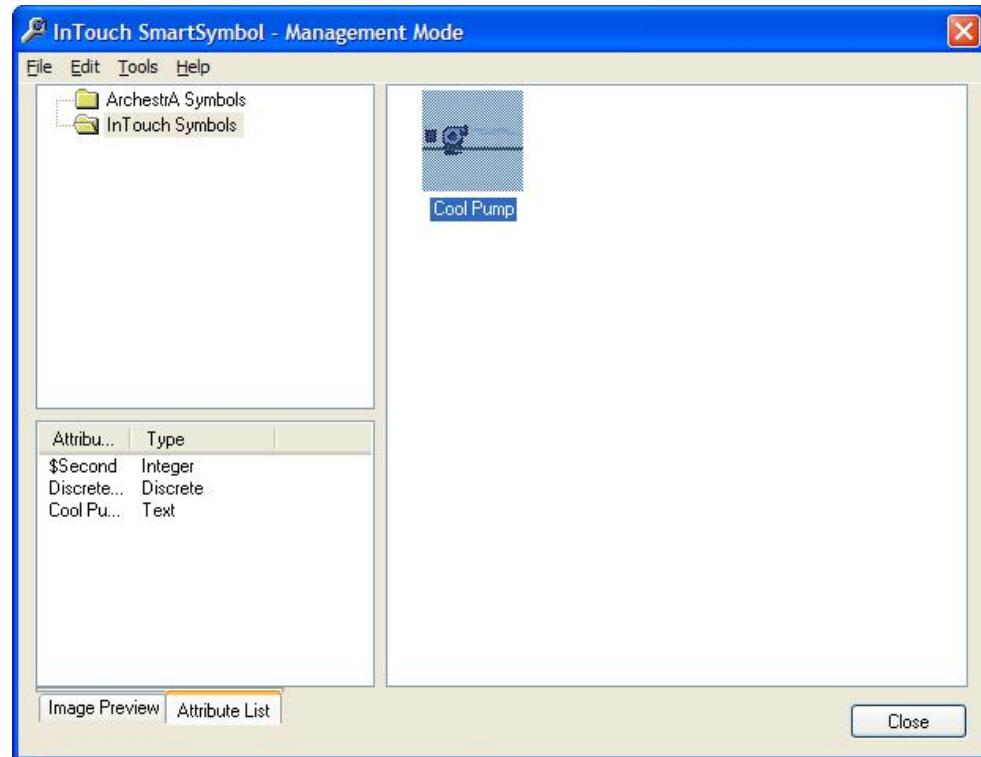
- | SmartSymbol의 속성들을 확인 할 때
- | SmartSymbol을 정리 할 때
- | SmartSymbol을 복사하거나 지울 때
- | SmartSymbol을 import하거나 export 할 때

SmartSymbol 속성 보기

SmartSymbol 을 생성하면, 태그명과 텍스트(문자열들)과 같은 속성들을 가지고 있습니다.

이런 속성들은 SmartSymbol instance 가 생성될 때 바뀔 수 있으며 아래와 같이 하시면 됩니다.

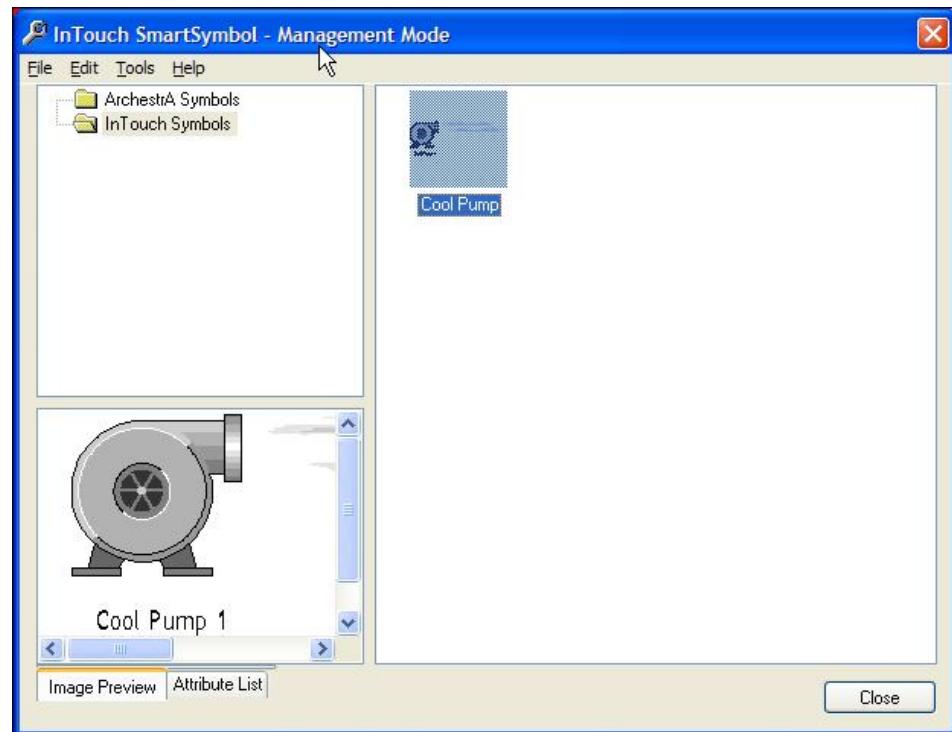
- a. Special / SmartSymbols / Manage SmartSymbol를 클릭합니다.
- b. SmartSymbol을 선택합니다.
- c. 좌측 하단에 Attribute List를 선택하시면 아래 그림과 같은 이미지를 확인 하실 수 있습니다.



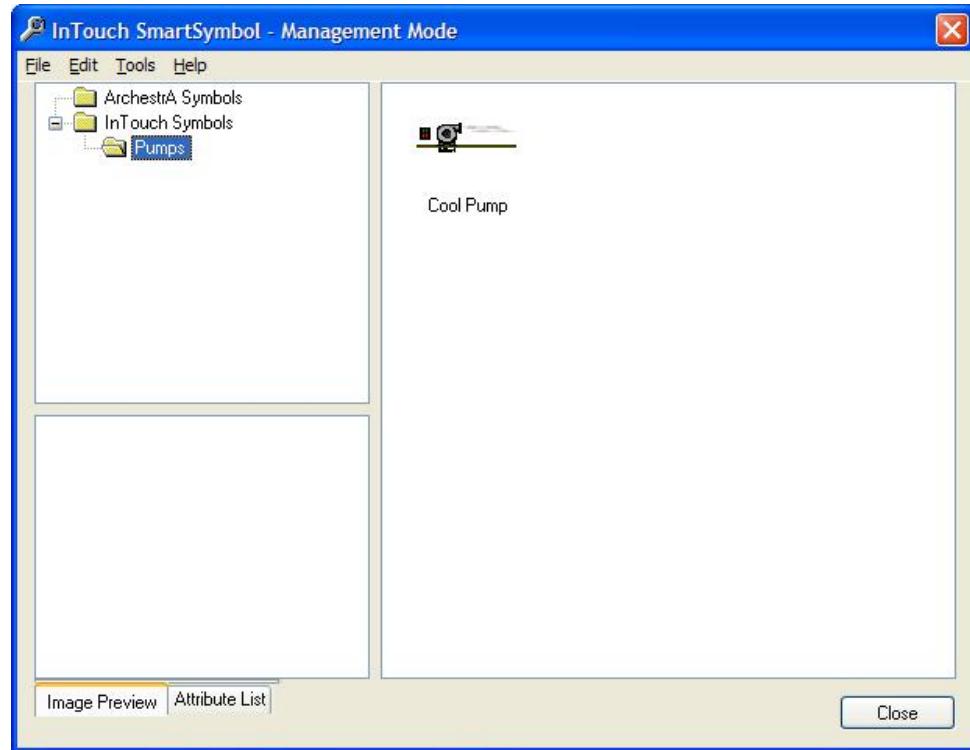
SmartSymbol 정리하기

SmartSymbol은 형태, 카테고리, 장소, 생성물선 또는 조직의 중요한 기관 방법이라든지 폴더들 안에 정리 될 수 있습니다. SmartSymbols을 분류 하시려면 SmartSymbol Manager를 실행 해야 합니다..

- a. Special / SmartSymbols / Manage SmartSymbol을 클릭합니다.
SmartSymbol - Management Mode 윈도우를 보여줍니다.



- b. 그것에 포커스를 주기 위해 InTouch Symbol을 클릭합니다.
- c. File / New Folder를 클릭합니다.
- d. 새 폴더 이름은 선택한 심볼 종류에 맞게 이름 지어집니다..



- e. 인터치 Symbol폴더로 부터 적용 폴더로 Symbol을 드래그 합니다.

SmartSymbol 이름 변경, 복사, 삭제, 복원 하기

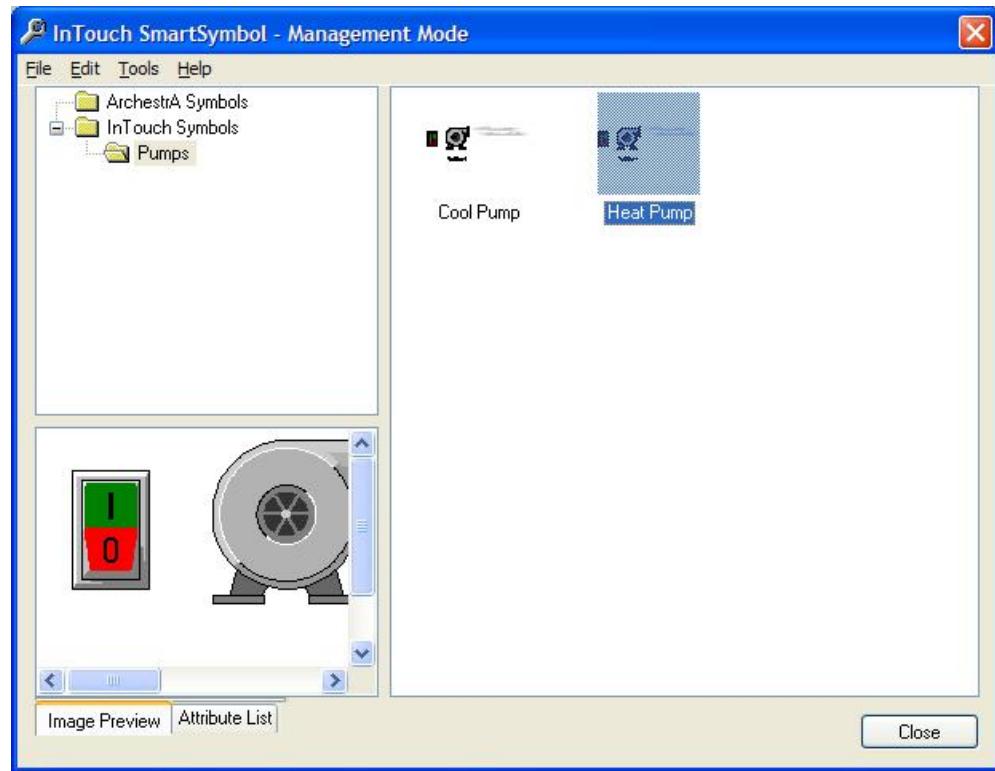
SmartSymbol Manager 를 이용하시면 SmartSymbol 의 이름 변경, 복사, 삭제, 복원 하기가 모두 가능합니다.

SmartSymbol 의 이름을 변경하기 위해

- 그것에 포커스를 주기 위해 SmartSymbol을 클릭합니다.
- Edit / Rename를 클릭합니다.
- 적절한 이름으로 변경하신 후에 Enter를 누릅니다.
- SmartSymbol의 새이름이 보여 줍니다.

SmartSymbol 을 복사하기 위해

- 그것에 포커스를 주기 위해 SmartSymbol Manager안의 SmartSymbol을 클릭합니다.
- Edit / Copy를 클릭합니다.
- 새로운 SmartSymbol이 나타나기 원하는 곳의 위에 폴더를 클릭합니다.
- Edit / Paste를 클릭합니다.

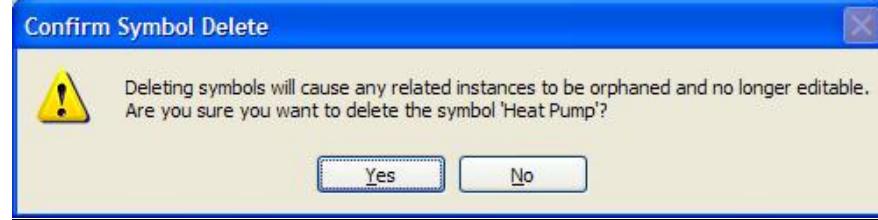


새로운 SmartSymbol이 나타납니다. 원래처럼 같은 폴더를 놓는다면 그것의 이름은 <original name>의 이름이 복사 되어질 것입니다.

SmartSymbol을 삭제하기 위해

- 그것에 포커스를 주기 위해 SmartSymbol을 클릭합니다.
- File / Delete를 클릭합니다.

Confirm Symbol Delete 윈도우가 보여집니다.



삭제된 SmartSymbol을 복원하기 위해

- 인터넷 어플리케이션 윈도우에 SmartSymbol을 대신해 남아 있는 SmartSymbol를 클릭합니다.

Note: 만약에 한번도 복사 하실 SmartSymbol을 사용하지 않으셨다면, 그 SmartSymbol은 복구 될 수 없습니다.

- Special / SmartSymbol / Recover SmartSymbol을 클릭합니다.
- SmartSymbol이 생성될것이며 SmartSymbol안에 보여 질것입니다. 관리 모드 윈도우와 New Symbol의 이름. **New Symbol** 적절하게 재명명합니다.

SmartSymbol Library 내보내기

다른 인터치 어플리케이션들에 사용하기 위해 SmartSymbol Library를 export 할 수 있습니다.

- SmartSymbol Manager를 작동합니다.
- File / Export를 클릭합니다.



- Export 할 SmartSymbol 파일에 이름을 주고, 선택한 디렉토리에 놓습니다.
- Save를 클릭합니다. Export된 SmartSymbol library는 지금 어떤 인터치 어플리케이션으로도 import 할 수 있습니다.

SmartSymbol Library 가져오기

- e. 원하는 SmartSymbols library를 import 위해서 인터치 어플리케이션을 엽니다.
- f. 모든 InTouch Application Windows 를 확실히 닫아줍니다.
- g. Special / SmartSymbol / Manage SmartSymbols를 클릭합니다.
- h. File / Import를 클릭합니다.
- i. Import 하길 원하는 .www 파일을 클릭하고 Open를 클릭합니다.

어떤 어플리케이션에 SmartSymbol 들이 존재하고 이들처럼 import 된 것들이 같은 이름이라면 그들을 건너뛰우고 겹쳐쓰기하거나 재명명을 선택 할 수 있습니다.

- j. OK를 클릭합니다.

Lab 4 - SmartSymbols 사용

소개

다음 랩은 어떻게 SmartSymbol 들을 생성하고 조작하는지를 설명합니다. 새로운 SmartSymbol 를 생성하고 수정하고 SmartSymbol과 다른것들을 관리할 것입니다.

랩 1에서, 물리적인 공장 객체를 대표하기 위해 그래픽 객체 사용을 생성하였습니다. 랩 3에서는 밸브 열고 닫기를 테스트하기 위해 밸브라고 불려지는 DiscTag1에 메모리 이산태그명을 정의했습니다. 또한 어떻게 밸브에 움직임을 주고 그것의 상태(DiscTag1 값의 On, Off 사용)를 보여주는지도 배웠습니다. 밸브 복사와 변경은 값을 움직임들의 고정을 대표하는 DiscTag2로 평가됩니다. 지금 밸브 symbol들과 센서 둘다 추가 변경 할 것입니다. 각각의 symbol 을 수정하기 보다 아래의 랩에서 하나의 symbol 을 셀로 변경 할 것이고, Links 윈도우에서 SmartSymbol 로 변경되는 SmartSymbol 를 바꿉니다.

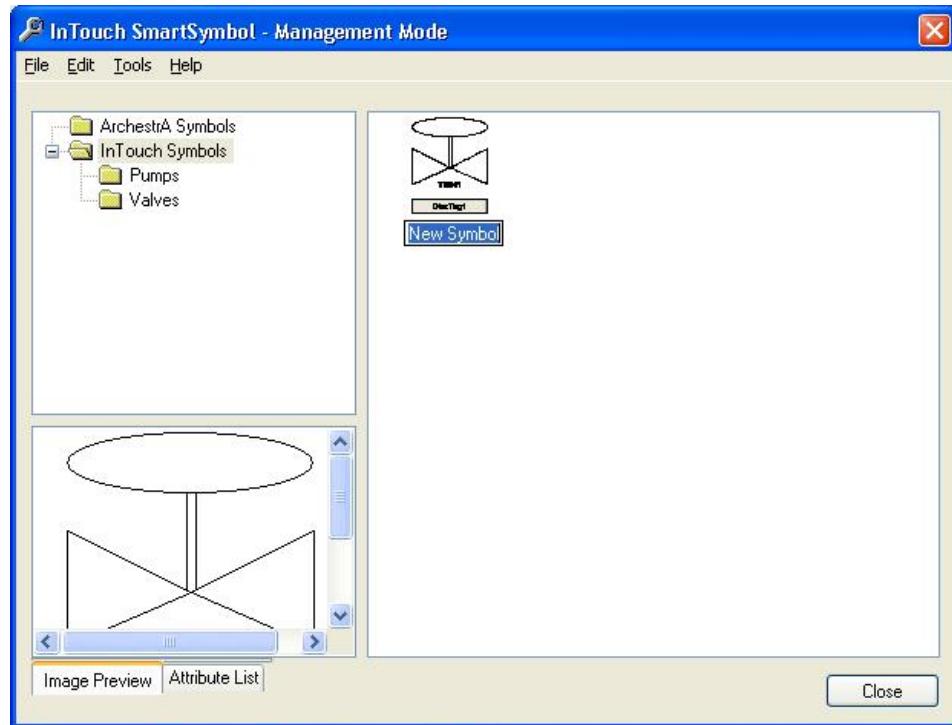
Detailed Lab

SmartSymbol 생성하기

SmartSymbol 생성

이 랩은 존재하는 Valve1 그래픽에서 SmartSymbol로 변경에 의해 시작될것입니다.

1. Links 가 열려있지 않으면, 인터치 어플리케이션에서 Links 윈도우를 엽니다.
2. 그것을 선택하기 위해 Valve1 symbol을 클릭합니다.
3. Shift 키를 유지한채 DiscTag1버튼을 클릭합니다.
4. Make Cell 버튼을 클릭합니다.
5. 새로운 셀을 오른쪽 버튼 클릭하고 SmartSymbol / Generate SmartSymbol를 선택합니다.
6. SmartSymbol - Management Mode 윈도우에 새로운 SmartSymbol과 새로운 NewSymbol. 이 나타날것입니다.

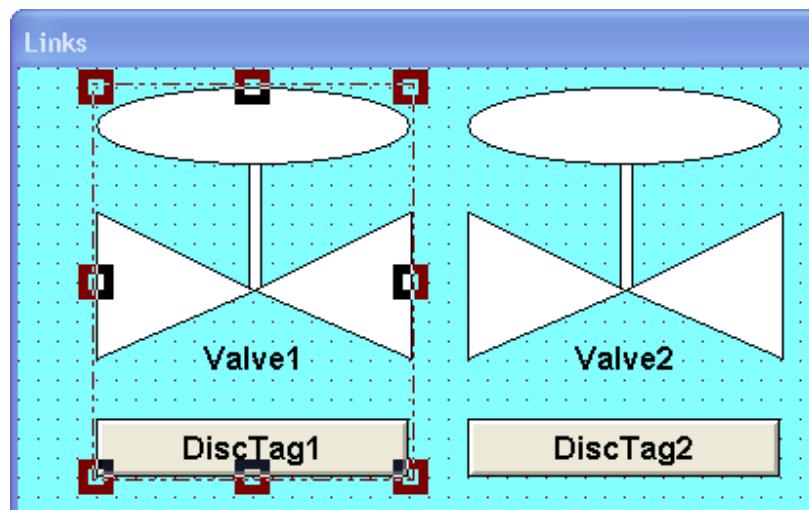


7. 수정 가능한 새로운 필드를 만들기 위해 NewSymbol를 클릭합니다.
8. 이름을 Valve로 변경합니다.
9. SmartSymbol - Management Mode 윈도우를 닫기 위해 Close를 클릭합니다.

새로운 셀과 새로운 SmartSymbol을 재배치하기 위해 WindowMaker 프롬프트 창이 나타납니다.



10. Yes를 클릭합니다.
11. Links 윈도우안에 Valve1은 지금 SmartSymbol이며, 선택 되었을때 핸들들이 보입니다.

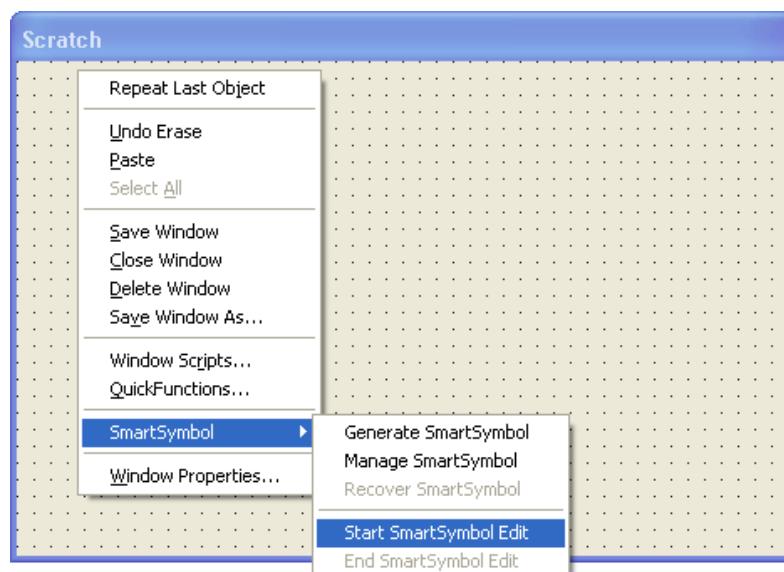


SmartSymbols 설정과 변경을 전파

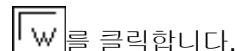
지금 Sensor.를 보여주기 위해 Valve SmartSymbol 을 변경 할 것입니다.

SmartSymbol 설정 시작

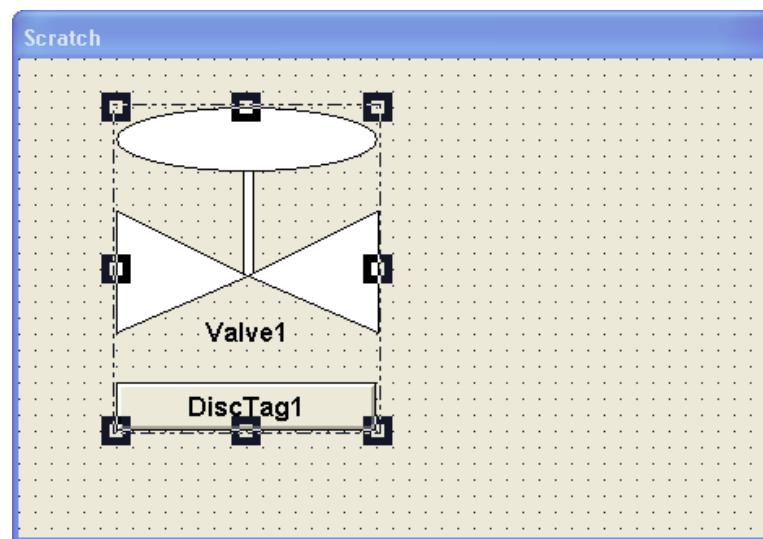
12. Links 윈도우를 닫고 Scratch윈도우를 엽니다.
13. 윈도우의 공백 부분에서 오른쪽 버튼을 클릭합니다.
14. SmartSymbol / Start SmartSymbol Edit를 선택합니다.



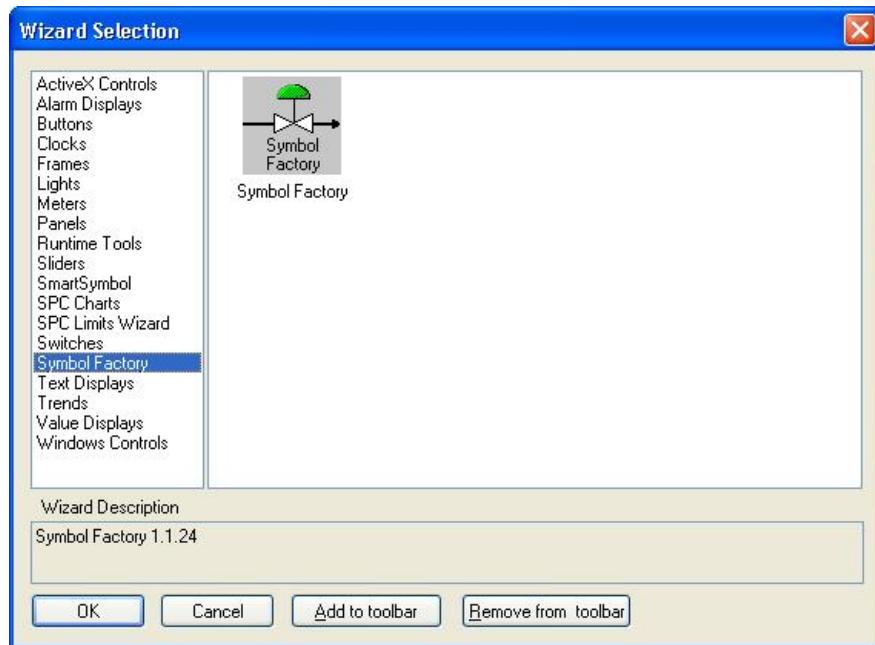
15. 수정하는 동안 임시적으로 SmartSymbol 를 놓기 위해 윈도우 공백지역의 커서



16. Valve SmartSymbol 선택하고 OK를 클릭합니다. Valve SmartSymbol 보여집니다.



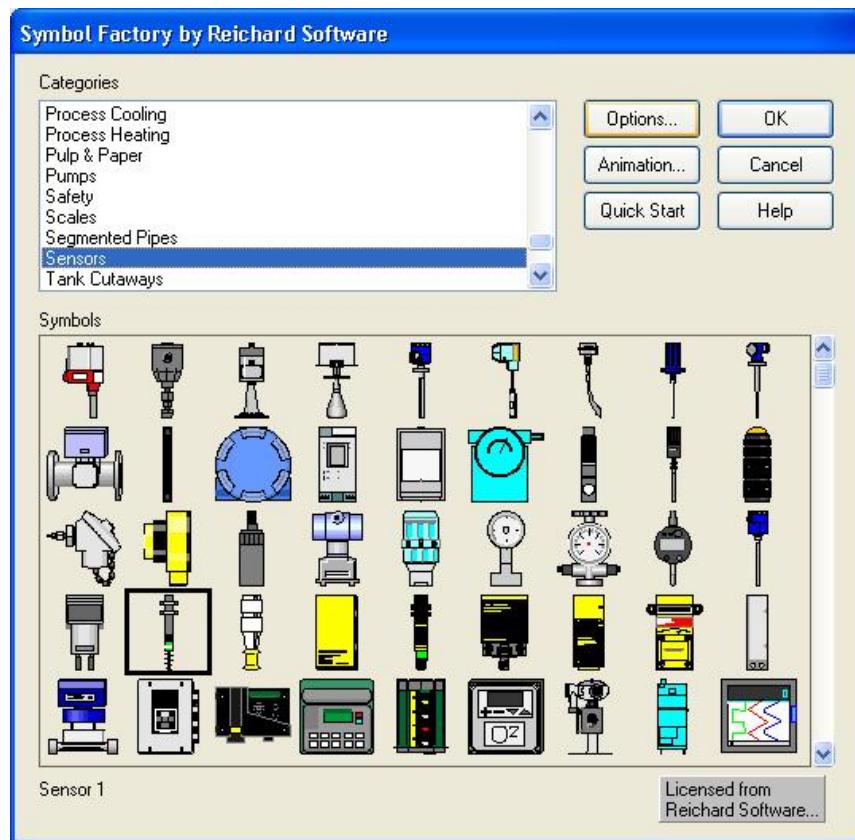
17. Wizards button을 클릭합니다.



18. Symbol Factory를 클릭하고, Symbol Factory Wizard를 선택하고 OK를 클릭합니다.

19. Valve 1 문자 위의 커서를 클릭합니다.

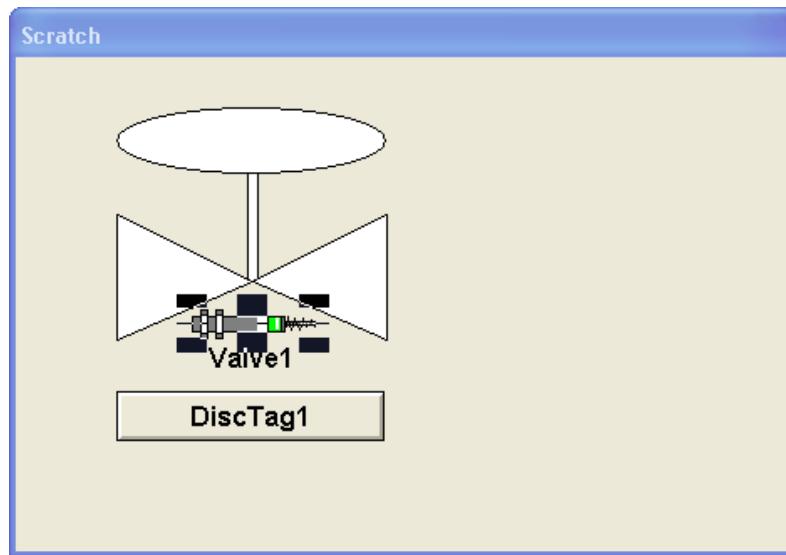
Symbol Factory 대화 상자를 보여줍니다.



20. Sensors 카테고리안에 Sensor1를 찾습니다.

Hint: 카테고리를 선택하고, 형태를 SEN으로 합니다. 밝게된 카테고리가 Sensors로 변할 것입니다.

21. Flip Horizontal 체크박스를 클릭하고 +90버튼을 클릭합니다.
22. Options를 클릭합니다.
23. Symbol Options 윈도우에서 OK를 클릭하고 Symbol Factory 윈도우에서 OK를 클릭합니다. Sensor가 나타납니다.



Note: Sensor 을 놓은 후에 셀 쪼개기와 회전 선택 또는 Sensor 더블 클릭과 옵션으로
가기로에 의해 Sensor 을 회전 시킬 수 있습니다. 맨나중 방법은 원래 차지하는
공간안에서 객체를 회전시키는 것이고 원하는 효과가 일어나지 않을 것입니다.

24. 객체 둘다 선택하고 Make Cell 버튼을 클릭합니다.

SmartSymbol 수정 끝내기

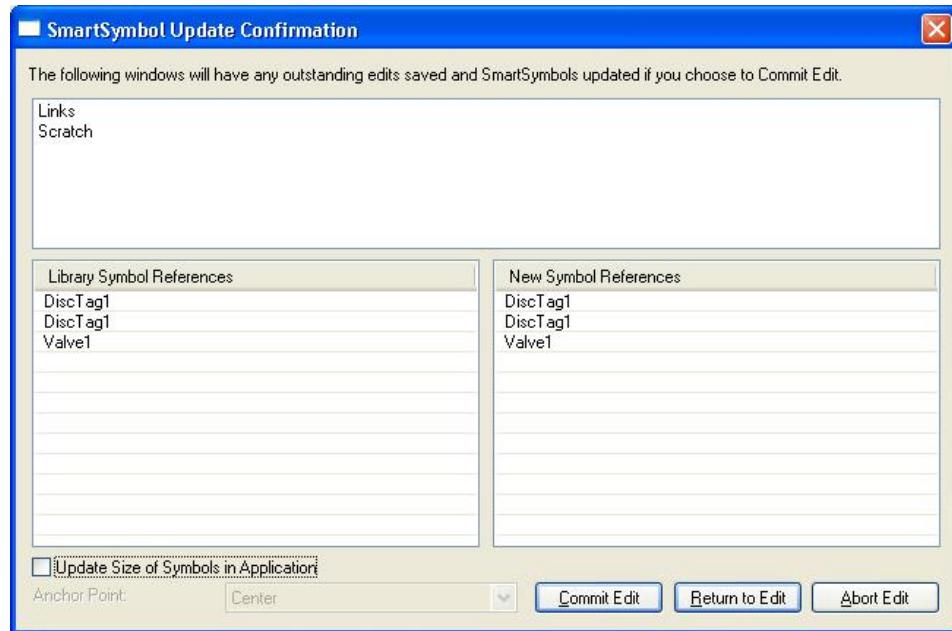
25. 셀을 오른쪽 버튼 클릭하고, SmartSymbol / End SmartSymbol Edit를 선택합니다.

Commit Edit

어플리케이션에서 모든 윈도우 목록은 수정된 SmartSymbol 을 포함하고 있습니다. 그 수정은 이들 윈도우들 위에 SmartSymbol 들로 전달 되는 것이 확인됩니다.

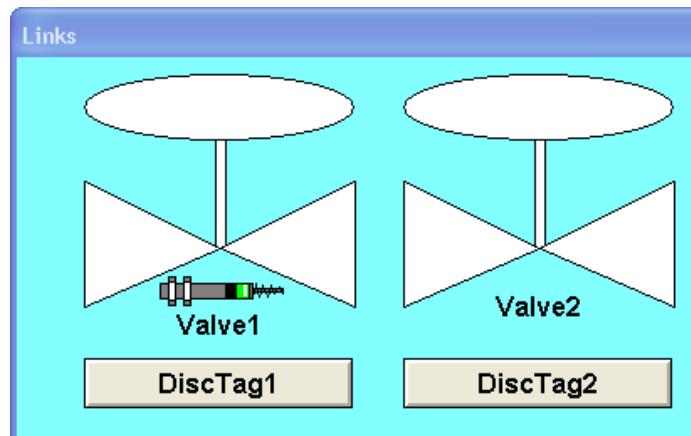
New Symbol References 칸에 어떠한 태그명으로의 변경들도 확인을 위해 보여 집니다.
Library Symbol References 칸의 값들은 New Symbol References 칸에 맞춰 지게 되어 변경이
없게 만들어집니다.

26. Commit Edit 버튼을 클릭합니다.



27. Scratch 윈도우를 닫습니다. 만일 저장하지 않았다면 저장합니다.

28. Links 윈도우를 엽니다. Valve2가 변경되지 않은 것을 확인합니다.



지금 자신의 객체가 자신의 것입니다.

왜 Valve2 는 새로운 Sensor 를 보여주지 않는 이유는?

Hint: Replace” on page 4-71 를 언급합니다.

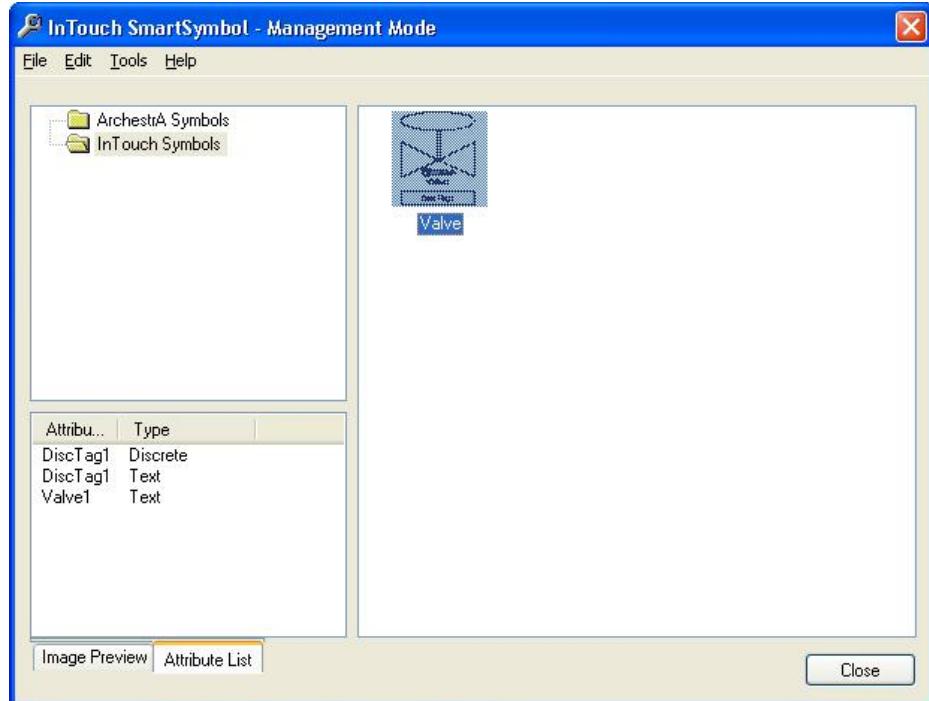
SmartSymbol 관리하기

랩의 이번 단락은 아래와 같은 것들을 합니다.

- | SmartSymbol 속성을 보여줍니다.
- | SmartSymbol 정리합니다.
- | SmartSymbol을 재명명하고 복사하고 삭제하고 복구합니다.
- | SmartSymbols를 가져오고 내보내기

SmartSymbols 속성들을 보여줍니다.

29. SmartSymbol Manager (Special / SmartSymbol / Manage SmartSymbol)를 작동합니다.
30. 그것에 포커스를 주기 위해 Valve SmartSymbol을 클릭합니다.

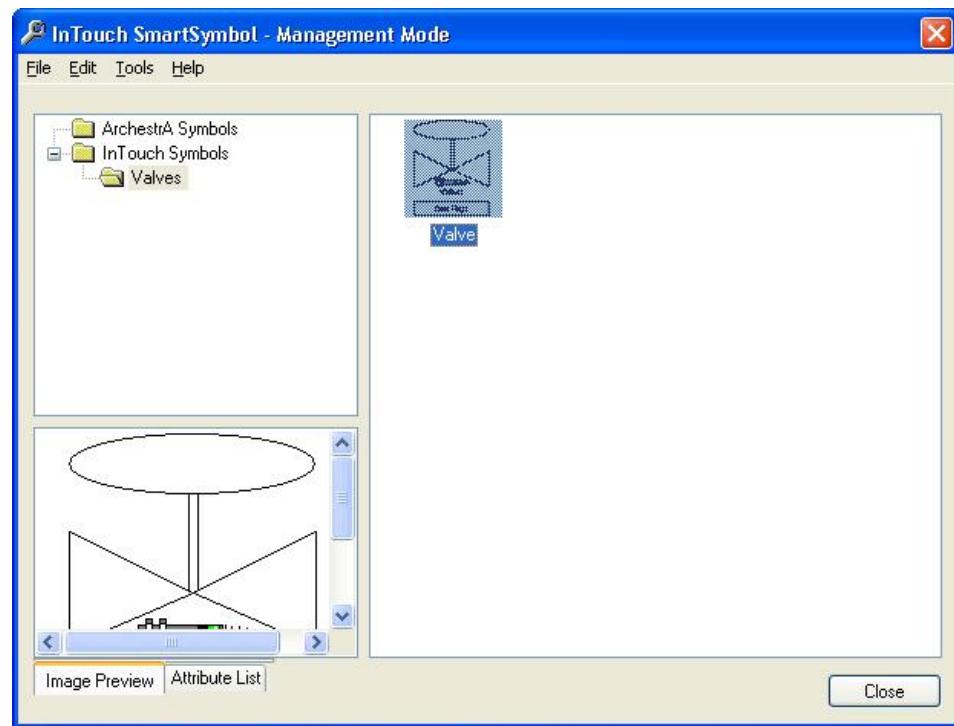


31. Attribute List 탭을 클릭합니다. Valve SmartSymbol을 위한 속성들을 보여줍니다.

SmartSymbol 들 정리하기

32. 그것에 포커스를 주기 위해 InTouch Symbols을 클릭합니다.
(SmartSymbol Manager가 실행되는 동안)

33. File / New Folder를 클릭합니다.
34. Valves 타입을 선택합니다.
35. InTouch Symbols 폴더를 선택합니다.

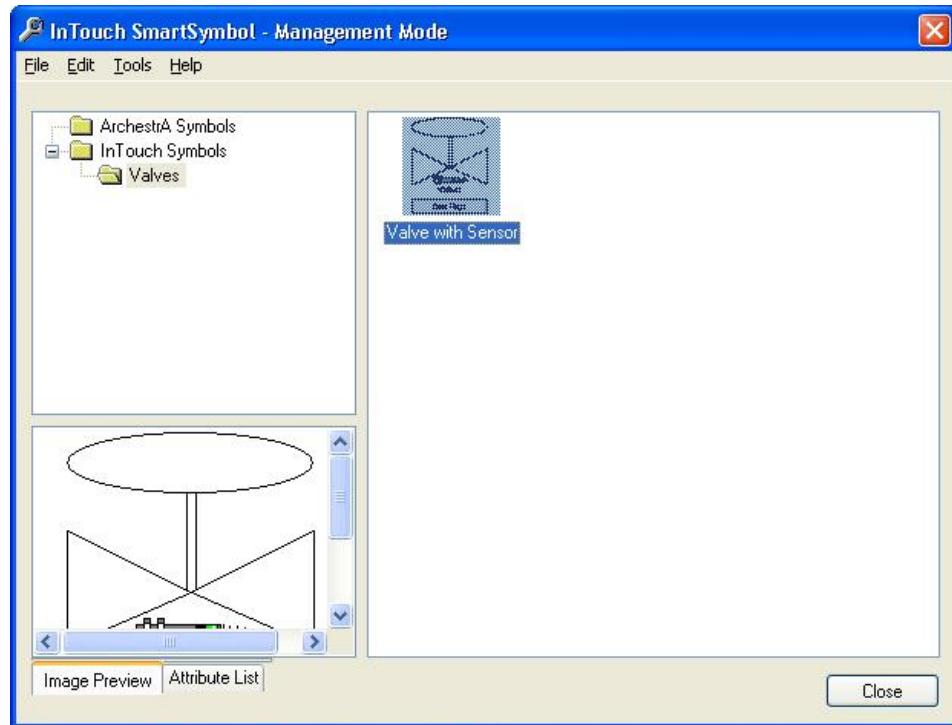


36. Valve 심볼을 새로운 Valves 폴더로 드래그 합니다.

SmartSymbol 재명명, 복사, 삭제, 복구하기

SmartSymbols 재명명

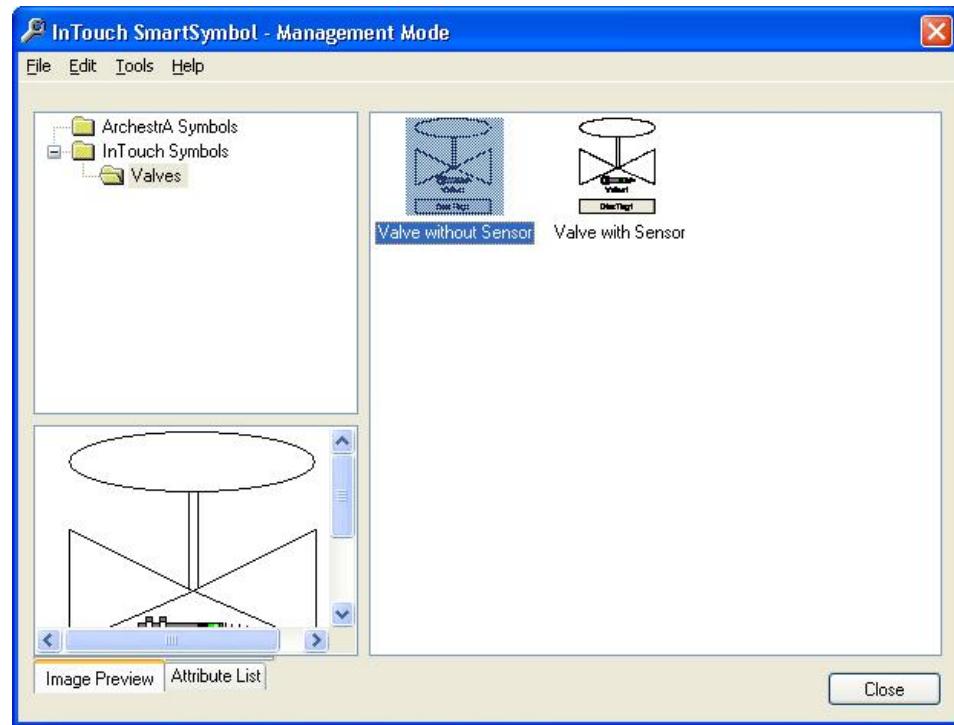
37. 그것에 포커스를 주기 위해 Valve SmartSymbol을 클릭합니다.
38. Edit / Rename 클릭합니다.
39. 다음과 같은 새로운 이름 : Valve with Sensor



40. 엔터를 누르고 valve를 재명명합니다.

SmartSymbol 복사

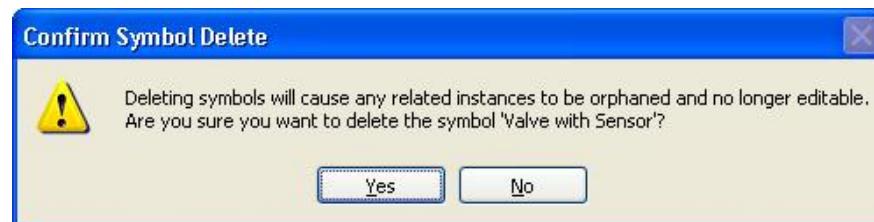
41. 그것에 포커스를 주기 위해 Valve 와Sensor SmartSymbol을 클릭합니다.
42. Edit / Copy를 클릭합니다.
43. Edit / Paste를 클릭합니다.
44. 새로운 valve를 다음과 같이 재명명 합니다. : Valve without Sensor
45. 엔터를 누릅니다.



지금 화면위에 새로운 Valve Without Sensor SmartSymbol 을 수정하고 Sensor.를 제거합니다. Hint: 수정하는 동안 셀 토큰화기를 사용합니다.

SmartSymbol 삭제하기

46. SmartSymbol Manager를 작동합니다.
47. 그것에 포커스를 주기 위해 Valve with Sensor SmartSymbol를 클릭합니다.
48. File / Delete을 클릭합니다.
49. Confirm Symbol Delete 대화상자에서 YES를 클릭합니다.



삭제된 SmartSymbol 복구하기

50. 인터치 어플리케이션에서 Links 윈도우를 엽니다.
51. 그것에 포커스를 주기위해 Valve1그래픽을 클릭합니다.
52. Special / SmartSymbol / Recover SmartSymbol 클릭합니다.
셀로 부터 SmartSymbol 이 생성 될것이며 New Symbol 이라는 이름으로 SmartSymbol -Management Mode 윈도우가 나타날 것입니다.
53. Valve with Sensor처럼 새로운 SmartSymbol을 재명명하고 엔터를 누릅니다.
54. Valve with Sensor를 Valves폴더로 이동합니다.

SmartSymbol 를 내보내기

55. SmartSymbol Manager로 부터 File / Export를 클릭합니다.
56. 내보내진 SmartSymbol을 원하는 이름으로 정하고 선택한 디렉토리에 저장합니다.
SmartSymbol library 는 지금 어떠한 인터치 어플리케이션으로 import 할 수 있습니다.

SmartSymbol 가져오기

57. 인터치 어플리케이션에서 모든 윈도우들을 닫고, 필요하다면 저장합니다.
58. Special / SmartSymbol / Manage SmartSymbols 클릭합니다.
59. File / Import를 클릭합니다.
60. 강사에 의해 제공된 폴더 위치로 이동합니다.
61. Import Preferences 윈도우가 보여지면, Rename imported symbol 체크박스를 클릭합니다.
62. 심볼의 이름을 다음과 같이 채웁니다.: _Instructor
63. OK를 클릭합니다.
64. SmartSymbol Manager안의 가져온 SmartSymbols를 다시 봅니다.

Section 5 – ArchestrA Symbol 소개

IDE-Managed InTouch에서 사용될 수 있는 다양한 객체들이 존재 합니다. 이 장에서는 3 가지 객체를 사용하는 방법에 대해서 살펴봅니다.

- | InTouch Objects
- | InTouch SmartSymbols
- | ArchestrA Symbols

InTouch Application에 InTouch Object와 SmartSymbol을 사용하는 방법을 배워 보았습니다.

추가적으로, ArchestrA Symbol Library를 사용하여, IDE-Managed InTouch Application Window에 ArchestrA Symbol Library를 추가하고, tagname과 연결하는 방법을 배워봅니다.

IDE-Managed Application을 작업할 때, 개발자는 ArchestrA Symbol Library로부터 좀 더 세련된 그래픽을 사용할 수 있습니다. 이 그래픽 Symbol은 InTouch Tagname, 애니메이션, Script 등과 연결될 수 있습니다.

Note: InTouch Application의 Window에서는 ArchestrA Symbol을 사용할 수 없습니다.
ArchestrA Symbol은 IDE로부터 만들어지는 InTouch Application에만 사용할 수 있습니다.

ArchestrA Symbol에 대해

ArchestrA Symbol은 InTouch HMI Application에 비주얼 data에 사용되는 그래픽 심볼입니다.

이것은 IDE에서 관리되며, IDE-Managed InTouch Application에서만 사용될 수 있습니다.

ArchestrA Symbol을 사용하면

- | Managed InTouch Application의 Window에 ArchestrA Symbol을 사용합니다.
- | InTouch에 사용된 ArchestrA Symbol은 크기를 조정하거나, 위치를 바꿀 수 있습니다.
- | ArchestrA Symbol은 사용자 속성을 구성할 수 있습니다.

InTouch Window 에 ArchestrA Symbol 사용하기

이 장에서는 Graphic Toolbar 로부터 ArchestrA Symbol 을 어떻게 사용하는지 설명합니다.

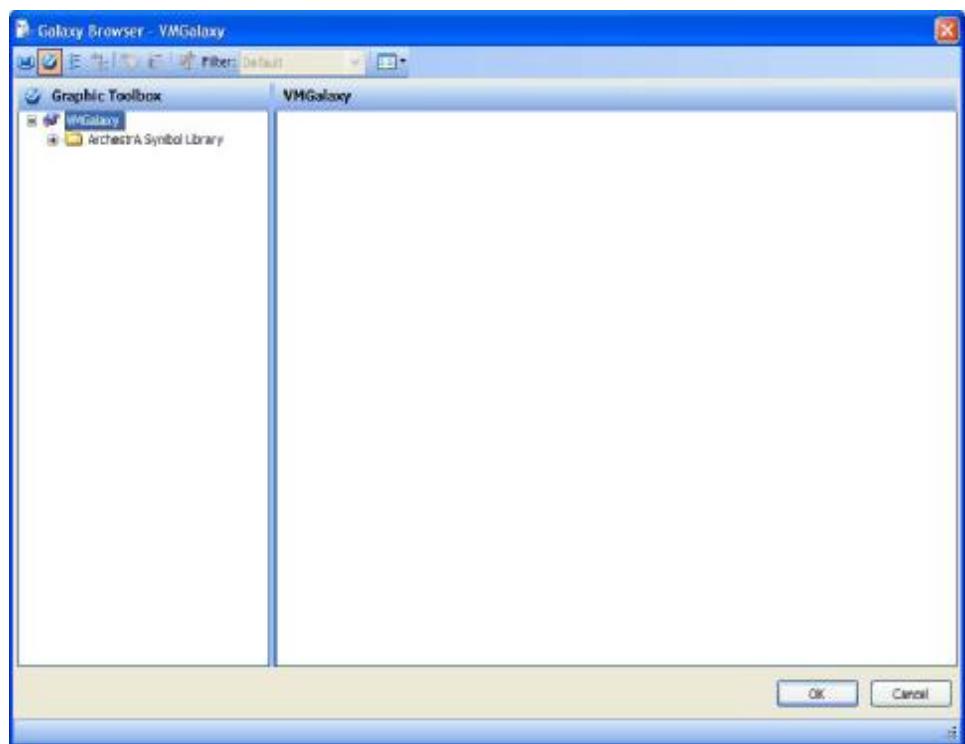
ArchestrA 버튼을 삽입 하기 위해 :

a. WindowMaker 에서 Link 할 Window 를 엽니다.

b. Wizard toolbar 에서 ArchestrA Symbol 버튼  를 클릭합니다.

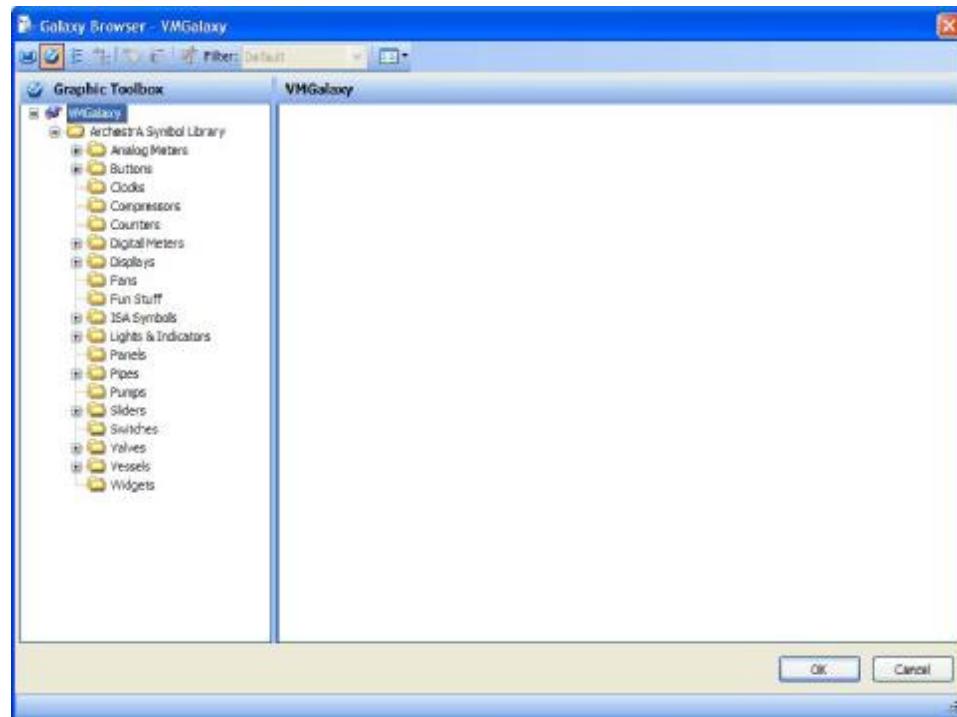
Galaxy Browser 대화상자가 나타납니다.

Graphic Toolbox 에 만들어 놓은 Galaxy 가보이고, 그 밑에 ArchestrA Symbol Library 폴더가 있습니다.

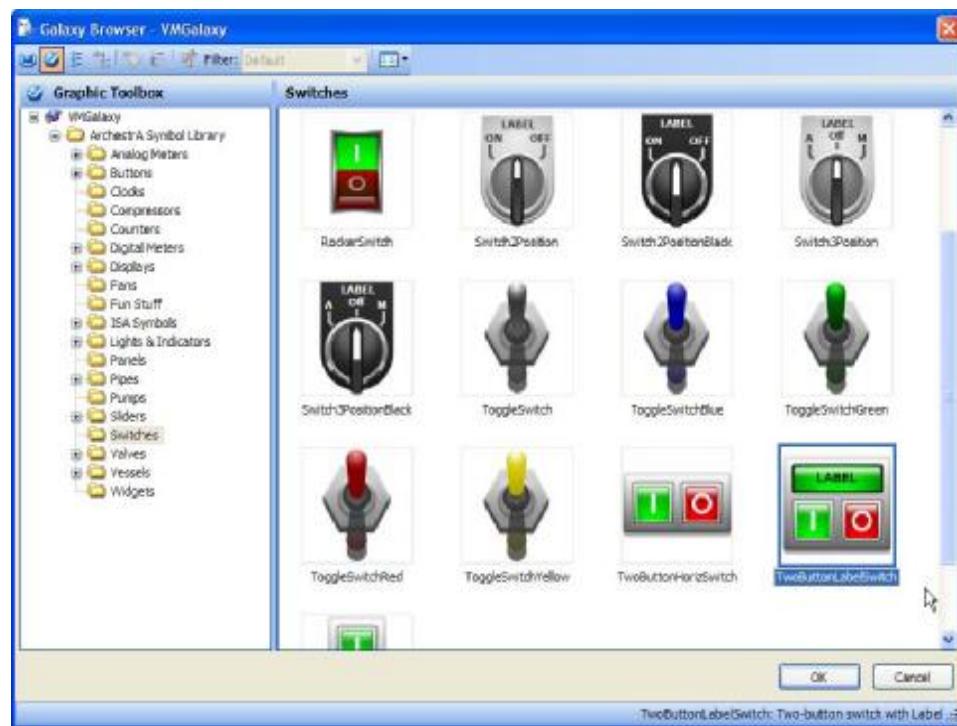


c. ArchestrA Symbol Library 를 확장 하며, Content 들이 표시됩니다.

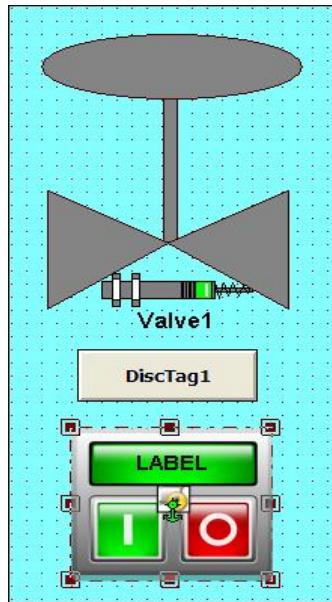
ArchestrA Symbol Library 폴더에는 미리 만들어진 ArchestrA 그래픽 객체들이 포함되어 있습니다.



d. Switches 폴더를 선택합니다.



- e. Switches 항목에 표시되어 있는 다양한 Symbol에서 TwoButtonLabelSwitch 를 선택합니다.
- f. OK 를 클릭합니다.
- g. DiscTag1 버튼 밑에 커서를 클릭하여 Symbol 을 그립니다.



Switch 가 Window 에 Link 됩니다.

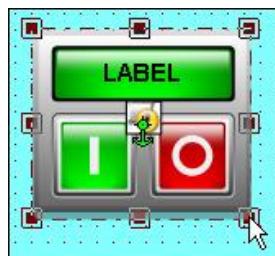
ArchesrA Symbol 크기 조절하기

InTouch Window 에 그려진 ArchesrA Symbol 은 객체의 넓이, 높이 등을 자유로이 조절할 수 있습니다.

추가된 ArchesrA Symbol을 그것의 최소 크기 보다 작게 조절 할 수 없습니다. 최소 크기는 구성요소를 포함하는 펜의 넓이에 의해 결정 됩니다.

ArchesrA Symbol 의 크기조절하기위해 :

- a. ArchesrA Symbol 을 선택하면, 테두리에 작은 사각형이 나타날 것입니다.



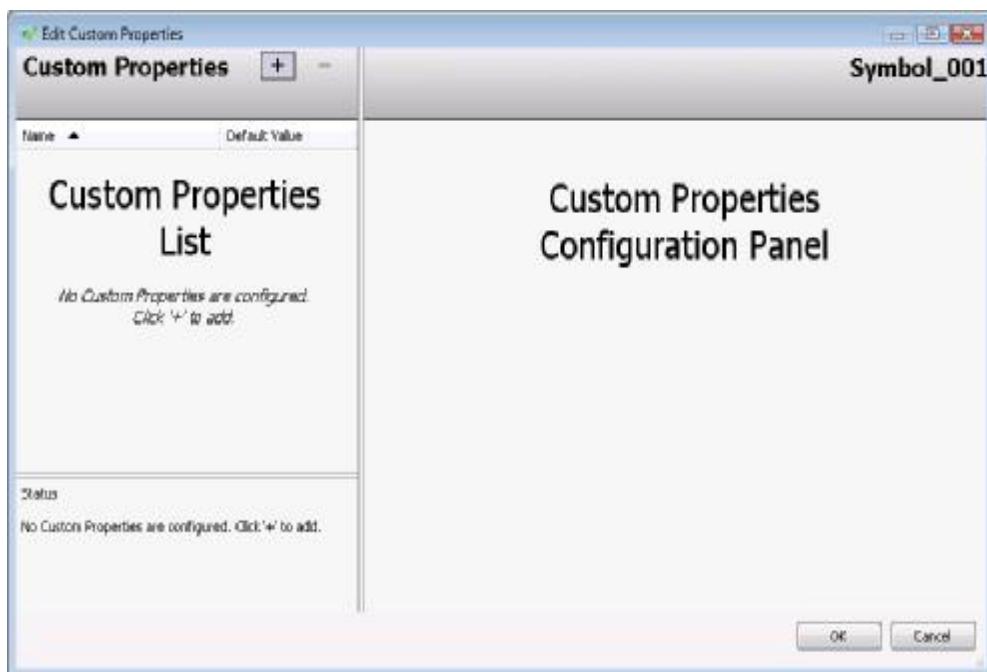
- b. Shift 키를 누른 상태에서 테두리의 사각형을 마우스로 찍고, Drag 함으로써 크기 조절을 할 수 있습니다.

InTouch Tagname을 ArchestrA Symbol에 연결하기

Edit Custom Properties 대화상자를 통해 ArchestrA Symbol을 InTouch의 tagname과 연결하도록 구성 할 수 있습니다.

Edit Custom Properties를 통해서 개발자는

- | ArchestrA Symbol의 Custom properties 대화상자를 볼 수 있습니다.
- | Symbol의 설정 및 동작을 확인할 수 있습니다.
- | InTouch Tagname을 ArchestrA Symbol에 연결할 수 있습니다.



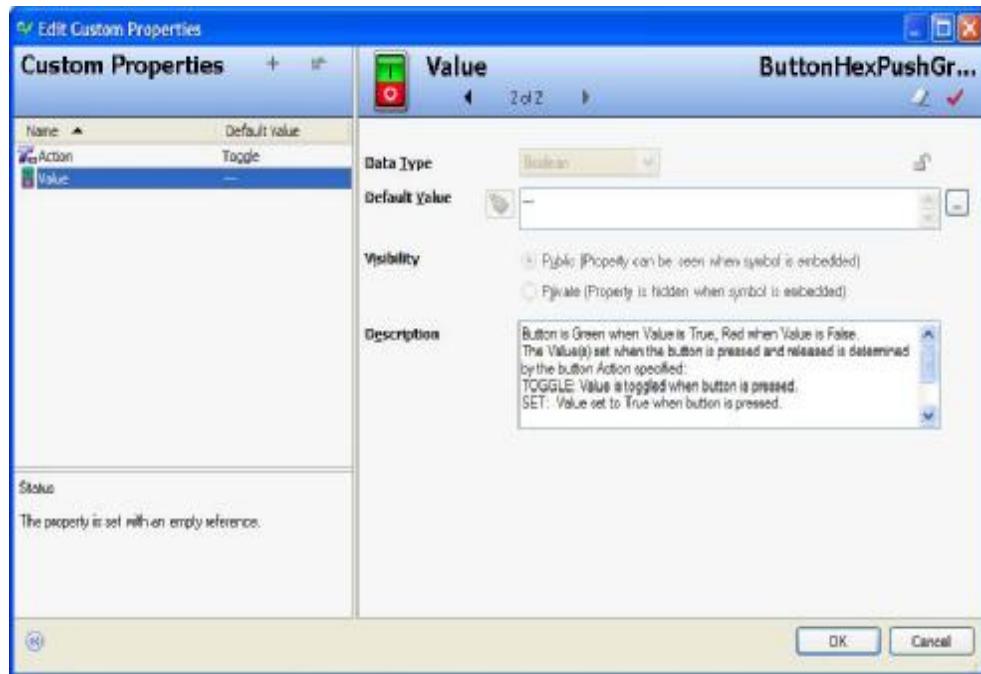
Edit Custom Properties 대화상자는 다음의 내용을 포함하고 있습니다.

- | Custom Properties List : ArchestrA Symbol에 정의 되어있는 속성리스트를 볼 수 있습니다.
- | Custom Properties Configuration Panel은 각 Custom property의 구성을 포함합니다.
- | Configuration Header의 오른쪽에는 Symbol name이 보이고, Custom Property name은 Custom Property List의 활성화된 속성을 나타냅니다.

InTouch Tagname 을 ArchesrA Symbol 에 연결하기 위해서

- InTouch window에서 ArchesrA switch 를 더블 클릭합니다.

Edit Custom Properties 대화상자가 나타납니다.



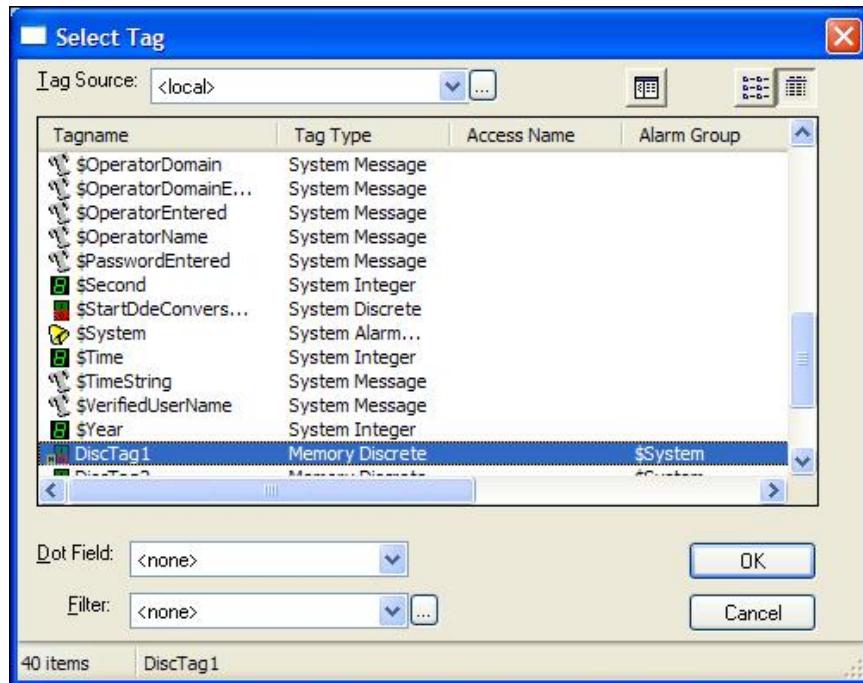
- Custom Properties 리스트에서 Value 를 선택합니다.

Note: Configuration 의 header에는 Symbol name 이 보여집니다. 위의 경우에는 오른쪽에 ButtonHexPushGreenRedLight 가 Symbolname 이며, 왼쪽의 Value 는 Custom Property name 입니다.

Default Value 상자는 Symbol 의 value, reference 또는 expression 을 나타냅니다.

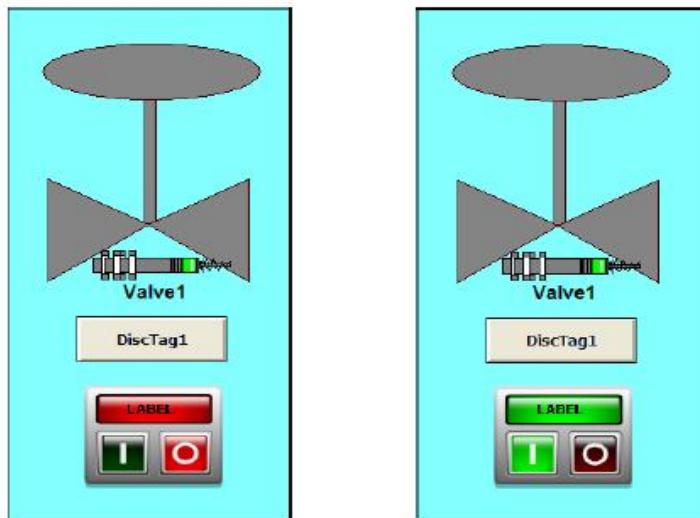
Default Value 상자 안에서 더블클릭을 하면, Tag Browser 창이 나타나고, Tag 를 선택함으로써 Symbol 에 연결할 수 있습니다.

- Default Value 상자에서 더블클릭을 하면, Select Tag 대화상자가 나타납니다.



- d. DiscTag1 을 선택하고, OK를 클릭합니다.

Custom property로 구성된 ArchestrA Symbol의 애니메이션은 이제 InTouch Tagname에 영향을 주게 됩니다.



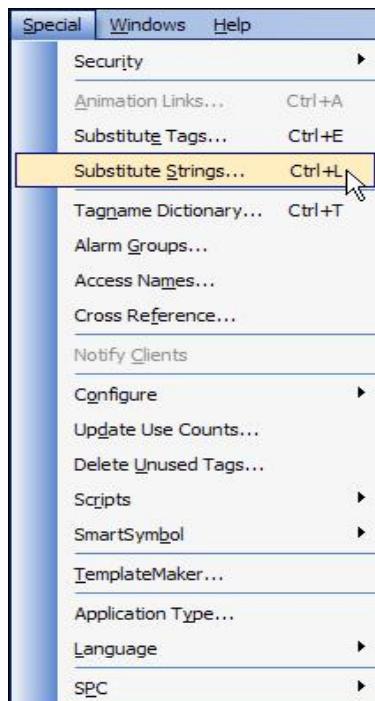
- e. Runtime을 실행시켜, ArchestrA switch를 실행 합니다.
f. 개발환경으로 다시 돌아옵니다.

ArchestrA Symbol 의 문자 교체

개발자는 InTouch 객체의 문자를 치환 할 수 있었던 것처럼, 교체 문자를 ArchestrA Symbol에 대입함으로써 문자 교체를 할 수 있습니다.

ArchestrA Symbol 의 문자 치환

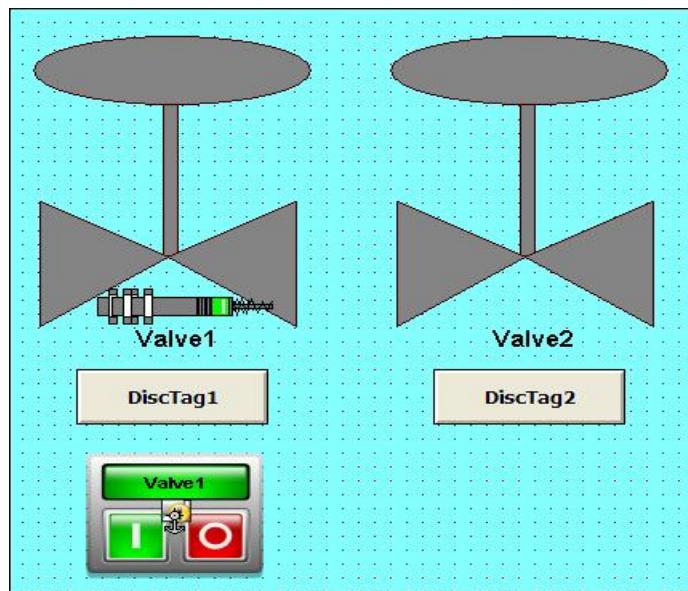
- Switch symbol 을 선택하여 클릭합니다.



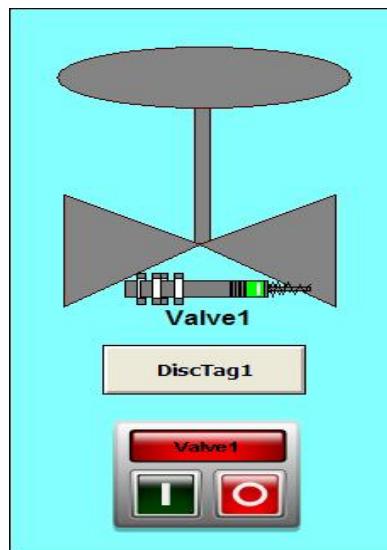
- Special 메뉴에서 Substitute String 을 클릭합니다.



- Substitute Strings 대화상자가 나타납니다. New String 필드에 Valve1 이라고 입력합니다. OK를 클릭합니다.



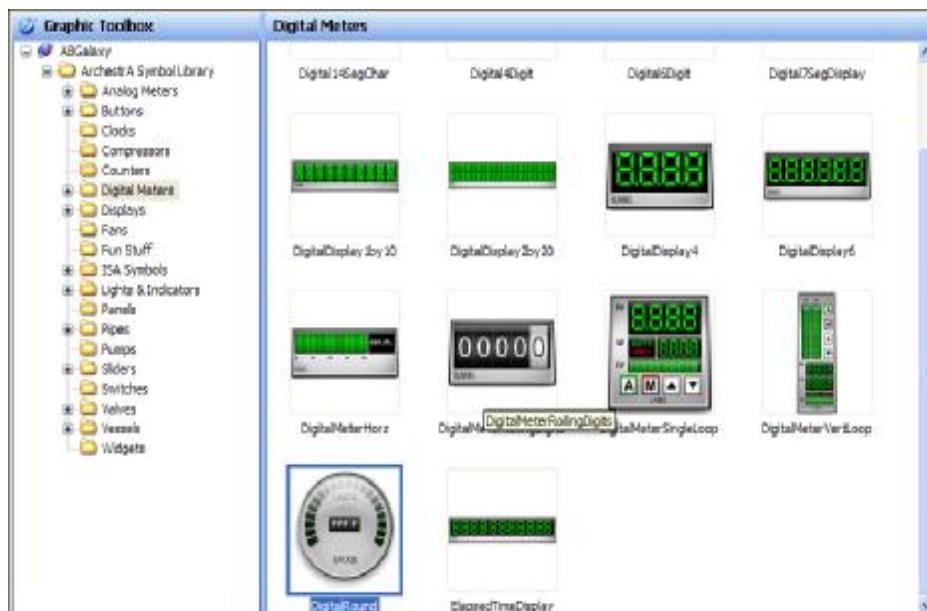
ArchestrA symbol에 입력 된 문자는 새로운 교체 문자로 치환됩니다.



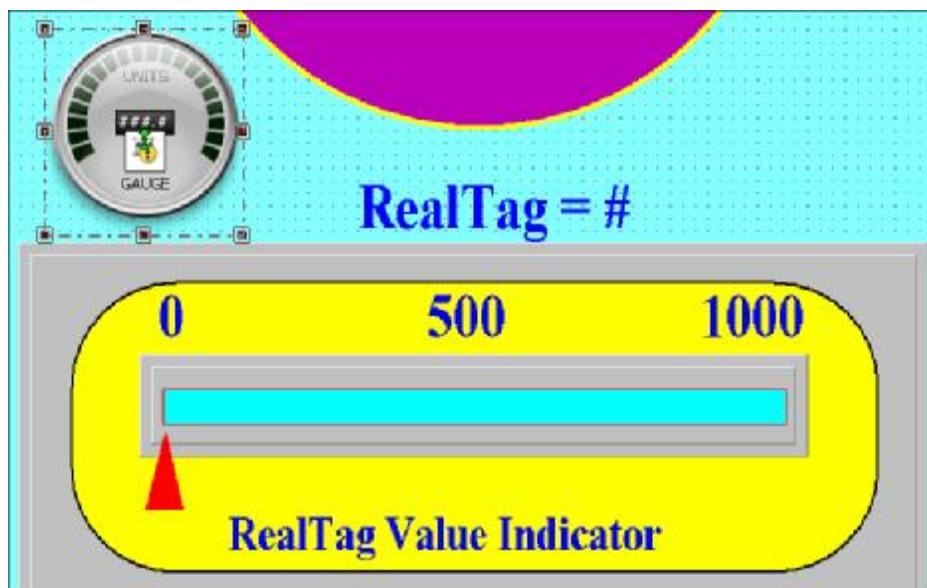
- d. Runtime을 실행합니다.
- e. 개발 환경으로 전환합니다.

Application Window ArchesrA Digital Meter 추가하기

-  버튼을 클릭하여, ArchesrA Symbol Library를 실행시킵니다.
- Digital Meters 폴더를 선택합니다.

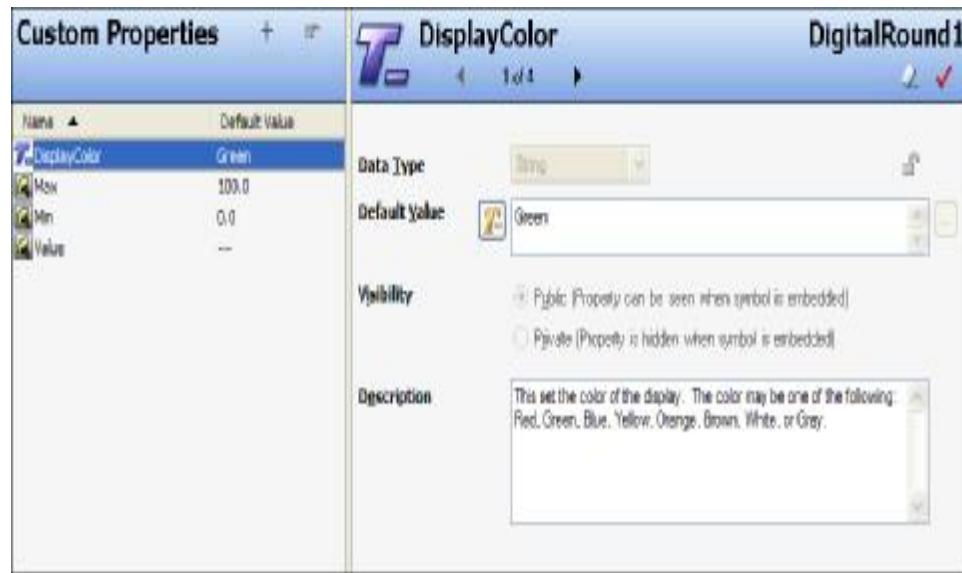


- DigitalRound meter를 더블클릭하고, 윈도우에 그려 넣습니다.

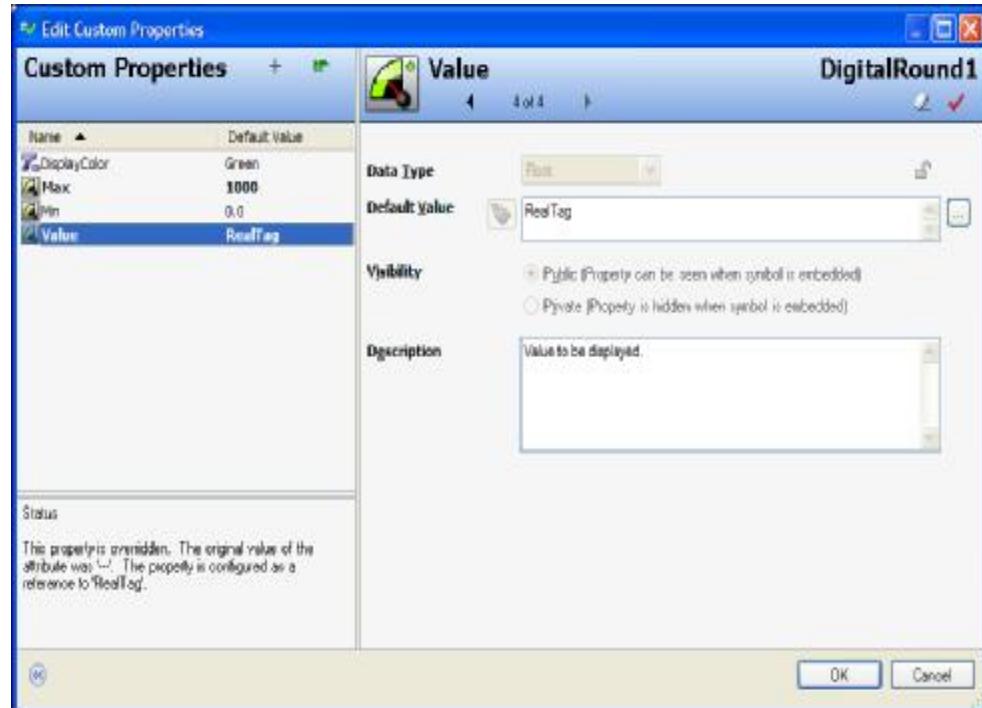


- 슬라이더 윗부분, RealTag라는 텍스트 왼편에  커서를 클릭함으로써, digital meter symbol을 위치시킵니다.

- e. Digital meter symbol을 더블클릭 하며, Edit Custom Properties 대화상자가 나타납니다.

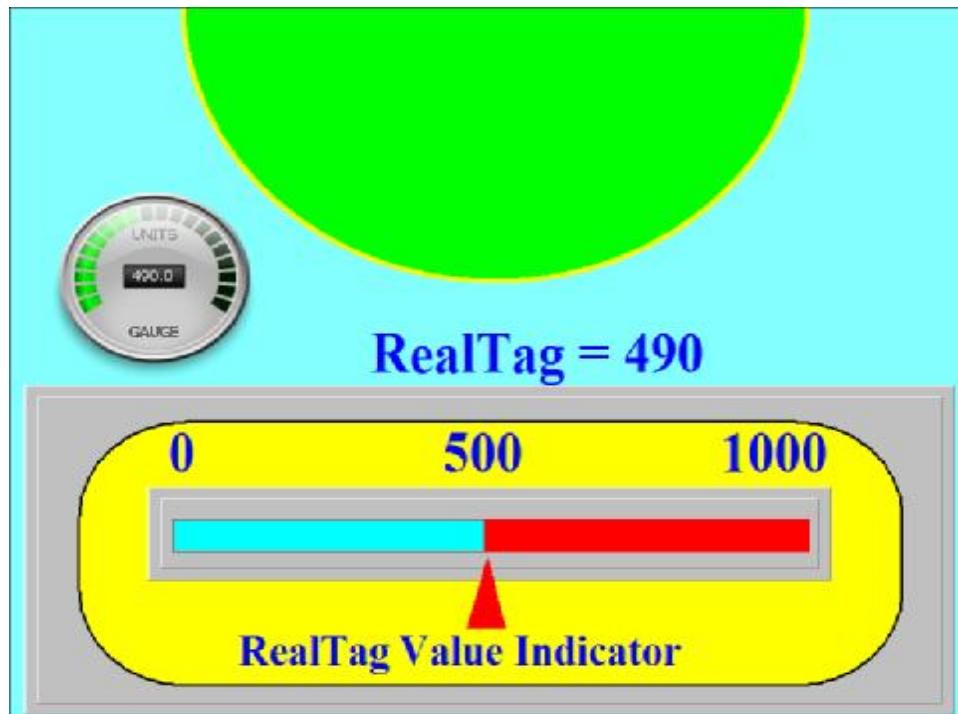


- f. Max 속성의 Default Value에는 1000을, Min 속성의 Default Value에는 0.0을 입력합니다.
 g. Value 속성을 선택하고, Default Value box에서 더블클릭을 합니다.



- h. Select Tag 대화상자가 나타나면, RealTag를 선택하고 OK를 클릭합니다.

- i. Custom Properties 대화상자에서 OK를 클릭하고 닫습니다.



- j. Runtime으로 전환하고 슬라이더의 포인터를 앞뒤로 움직이며, digital meter를 테스트합니다.
RealTag로 링크된 모든 객체들은 RealTag 의 값을 반영해 digital meter 에 나타납니다.
k. 개발 환경으로 전환합니다.

Lab 5 – ArchestrA Symbol 사용하기

소개

이 장에서는 Lab 4에서 사용한 그래픽에 이어서 실습하게 됩니다. 두 개의 ArchestrA Symbol을 추가하고, 그것들을 InTouch Tagname과 연결할 것입니다.

InTouch Window에 ArchestrA Symbol을 어떻게 사용하는지 InTouch Tagname과 어떻게 연결하는지 실습해보도록 합니다.

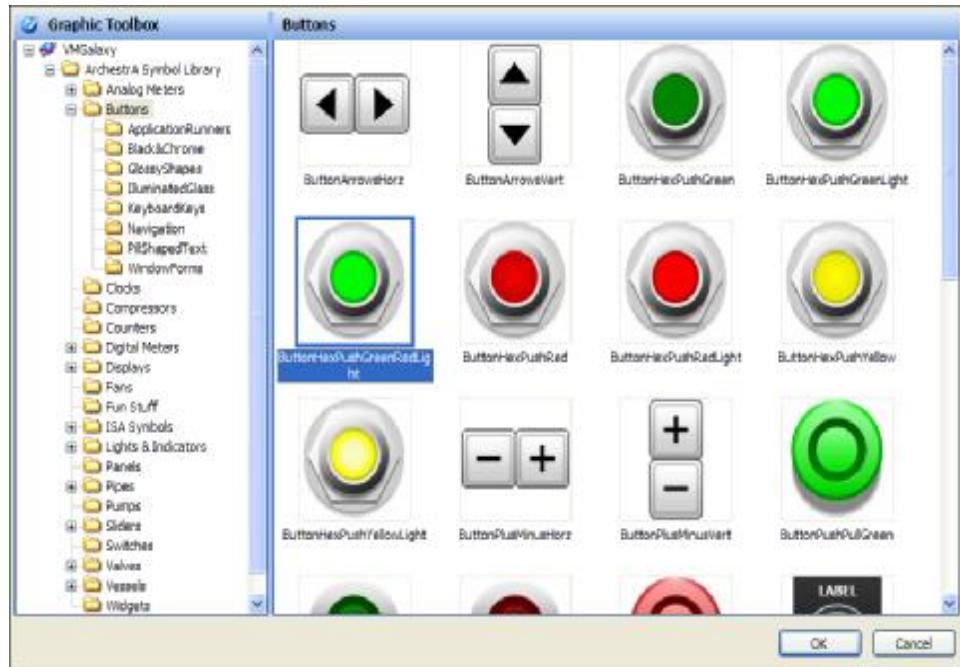
Note: ArchestrA Symbol은 Managed InTouch Application에서만 사용될 수 있으며, InTouch에서는 사용할 수 없습니다.

ArchestrA Symbol 삽입과 InTouch Tagname 연결하기

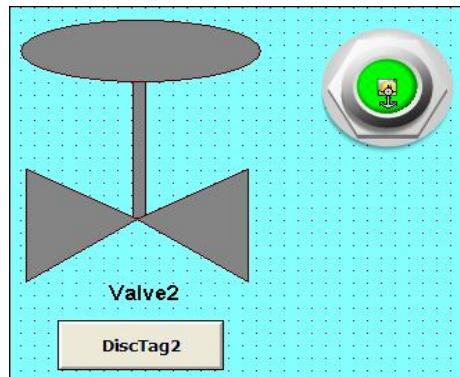
Note: 각 실습 끝에는, Runtime으로 실행시켜 결과를 확인하도록 합니다.

ArchestrA Symbol 삽입

1. Embed ArchestrA Symbol 버튼을 클릭하면, ArchestrA Symbol Library 가 나타납니다.
2. Button 폴더를 클릭합니다.



3. ButtonsHexGreenRedLight 버튼을 선택하고, OK를 클릭합니다.

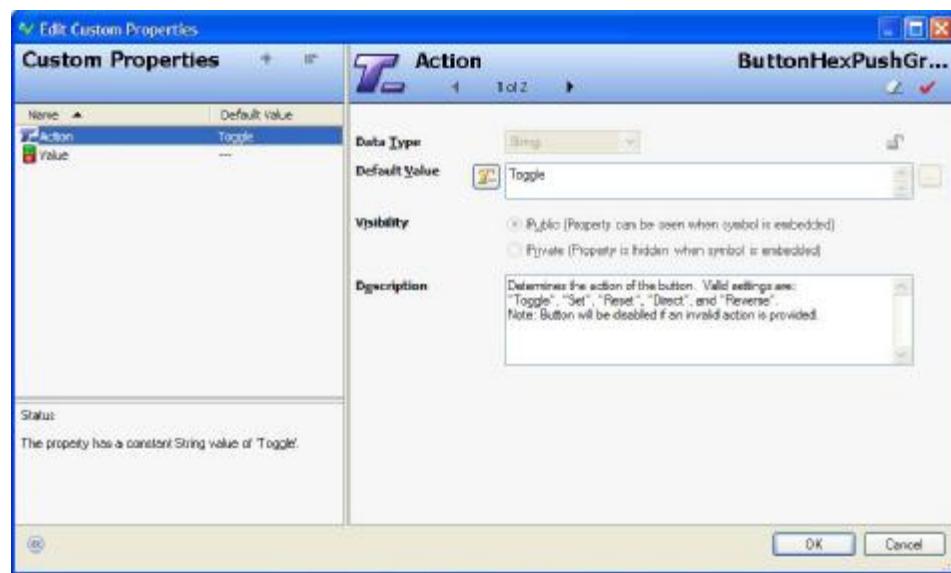


4. Valve2 symbol의 왼쪽에 ArchesrA 버튼을 그려 넣습니다.



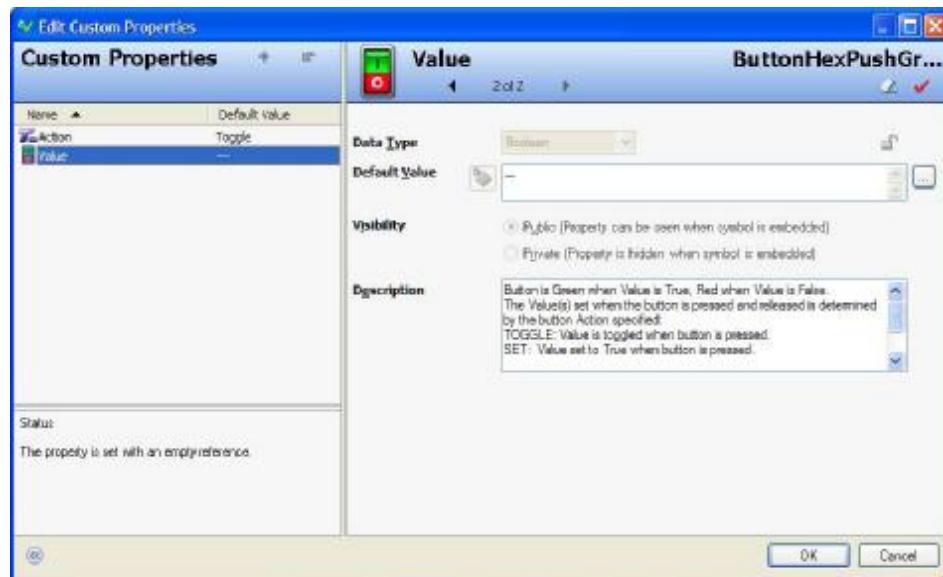
5. 버튼이 너무 크게 그려지면, Shift Key를 누른 상태에서, 버튼객체의 테두리를 마우스로 드래그 해서 크기를 조정합니다.

ArchesrA Symbol의 InTouch Tagname 연결하기



6. ArchesrA Symbol을 더블클릭 합니다.

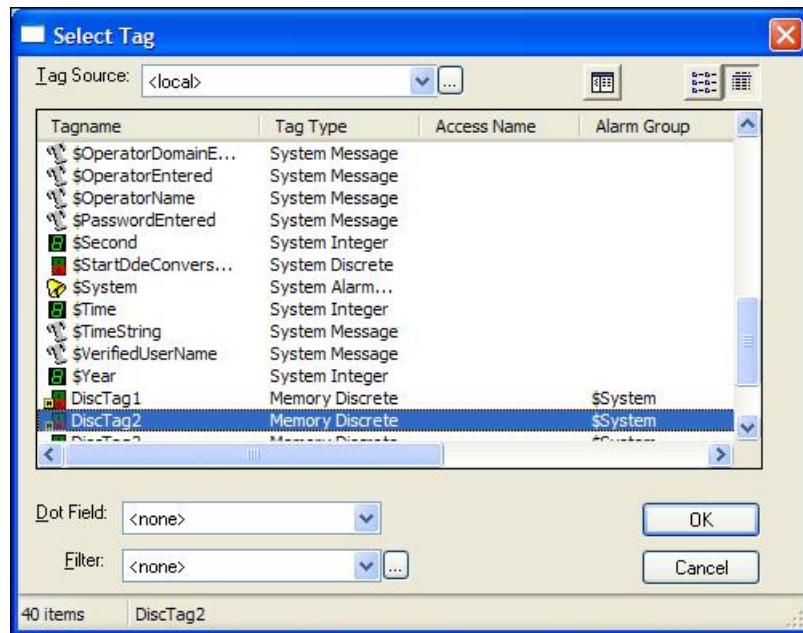
Edit Custom Properties 대화상자가 나타납니다.



7. 왼쪽 패널의 Custom Properties에서, Value를 선택합니다.

선택된 Custom property의 구성이 오른쪽 패널에 나타납니다.

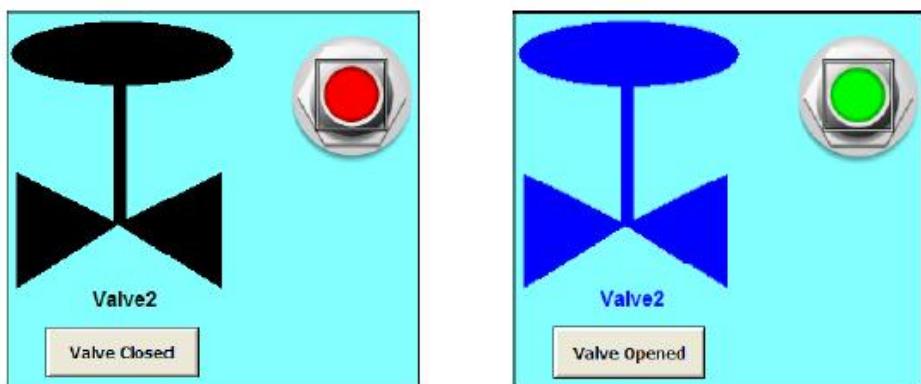
8. Default Value 의 빈칸을 선택합니다.



9. Defult Value 상자를 더블 클릭하면, Select Tag 대화상자가 나타납니다.

10. DiscTag2를 선택하고 OK를 클릭합니다.

11. Edit Custom Property 에서 OK를 한번 더 클릭합니다.

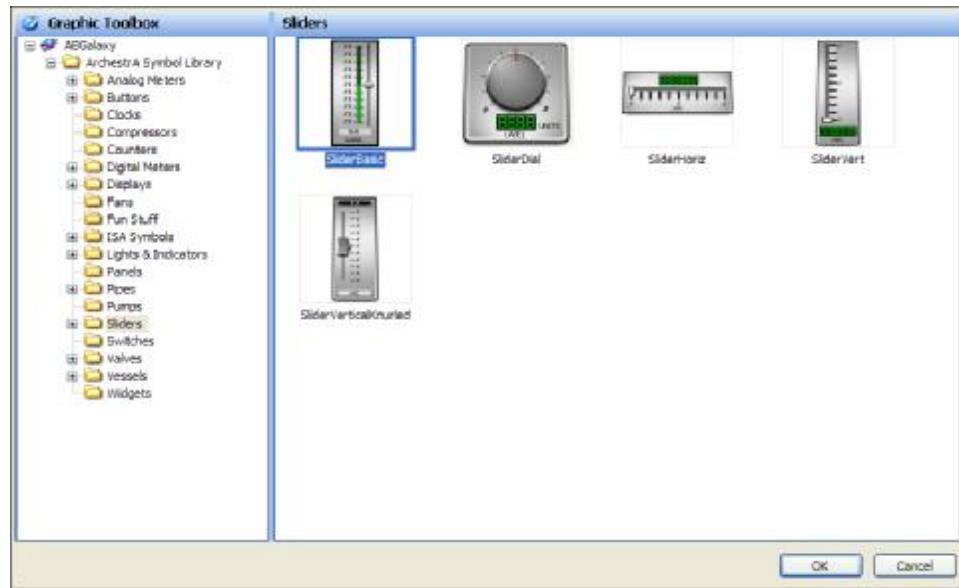


12. InTouch를 실행시키고, ArchestrA button 을 Test 합니다.

13. 개발환경으로 전환합니다.

ArchestrA Slider Test

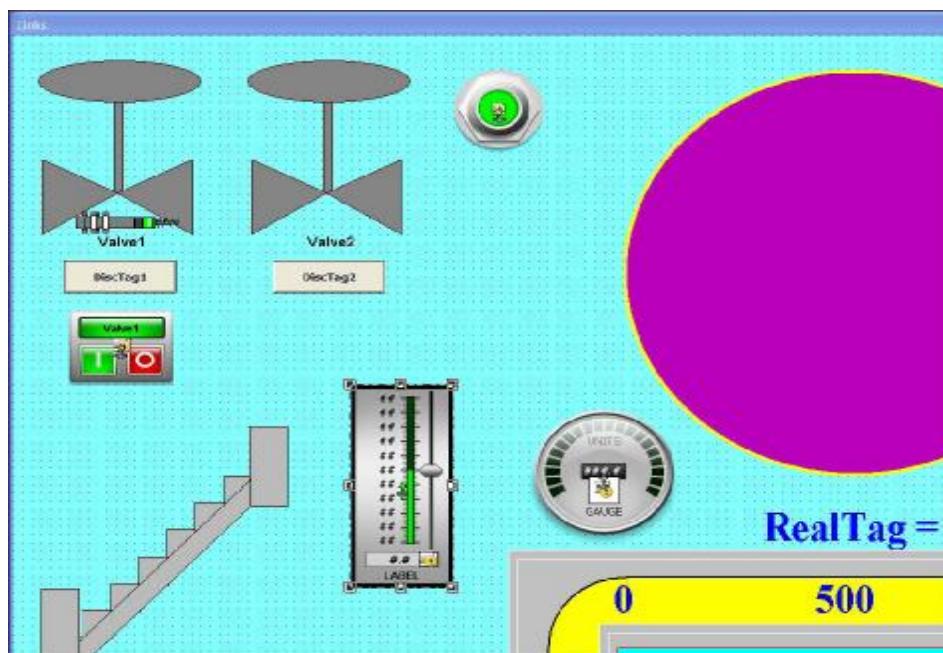
14. ArchestrA Symbol Library  버튼을 클릭합니다.



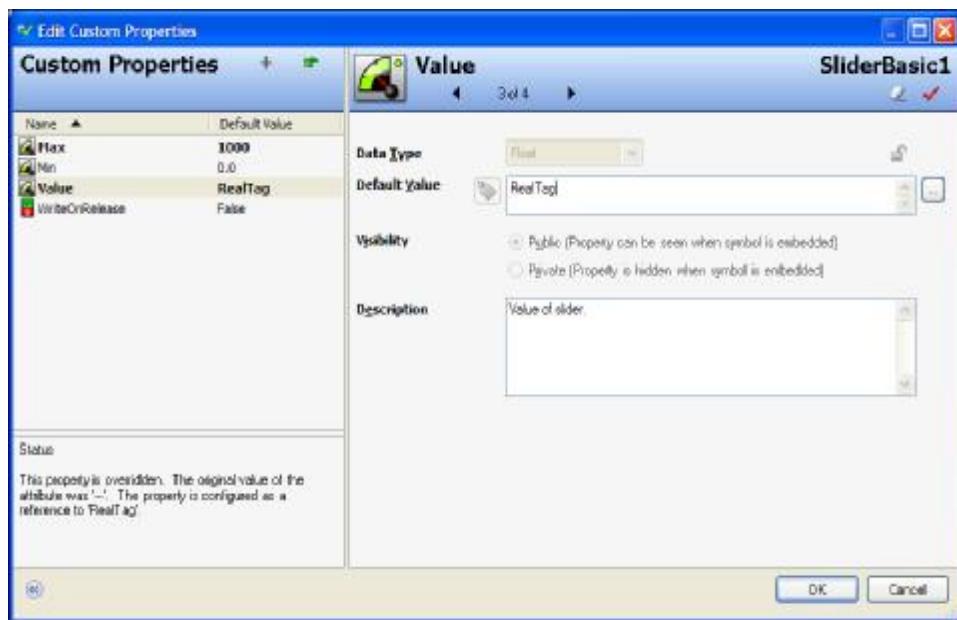
15. Sliders 폴더를 선택합니다.

16. Sliders 항목에서 첫번째 Symbol을 더블클릭하고, InTouch Window로 돌아옵니다.

17. Window 의 digital meter 오른쪽에 Slider를 삽입합니다.



18. Slider Symbol 을 더블 클릭해서, Edit Custom Properties 대화상자를 엽니다.

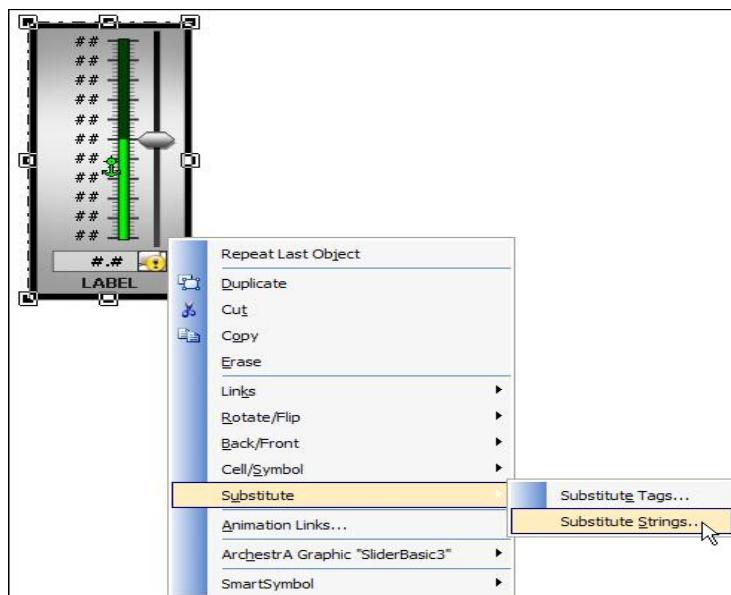


19. 기본 값을 다음과 같이 바꿔 줍니다.

- | Max : 1000
- | Value : RealTag

20. OK를 클릭합니다.

ArchesrA Symbol 의 문자값 바꾸기



21. Slider Symbol에서 오른쪽 마우스 클릭 후, Substitute/ Substitute Strings을 선택합니다.



22. Substitute Strings 대화상자에서, LABEL에 RealTag 라고 입력합니다.

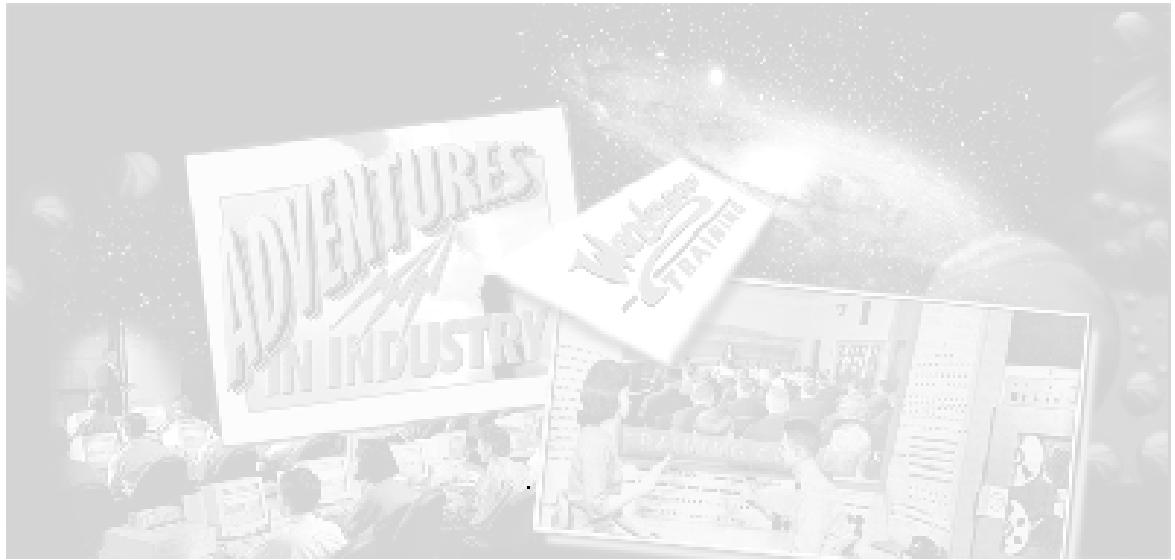
23. OK를 클릭합니다.

ArchestrA Symbol에 입력된 문자가 RealTag로 변환된 것을 볼 수 있습니다.



24. WinodwViewer를 실행시키고 ArchestrA Slider를 움직이며 테스트 합니다.

25. 개발 환경으로 돌아 옵니다.



Module 5

InTouch Quick Scripts

Section 1 – QuickScript Types

Section 2 – Script Functions

Lab 6 – Using Intouch QuickScropts

Section 1 - QuickScript Types

소개

QuickScript는 스크립팅 언어의 이름이며, InTouch 환경에서 사용 가능한 함수입니다. 현재 스크립팅 언어는 IEC611-31 와 구조적 문자와 같은 PLC 언어와 매우 유사합니다.

InTouch QuickScript는 키를 누르거나 윈도우를 열거나 값이 변경되게 하는 등 특별한 기준을 근거로 그 기준을 충족하는 명령과 논리적인 작동을 실행하기 위한 기능을 가지고 있습니다. 모든 InTouch QuickScript는 이벤트에 의해 수행 됩니다. 그런 이벤트은 값의 변화, 상태, 마우스의 클릭, 타이머 등일 수 있습니다.

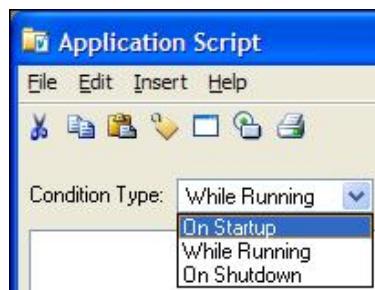
프로세싱의 순서가 명확한 Application입니다. 반면에, 같은 이벤트에 의해서 시작된 여러 개의 script들은 수행되는 방법에 있어서, 어떤 고유한 순서가 있는 것처럼 보이지만, 그런 특별한 순서를 보장 하지는 않습니다. 그러므로, 프로세싱의 순서에 의존해서는 안됩니다.

Script Type	Description
Application	모든 Application과 연관된 script.
Window	어떤 특정한 윈도우와 연관된 script.
Key	어떤 특정한 키나 키보드에서 복합적인 키와 연관된 script.
Touch Action	Touch Link – Touch Pushbutton – Action 이라는 Animation link의 객체와 연관된 script.
Data Change	단지 tagname 또는 tagname.field와 연관된 script.
Condition	Discrete tagname이나 식에 연관된 script.
ActiveX Event	Runtime 시, ActiveX control 이벤트 수행에 연관된 script.
QuickFunctions	다른 script나 Animation link의 식(expression)에서 호출하여 사용 할 수 있는 script입니다. QuickFunctions은 다른 모든 type의 script가 단지 동기적인 것에 반해 동기적, 비동기적 일 수 있습니다.

Application Scripts

Application Scripts는 전체 Application에 연결되어있으며, 다른 Application을 작동시키고, 프로세스 시뮬레이션을 만들고, 변수들을 계산하는데 사용 될 수 있습니다.

Application Scripts에는 세 가지 유형이 있습니다.



Condition Type: On Startup

- I **On Startup** : Application이 처음 시작 될 때 한 번 수행 됩니다. 어떠한 원도우가 열리거나 Runtime이 초기화 되기 전에 실행됩니다. On Startup script안에서는 ActiveX 의 방법, 속성 또는 이벤트를 참조 할 수 없습니다. 또한, I/O 통신들은 Application의 On Startup script 실행 후에 초기화됩니다. I/O-type 태그 또는 원격 태그들은 On Startup script안에서는 참조 될 수 없습니다.

Condition Type: While Running Every 0 Msec

- I **While Running** : Application이 실행되고 있는 동안 특별한 시간 간격으로 계속 수행 됩니다. script가 선택되어 실행되는 동안 Every Msec 필드가 활성화 되며 그 필드에는 script가 실행되기 전에 반드시 경과하는 밀리세컨드(millisecond)의 숫자를 입력해야 합니다. script는 특별히 설정된 빈도만큼 반복적으로 실행됩니다. script를 즉시 실행하기 위해서는 동일한 On Startup script를 생성해야 합니다. While Running script를 위한 이벤트나 조건을 만날 때마다, 그 script는 특정한 시간의 간격으로 계속 수행 될 것입니다.

Condition Type: On Shutdown

- I **On Shutdown** : Application이 실행을 끝낼 때, 한 번 수행 됩니다. I/O-type의 태그와 원격 tagname의 참조는 On Shutdown script에서 업데이트 되지 않습니다. 또한, On Shutdown script는 다른 Application을 시작하기 위해서 사용 될 수 없습니다.

Window Scripts

특정한 윈도우안에 있는 윈도우 script 함수



윈도우에 적용될 수 있는 세가지 type 의 script 가 있습니다.

Condition Type: **On Show**

I On Show : 윈도우가 처음 보여질 때, 최초에 한 번 수행됩니다.

Condition Type: **While Showing** Every **0 Msec**

I While Showing : 윈도우가 보여 지는 동안, 특별한 시간 간격 동안 계속해서 수행됩니다. **While Showing**이 선택 되었을 때, **Every Msec** 필드가 활성화 되며 그 필드에는 script가 실행되기 전에 반드시 경과하는 밀리세컨드(millisecond)의 숫자를 입력해야 합니다. script는 특별히 설정된 빈도만큼 반복적으로 실행됩니다. script를 즉시 실행하기 위해, 동일한 **On Show** script를 생성합니다. **While Showing** script를 위한 이벤트나 조건을 만날 때마다, 그 script는 특정한 시간의 간격으로 계속 수행 될 것입니다.

Condition Type: **On Hide**

I On Hide : 윈도우가 숨겨질 때, 한 번 수행 됩니다

Key Scripts

Key Scripts 는 어떤 특정한 버튼 또는 키보드의 복합적인 키들과 연관된 script 입니다.

그것들은 Application 을 위한 전역 키를 만드는데 사용 되어 질 수 있습니다. 예를 들면, 메인 메뉴 윈도우로 돌아오기, 운전자의 로그오프 등등이 있을 수 있습니다.



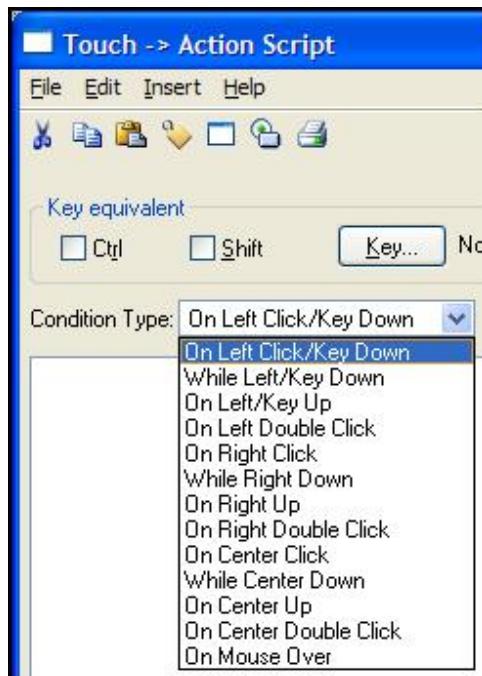
어떤 키에 적용 시킬 수 있는 Key script에는 3 가지 type이 있습니다.

- | **On Key Down** : 키를 처음 눌렀을 때 한 번 수행 합니다
- | **While Down** : 키를 누르고 있는 동안 계속해서 수행 합니다. **While Down**이 선택 된 경우, **Every Msec** 필드가 활성화 되며, 그 필드에는 script가 실행되기 전에 반드시 경과되어야 하는 밀리세컨드(milliseconds)의 숫자를 입력해야 합니다. script는 특별히 설정된 빈도만큼 반복적으로 실행됩니다.
- | **On Key Up** : 키를 눌렸다가 띄어질 때, 한 번 수행됩니다.

Note : 같은 키를 사용하는 script의 경우, 현재 실행중인 윈도우의 Touch Pushbutton - Action scripts에서 설정된 script는 같은 키를 사용하는 Key Script에서 설정된 script 보다 우선으로 사용이 된다.

Touch Action Scripts

Touch Pushbutton의 Action Scripts는 **Touch Link - Action Pushbutton**과 연결 되어있는 객체와 관련 되어 있다는 것을 제외 하고는 **Key Scripts**와 유사합니다. (script 에디터는 animation link selection 대화 상자를 통해서 접근 할 수 있습니다.) 이 Action Scripts는 operator가 객체나 버튼을 클릭 할 때 실행됩니다.



Touch Action Scripts에는 13 가지의 type을 가지고 있습니다.

키나 버튼이 최초로 클릭 되었을 경우, 아래의 Touch Action script들이 한 번 실행됩니다.

- | **I On Left Click/Key Down**
- | **I On Right Click**
- | **I On Center Click**

다음의 Touch Action script들은 키나 버튼이 클릭상태를 유지하는 동안 정해진 빈도로 반복적으로 실행합니다. 이것이 선택되면, **Every Msec** 필드가 활성화됩니다. 그 필드에는 script가 실행 되기전, 반드시 경과되어야 하는 milisecond의 숫자를 입력해야 합니다.

- | **I While Left/Key Down**
- | **I While Right Down**
- | **I While Center Down**

다음의 Touch Action script들은 키나 버튼이 클릭을 하고 난 뒤, 한 번 수행됩니다.

- | **I On Left/Key Up**
- | **I On Right Up**
- | **I On Center Up**

다음의 Touch Action script들은 마우스 버튼이 더블 클릭될 때 실행됩니다. 더블 클릭을 과동일한 대응 키가 없기 때문에 대응키가 선택되었을 때 비활성화 됩니다.

- | **I On Left Double Click**
- | **I On Right Double Click**

I On Center Double Click

다음의 Touch Action script 는 커서가 객체 위에 재설정 될 때 한번 실행됩니다. 스크립트는 After X msec 필드와 함께 지연됩니다. After X mse 는 script 실행전에 객체 위에 있어야 되는 커서의 길이입니다. 대응키 특징은 객체위로 마우스 커서를 움직이게 하는 대응키가 없기 때문에 조건 type 이 선택될때 비활성화됩니다.

I On Mouse Over

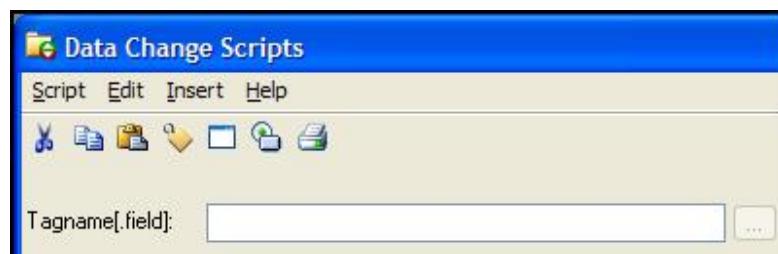
Note : 만약 활성화 된 윈도우의 어떤 객체나 Action pushbutton 이 key script 를 위해 사용되는 키에 동시에 할당 되었다면, 활성화 된 윈도우의 키에 key equivalent link 는 Key script 의 실행보다 우선 되어 질 것 입니다.

Assigning a Key Equivalent to a Script

Key script 에디터는 다른 QuickScript 의 에디터와는 다릅니다. 키에 적용되는 script 를 생성할 때, script 를 실행하기 위해 눌러야 하는 키를 특별히 정해야 합니다. script 에 할당된 대응키는 Module 4 Section 2, “Animating Objects.” 에서 다룹니다.

Data Change Scripts

Data Change Scripts 는 단지 tagname 과 tagname.field 에만 연결 되어 있습니다.

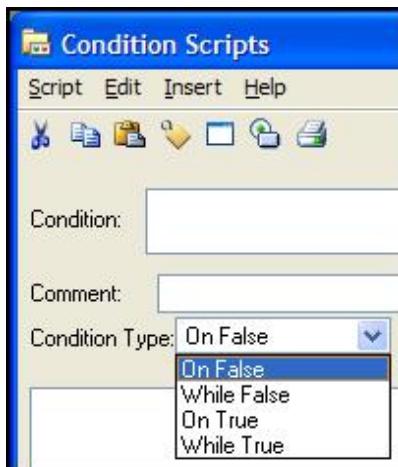


Tagname 또는 tagname.field 의 값이 Tagname Dictionary 에 정의 되어 있는 deadband 보다 더 큰 값의 변화가 있을 때 한번 수행 됩니다.

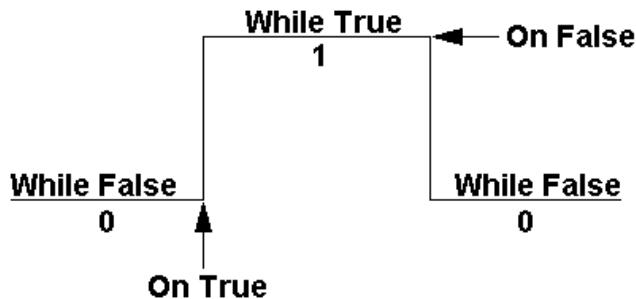
Data Change Script 는 데이터 값의 변화에 따라 수행되기 때문에, Tagname[field] 필드 안에 tagname 또는 tagname.field 를 반드시 명시 해야 합니다

Condition Scripts

Condition Scripts 는 Discrete tagname 또는 true/false 를 나타내는 식과 연결되어 있습니다. 이 script 는 가장 많이 사용되는 script 입니다. Analog tagname 을 포함하고 있는 Discrete 식 역시 사용 되어 질 수 있습니다. 조건은 참 또는 거짓에 상관없이 1024 개의 문자를 포함 할 수 있습니다.



Condition Scripts 는 만나는 조건에 근거해서 수행 되어 지기 때문에, **Condition** 입력란에 특별한 조건(Discrete tagname 혹은 식)을 반드시 명시 해야 합니다. 식의 경우, true 또는 false 를 표현하는 tag 같이 true 또는 false 를 결과 값으로 갖는 조건을 가져야 합니다.



예:

TankLevel >= 75.

tagname **TankLevel** 의 값이 75 보다 크거나 같을 경우, script 가 수행 될 것입니다.

Note : 조건식의 값은 script 가 실행되기 전에 참 또는 거짓이 되기 위해서, 반드시 변환 되어야 합니다. 예로, WindowViewer 시작시 초기값이 참일 경우, 값은 반드시 거짓이 되어야 하고 On True script 를 실행하기 위해 다시 참으로 변환 되어야 합니다.

Condition에 적용 되어 질 수 있는 4 가지 script type이 있습니다.

- | **On False** : 조건이 false로 바뀔 경우, 한 번 수행됩니다.
- | **While False** : 조건이 false인 동안, 계속해서 수행 됩니다. 실행 빈도는 **Every Msec** 필드의 값에 근거합니다.
- | **On True** : 조건이 true로 바뀔 경우, 한번 수행 됩니다.
- | **While True** : 조건이 true인 동안, 계속해서 수행 됩니다. 실행 빈도는 **Every Msec** 필드의 값에 근거합니다.

While True 와 **While False**, 이 두개의 script type은 특별히 정해진 milisecond의 시간 만큼
경과 한 후 실행된다. 즉시 실행하기 위해서는 **On True** 와 **On False**를 중복 생성합니다.

4 가지 script type 모두 같은 조건에 적용될 수 있습니다.

ActiveX Event Scripts

대부분의 ActiveX control은 그것들과 연관된 이벤트를 가지고 있습니다. 예를 들면, 클릭, 더블-클릭, 마우스 다운 휠 클릭 그리고 키가 눌려지는 이벤트들은 ActiveX control 안에서 많이 사용 되어지는 이벤트들입니다. InTouch AxtiveX Event scripts는 이벤트 액션을 지원하기 위해 제공 됩니다. 한 개의 ActiveX Event script는 각각의 이벤트에 연결되어 질 수 있으며, ActiveX control 이벤트는 Runtime(WindowViewer)시에 수행 됩니다.

QuickFunctions

QuickFunctions은 다른 script나 식으로부터 불려 질 수 있는 script입니다.

이 코드는 한 장소에 저장되며, script의 instance들이 어떤 편집 동안에 수정되는 것을 허용합니다.

QuickFunctions의 재사용은 script에 복사하거나 붙여넣기를 하는 복사코드의 양을 줄임으로써 Application의 유지보수를 감소 시킬 수 있을 것 입니다. 재사용된 코드는 한 script 안에 위치하며, 한 번의 수정으로 인해서 모든 script를 갱신 할 수 있습니다.

QuickFunctions은 실행될 때 비동기적으로 정의 되기 때문에, main WindowViewer(런타임) process의 환경에서 실행 될 것입니다. 이러한 기능은 WindowViewer가 main program으로부터 자원을 소모하는 프로세스(SQL database 호출과 같은)을 세분화하는 것을 허용합니다.

자원을 소모하는 프로세스가 수행 될 경우, 비동기적인 QuickFunction은 프로세스가 실행되는 동안, 모든 animation link들과 활성화 되어 있는 다른 InTouch 기능들에 효율적인 방법을 제공합니다.

Animation link 표현식에는 1023 문자까지 입력하도록 제한하고 있습니다. 그러나 좀 더 복잡한 QuickFunction을 생성하고, 이를 Animation link 표현식에서 호출할 수 있습니다.

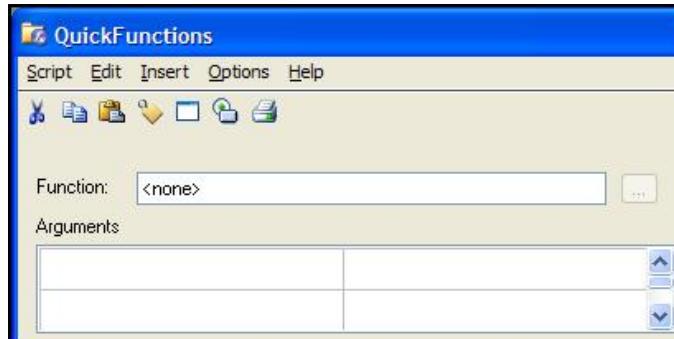
개발자는 표현식에서, 결과 값을 돌려주기 위한 **RETURN** 명령 문자를 포함하고 있는 script를 호출하기 위해서, **CALL**이라는 명령 문자를 사용 할 수 있습니다.

예를 들어, 각각 30 개의 문자를 갖는 여러 tagname 이 추가될 때 (“&” 또는 “:”를 사용해서 추가), 표현식에서 오직 8 개의 tagname(공백도 포함) 만 사용할 수 있습니다. 그러나, 표현식에서 **CALL MYSCRIPT(\$Second)**이라는 문장을 사용함으로써, 30 개의 문자를 갖는 태그명 100 개 이상을 포함하는 QuickFunction을 사용할 수 있습니다. 이러한 QuickFunction은 표현식으로 값을 전달하기 위해서 **RETURN**이라는 명령 문자를 사용합니다.

Note : "trigger" tagname은 Animation Link를 갱신하기 위해 QuickFunction에서 parameter로써 사용되어야 합니다. 예로, QuickFunction의 parameter로써 \$Second를 사용하게 되면, 매 초 변경되는 \$Second의 값을 산출해서 Animation의 표현식으로 전달 됩니다.

QuickFunction의 결과 값은 매 초 호출 됩니다.

일단 QuickFunction을 생성하고 저장 하게 되면, 어떠한 다른 script나 표현식을 그 이름으로 즉시 호출할 수 있습니다.



이름은 31 개의 문자길이로 구성할 수 있으며, 공백과 중복된 이름은 사용할 수 없습니다. 이것은 다른 QuickScripts나 QuickFunction을 호출하기 위해 사용될 표현식의 이름입니다. 이 이름은 또한 QuickScript 에디터에서 **All** 또는 **Quick** 버튼을 클릭 하게 되면 **Choose function** 대화상자에 나타나게 합니다.

Arguments 필드는 QuickScript의 각 argument 이름들을 포함하고 있습니다.

다음은 유효한 데이터 type입니다:

Data Type	Description
Integer	Used to pass integer variable, tagname, or constant values
Real	Used to pass real variable, tagname, or constant values
Discrete	Used to pass discrete variable, tagname, or constant values
Message	Used to pass string variable, tagname, or constant values up to 131 characters

Argument 이름에 사용될수 없는 키워드 : **Return, Call, Dim, As, Integer, Real, Discrete, Message.**

Argument 이름은 Argument 가 정의된 QuickFunction 안에 존재하는 지역변수입니다.
QuickFunction 당 16 개의 argument 까지 사용할 수 있습니다.

arguments 이름은 31 개의 문자 길이로 구성할 수 있으며, 공백은 사용할 수 없습니다.
arguments 이름은 반드시 알파벳 문자로 시작되어야 하며, 중복된 이름은 사용할 수 없습니다.

arguments 이름에 tagname 을 사용할 수는 없습니다. Tagname 은 같은 이름의 QuickScript 의 argument 보다 먼저 실행이 되며, QuickScript 자체가 적절히 실행되지 않을 수 있습니다.

Argument 이름은 지역 변수처럼 취급되기 때문에, 태그개수로 사용되지 않습니다

QuickFunction Arguments

Script parameter 들은 값에 의해 전달됩니다. Argument 표현식은 정수, 실수, Discrete 값 또는 Message type 의 값을 리턴받는 어떠한 script 표현식이 될 수 있습니다. 모든 Argument 표현식 값은 QuickFunction 을 실행하기 전에 script 를 호출하는 로직에 의해 결정됩니다.

예:

CALL Stuff (5.6, 237, "PI");

이 예제에서, 실수상수 5.6 은 argument 1 으로 전달되고, 정수상수 237 은 argument 2 로 전달되며, 메세지 상수 “PI”는 argument 3 으로 전달됩니다.

CALL Temp (IntegerTag);

IntegerTag 의 값은 argument 표현식의 값으로 전달됩니다.

CALL ValveOpen (Tag.MaxEU -5);

계산된 값(Tag.MaxEU -5)는 argument 표현식 값처럼 전달됩니다.

Argument Data Type Matching

호출하는 쪽의 argument 목록의 데이터 형태와 저장되고 호출된 QuickFunction의 데이터 형태 사이에는 엄격하게 일치가 되어야 합니다.

또한, 저장된 QuickFunction의 argument 목록에서 argument의 개수를 맞추기 위해서, 정확한 argument 개수도 맞춰 주어야 합니다.

강제 형태-변환은 실수에서 정수로의 형태-변환과 정수에서 실수로의 형태-변환을 위해 사용합니다. 데이터 값의 형태를 수정하는 이러한 기능은 어떤 analog argument에서 또 다른 analog 형태로 전달되는 것을 허용합니다.

예로, 실수 값 1.23을 정수 argument로 전달하게 되면 그 값은 1이 될 것이고, .23은 없어지게 됩니다. 유사하게, 정수값 1을 실수 argument로 전달하게 되면, 그 값은 1.0이 될 것입니다. 그러나 이 강제형태-변환기능이 있음에도 불구하고, QuickFunction argument 목록의 argument 형태와 호출하는 쪽의 argument 형태를 일치시키는 것을 추천합니다.

Valid QuickFunction Syntax

QuickFunction은 값을 반환합니다. QuickFunction 문법과 형식은 아래와 같습니다.

CALL QuickFunctionName ([arg1, ... arg16]);

where:

CALL은 QuickFunction을 호출하기 위해서 필요한, 모든 QuickScripts와 표현식에서 요구되는 키워드입니다.

QuickFunctionName은 저장된 QuickFunction에 할당된 이름과 일치하는 1~31개의 문자열입니다.

([arg1, ...arg16])는 중괄호에 0~16개 까지, 콤마로 분리된 argument들입니다.

Using the Return Statement

QuickFunction 은 다른 script(script 를 호출하는 쪽)에 의해 호출될 수 있는 script 입니다.

다음 예제는 QuickFunction 호출 script 입니다.

```
RETURNRESULT = CALL MyFunction( StartHour, EndHour);
```

RETURN 문장은 script 를 호출하는 쪽에서, 반환되는 값을 전달 받기 위한 Quickfunction 으로 사용된다.

RETURN 이라는 문장을 만나게 되면, QuickFunction 은 바로 실행됩니다. 중점은, QuickFunction 은 script 를 호출하는 쪽으로 값을 반환하게 되는 것입니다. 반환된 데이터 형태는 Discrete, 실수, 정수 또는 Message 형태로 될 수 있습니다.

위의 예제에서, **RETURNRESULT** 태그는 변경된 값을 받기 위해 Discrete, Message, 실수 또는 정수형태를 사용 할 수 있습니다.

아래의 예제에서, ConvertType 의 tagname 0이 0 이면, abs 표현식 **abs((Max /Maxprogress) * 100)**는 계산되어 지고, script 를 호출하는 쪽에 결과 값인 실수 형태 데이터를 반환하게 됩니다.

```
IF ConvertType == 0 THEN
    RETURN abs( (Max / Maxprogress) * 100 );
ELSE
    RETURN 0;
ENDIF;
```

문법에 따라 반환된 값의 데이터형태가 결정됩니다.

예로:

```
RETURN AnalogTag;
```

AnalogTag tagname 0| Memory Integer 로 정의되면, **RETURN** 문장은 호출하는 위치에 Analog 값을 전달 할 것이며, 오로지 하나의 값만이 반환됩니다.

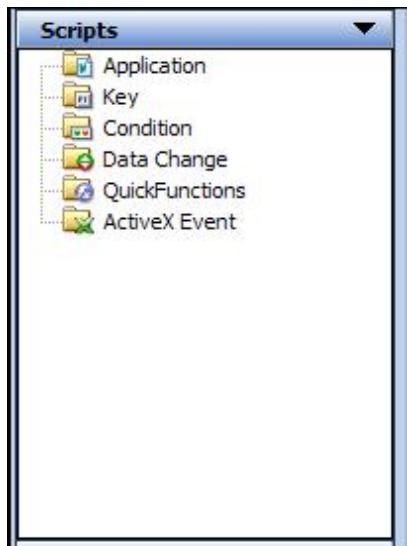
Section 2 - Working with the Script Editor

이번 장에서는 QuickScript Editor의 기능과 특징들을 설명하고 있다. 특정 script type의 유일한 특징들과 기능들은 이번 장 마지막 부분에서 설명할 것이다.

Open the QuickScript Editor

새로운 script를 생성하기 위한 QuickScript editor 실행은 Special menu 또는 Application Explorer(Project View)의 Scripts panel에서 실행할 수 있습니다.

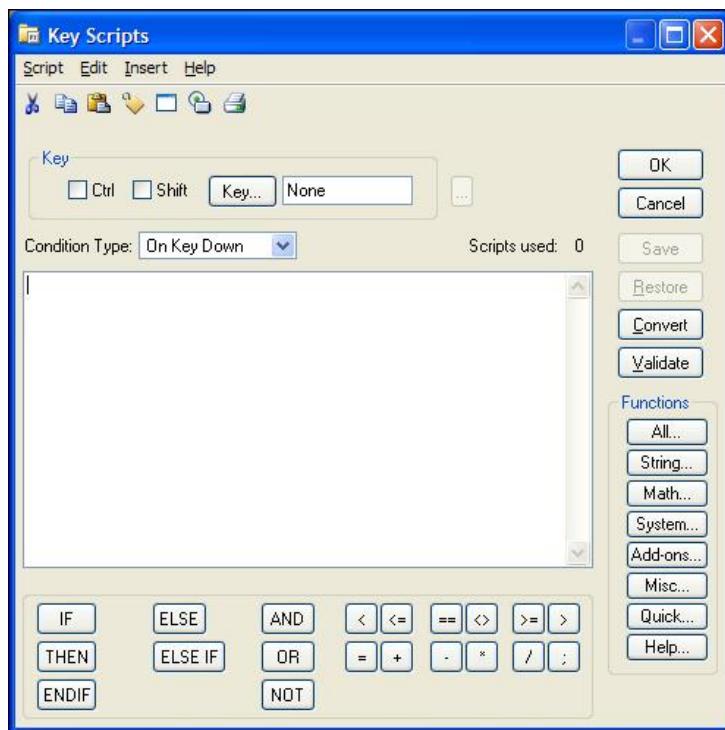
Application Explorer로부터:



- a. QuickScript type에서 오른쪽-클릭을 한 후, **Open** 또는 **New**를 선택합니다. submenu에서는 어떤 QuickScript 형태는 **New** 명령을 포함하고 있지 않습니다.

Note: 기존에 생성되어 있던 QuickScript는 Application Explorer의 **Windows & Scripts** 패널의 Scripts 풀더안에서 알파벳 순서로 보여집니다.

InTouch QuickScript Editor는 모든 script type에서 동일한 기본적인 인터페이스를 보여줍니다.



The QuickScript Toolbar



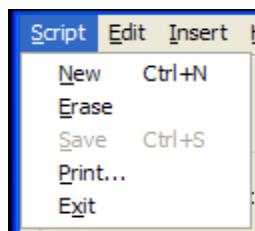
QuickScript toolbar 는 기본적인 편집 기능을 수행합니다.

Toolbar Icon	Description
	Cut selected text from the script
	Copy selected text from the script
	Paste text into the script from the clipboard
	Insert a tag from the Tagname Dictionary
	Insert a window name from the current application
	Insert an ActiveX Control via the ActiveX Control Browser
	Print script

QuickScript Editor는 script 안의 로직 입력 영역을 크게 키우기 위해서 크기를 재조정 할 수 있습니다. 외곽 경계선 부분을 드래그함으로써 editor의 크기를 재조정 할 수 있습니다. 또는 화면상에 editor를 최대화하기 위해서는 **Maximize** 버튼을 클릭할 수도 있습니다.

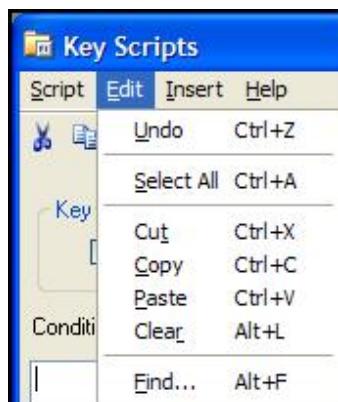
QuickScript Editor Common Command

Script Menu Options



Script 메뉴는 **New**, **Save**, **Print** 그리고 **Exit** 과 같은 일반적인 윈도우 명령을 포함하고 있습니다. 추가적으로, **Erase** 명령은 script를 완전히 삭제하기 위해 사용됩니다.

Edit Menu Options



Undo : 마지막 편집 작업을 되돌립니다.

Select All : script 전체를 선택합니다.

Cut : 제거하려는 text를 선택합니다. 해당 text는 script로부터 삭제되고 윈도우 클립보드로 복사됩니다. 다른 script 또는 이 script의 다른 위치에 해당 문자를 붙여 넣을 수 있습니다.

Copy : 제거하려는 문자를 선택합니다. 문자는 윈도우 클립보드로 복사됩니다. 다른 script 또는 이 script의 다른 위치에 해당 문자를 붙여 넣을 수 있습니다.

Note : 문자를 자르거나 복사 할 경우, 이는 자동적으로 윈도우 클립보드에 쓰여집니다. 이 정보는 다음의 자르기 또는 복사 명령이 수행 되기 전까지 클립보드에 남겨집니다.

Paste: 윈도우 클립보드의 내용들을 커서 위치의 script에 붙여 넣기를 합니다.

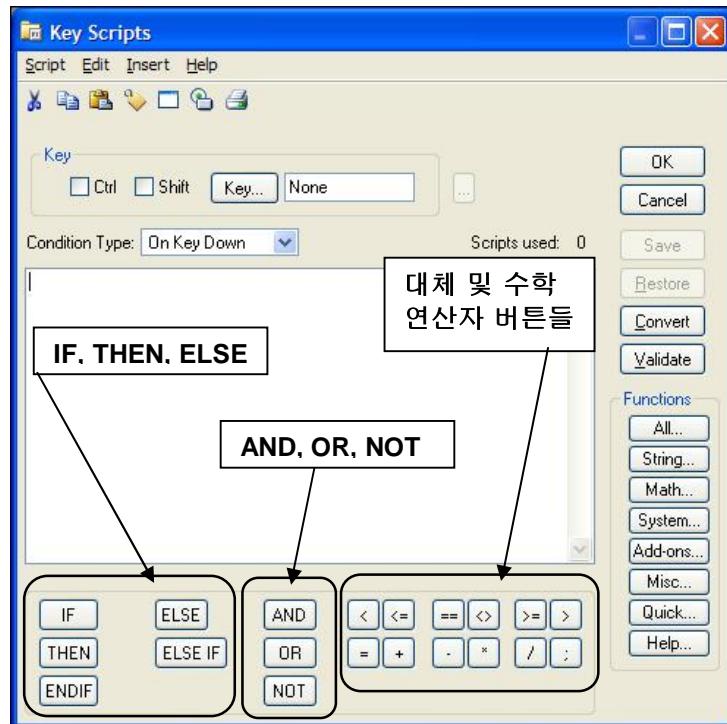
Clear : Script 안의 모든 문자는 지워집니다. 그러나 script가 Application으로부터 지워지는 것은 아닙니다. 만약, 이 명령을 선택한 후, script editor를 취소하고, 다시 열게 되면, 기존의 script가 다기 열릴 것입니다.

Script를 완전히 지우기 위해서는, Script / Erase 명령 또는 script 전체 선택을 사용해야 합니다.

Note: Script / Erase 명령은 어떤 QuickScript type에도 적용할 수 있습니다. 삭제된 문자는 윈도우 클립보드에 쓰여지지 않습니다.

Find : Replace 대화 상자창을 실행합니다.

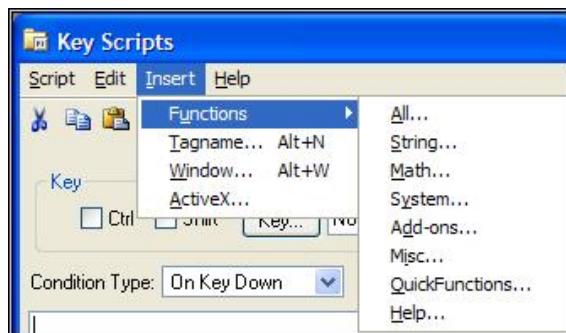
Using the Script Editor Screen



Equivalency and Mathematical Operator Buttons

Equivalency 와 mathematical 작동 버튼은 QuickScript editor 의 하단에 위치합니다. 각각의 버튼은 커서 위치의 script 안에 키워드, 함수 또는 symbol 을 빠르게 삽입하기 위해서 사용 됩니다.

Insert Menu Options



Script Editor 메뉴에서 Insert / Functions 명령을 선택함으로써, 함수를 삽입하고 사용할 수가 있습니다. 함수 category 에서는 이름을 선택합니다.

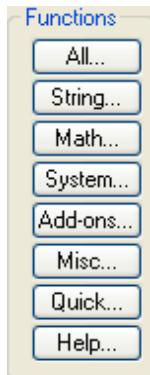
또한 Tagname, Windows, ActiveX control 을 Insert 메뉴로부터 삽입할 수 있습니다.

Built-in Script Functions(내장 script 함수들)

InTouch 는 다중 객체나 버튼 또는 다중 작업을 수행 하기 위한 script 사용을 위해 내장함수들을 제공하고 있습니다.

예로써, 알람을 ACK 시키거나, 윈도우 숨기기, pen 에 의해서 trending 되고 있는 tagname 의 변경 등이 있습니다. 이러한 함수들은 main menu 의 Insert / Functions 을 통하여나, Script Editor 의 Functions 영역의 다양한 버튼들을 클릭함으로써 접근할 수 있습니다.

일단 각각의 창에서 함수를 선택하게 되면, 함수와 요구되는 arguments 는 자동적으로 script 에서 커서의 위치에 붙여 넣기가 됩니다. 함수가 script 에 붙여진, argument highlight 를 수정하고 새로운 값을 입력합니다. 함수는 Script Editor 에 버튼 별로 다음과 같은 순서로 목록화 되어 있습니다.



다음은 이용 가능한 함수들의 명령버튼입니다.

Function	Description
All	각각 설치된 프로그램(Recipe Manager, SPC Pro, SQL Access Manager)에서 사용할 수 있는 함수들과, 사용 가능한 모든 함수들을 선택할 수 있는 창을 보여줍니다.
String	사용할 수 있는 모든 string 함수들을 선택할 수 있는 창을 보여줍니다.
Math	사용할 수 있는 모든 수학연산함수들을 선택할 수 있는 창을 보여줍니다.
System	사용할 수 있는 모든 system 함수들을 선택할 수 있는 창을 보여줍니다. 예를 들면, 다른 application 을 활성화 시키거나, 실행 시키기 위한 함수들, 파일이나 디스크 정보를 읽거나 쓸 수 있는 함수들 등등이 있습니다.
Add-ons	각각 설치된 프로그램(Recipe Manager, SPC Pro, SQL Access Manager)에서 사용할 수 있는, 모든 함수들을 선택할 수 있는 창을 보여줍니다.
Misc	사용할 수 있는 모든 기타 다양한 함수들을 선택할 수 있는 창을 보여줍니다. 예를 들면, 알람을 위한 함수들, trend 를 위한 함수들, windows control 을 위한 함수들, ActiveX Controls 를 위한 함수들 등등이 있습니다.

Quick	현재 script에서 호출하여 사용할 수 있는 모든 QuickFunction을 선택할 수 있는 창을 보여줍니다.
Help	사용할 수 있는 모든 함수들이 목록화되어 해당 함수의 도움말을 볼 수 있도록 보여줍니다. 각각의 도움말을 보려면 함수를 클릭하시기 바랍니다.

Script에 함수 적용

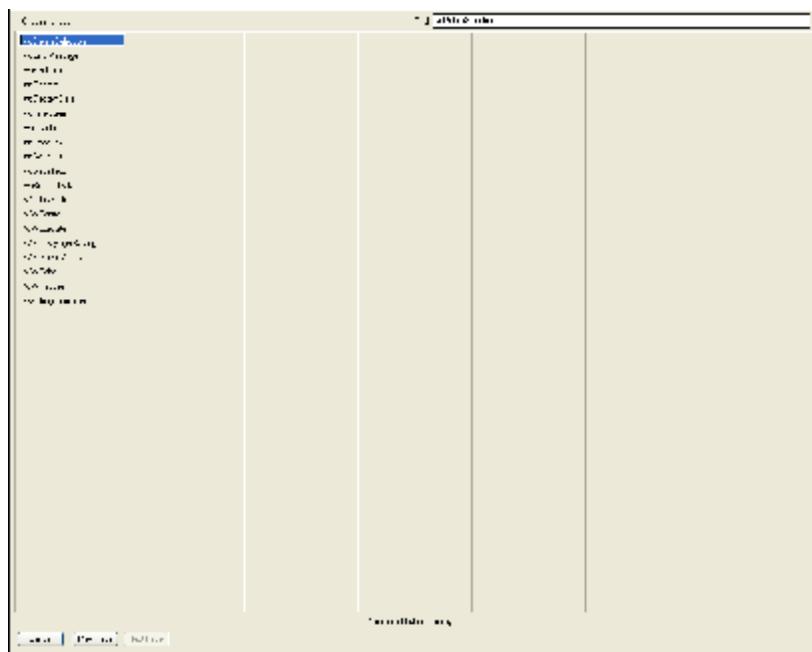
- a. QuickScript editor의 함수 영역에서, All 버튼을 클릭합니다.

Choose function 대화상자가 나타납니다:



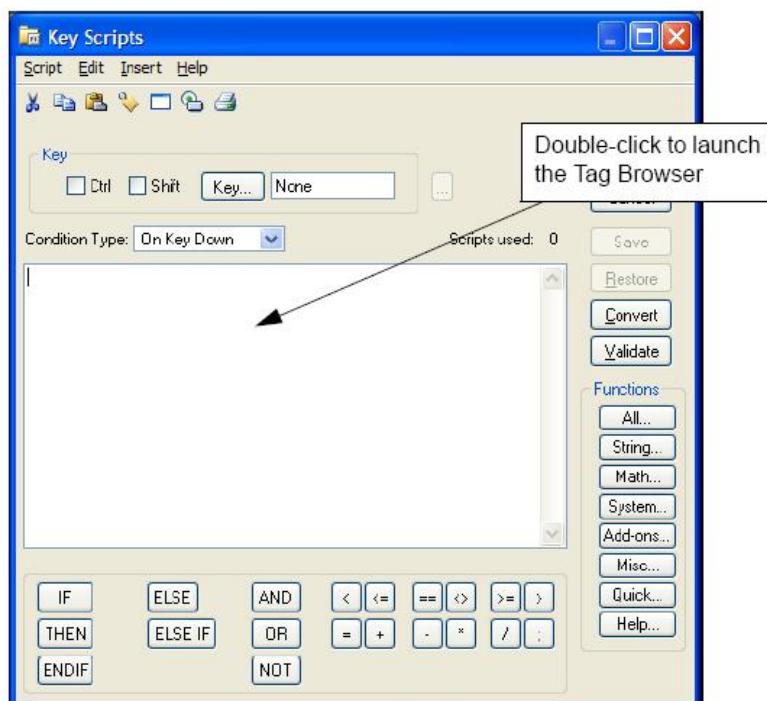
- b. 추가적인 함수들을 보기 위해 **Next Page**를 클릭합니다.

나머지 이용 가능한 함수들이 보여집니다:



함수를 선택한 후, 대화상자는 닫히며 함수들은 **script**에서 커서 위치에 삽입됩니다.

- c. Tag Browser에 빠르게 접근하기 위해, QuickScript window의 빈 영역을 더블-클릭합니다.



- d. Tagname Dictionary에서 특정한 tagname의 정의에 접근하기 위해서, QuickScript 윈도우에서 tagnameDMF 입력한 뒤, 더블-클릭합니다.

Insert a Tagname Dot Field into a Script

- a. Insert / Tagname 선택 : Unlimited selection mode 안에 Tag Browser 가 나타납니다..

Note : Tag Browser를 통해 최근의 Tag Source안에 정의된 Tag을 볼 수가 있습니다.
Tag Source를 변경하기 위해서, Tag Source 화살표를 클릭하고, 목록 안에서 다른 Tag Source를 클릭합니다. Tag Source 목록으로부터 Tag Source를 추가하거나 제거하기 위해, Define Tag Sources 버튼을 클릭합니다.

- b. Tagname을 선택하고 Dot Field화살표를 클릭합니다.
c. 목록안에서 tagname을 사용하기 위해 .field를 선택합니다.
d. OK버튼을 클릭합니다.

선택된 tagname .field는 QuickScript에서 커서 위치에 삽입될 것입니다.

Tip : Tagname .field 를 빠르게 삽입하기 위해서, 마침표(.) 뒤에 따라오는 tagname 을 입력하고, 마침표의 오른쪽을 더블-클릭합니다.

Choose field name 대화상자가 열리면, 사용하기 원하는 .field 를 클릭합니다. 대화상자가 닫히면, 선택된 .field 가 자동적으로 QuickScript 의 커서 위치에 삽입 될 것입니다.

Selecting Dot Fields from the Tag Browser

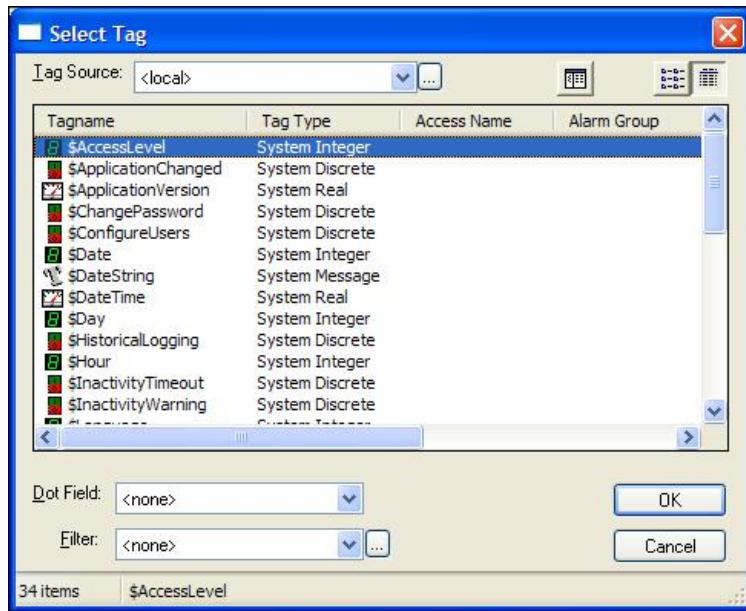
Unlimited selection mode 는 선택된 태그에 tagname .fields 가 선택 될 수 있도록 합니다.

태그와 Dot Field 가 선택 되면, 자동적으로 InTouch QuickScript 이나 Animation link 의 tagname 또는 expression 상자, 혹은 Tag Browser 에 입력됩니다.

Local 이나 원격태그 소스에서 정의된 tagname 은 Unlimited selection mode 에서 선택되거나 볼 수 있습니다.

Unlimited selection mode 는 InTouch QuickScript 윈도우의 빈 영역, Animation link 의 tagname 또는 expression, 또는 Substitute Tagnames 창의 New Name 의 빈 영역을 더블-클릭 함으로써 접근 할 수 있습니다.

또한, Unlimited selection mode 는 Industrial Application Server처럼 미리 정의된 tag source 를 선택하는 것도 허용합니다.



Selecting a Dot Field

- Special / Scripts / Windows Scripts** 클릭 합니다.
- 빈 영역을 더블-클릭합니다.
- 선택된 태그의 형태와 연관될 수 있는 .fields의 목록을 열기 위해 Dot Field 화살표를 클릭합니다.

기본적으로, 초기에는 <none>이 모든 tagname의 형태에 보일 것입니다.

- 선택된 태그에 목록을 추가하기 위해 .field를 클릭합니다.

Unlimited selection mode에서 Tag Browse를 보여주기 위한 주요 방법:

- | Animation link의 tagname 또는 expression 입력상자를 더블-클릭.
- | ActiveX Control 또는 tagname 마법사 또는 expression 입력상자 더블-클릭.
- | InTouch QuickScript 윈도우안에 빈 영역을 더블-클릭.
- | InTouch QuickScript editor안에서 **Insert**메뉴에서 **Tagname** 명령 선택.
- | InTouch QuickScript editor안에서 **ALT + N** 키를 누릅니다.
- | Substitute Tagnames 창에서 빈 영역인 New Name상자를 더블-클릭.
- | SQL Access BIND List Configuration 창에서 Tagname.**FieldName** 입력상자를 더블-클릭.

Tag Browser의 상태 바는 현재 보여지는 tag source에 아래 항목의 상태를 보여줍니다:

- | Application안에서의 item의 총 개수
- | 현재 선택된 item의 이름

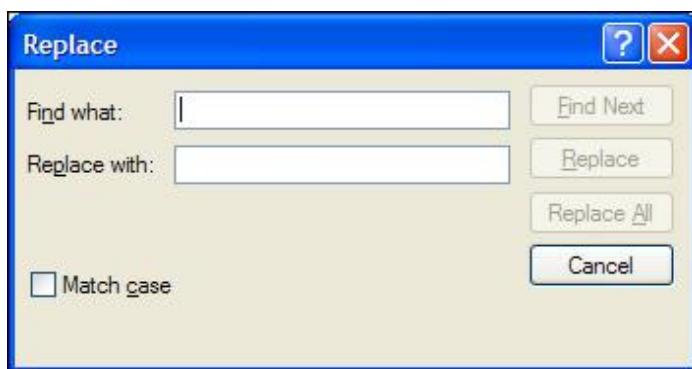
- | 선택된 tagname.field
- | Tag source에 연관된 Access Name

Note: Dot field 는 Module 6, “Alarms”에서 자세히 설명 할 것입니다..

Find or Replace within a Script

- a. **Edit / Find** 를 선택합니다.

Replace 창이 나타납니다.



- b. 찾기를 원하거나 교체하기 원하는 item을 입력하고, **Find Next** 버튼을 클릭합니다.
- c. 이전의 tagname을 교체하기 위해, **Replace with** 상자에 새로운 item을 입력하고 **Replace** 또는 **Replace All** 버튼을 클릭합니다.
- d. Match case옵션은 item의 특정한 대문자나 소문자의 instance를 찾습니다.
- e. 끝낼 때에는, **Cancel** 버튼을 클릭합니다.

Tip : item 의 어떤 instance 를 교체하기 원하면 , **Find Next** 를 클릭합니다. InTouch 는 예전 item 의 script 를 검색 하기 시작하며, 예전 item 이 발견되면, 표시가 됩니다 새로운 item 으로 교체하기 위해 **Replace** 버튼을 클릭하거나, 또는, skip 하기 위해서 **Find Next** 버튼을 클릭하면 계속해서 검색하게 됩니다. 검색된 특정한 item 을 모두 교체하기 위해서는, 검색하는 동안 어떠한 때라도 **Replace All** 버튼을 클릭합니다.

Insert a Window Name into a Script

- a. **Insert / Window** 를 선택하거나, **Insert Window** 아이콘을 클릭합니다.  **Window Name to Insert** 창이 나타나고, Application안의 모든 윈도우 이름들이 보여집니다.



- b. 원하는 윈도우 이름을 클릭합니다. 창이 닫히고 윈도우 이름이 자동적으로 script안에 커서 위치에 삽입됩니다.

Validate a Script

- a. Script 문법이 적절한지를 증명하기 위해서는 Validate 버튼을 클릭하며, 언제라도 실행 가능합니다.
유효성은 OK 나 SAVE 버튼을 클릭할 때, 자동적으로 수행됩니다. Script의 유효성을 검사하는 동안, 만약 system에 에러 상황이 발생된다면, 에러 메시지 창이 나타납니다. 메시지창이 나타나지 않는다면, script는 실행됩니다.

Save a Script

- a. 만일, 다수의 script들이 설정되면, 최초의 script를 저장하기 위해 Save를 클릭합니다.
b. 새로운 script를 쓰기 위해 Script / New를 클릭합니다.
Action, Application, Window script는 이러한 기능을 지원하지 않습니다.

Restore a Script

- a. 만일, script를 변경하고, 변경사항들을 명확하게 정의하고 원래 script에 재저장하기를 원한다면, Restore 버튼을 클릭합니다.
일단 저장이 되고 나면, script는 재저장 될 수 없습니다. Application과 Window script는 이러한 기능을 지원하지 않습니다

Exit the Script Editor

모든 script editors는 OK 버튼을 클릭하거나, caption bar의 상단 오른쪽에 X버튼을 클릭함으로써 종료 시킬 수 있습니다.

- | Editor에서 Script / Exit를 선택하면, script editor는 닫힙니다.
- | OK 버튼 또는 Script / Exit 클릭은 script editor 닫기를 실행 할 것이고, 만약 error 상황이 없다면, 유효성 검사를 하고, 정상적으로 저장될 것입니다.
- | X 클릭은 script editor를 종료시킬 것입니다. 변경된 사항들은 반영되지 않으며, 유효성 검사는 실행되지 않습니다.

Specify a Script's Execution Frequency

While Running / Showing / Down Every 0 Milliseconds 에는 script 가 실행 되기 전 반드시 경과해야 하는 millisecond 숫자를 입력해야 합니다. Application script 의 While Running, Window script 의 While Showing , Condition script 의 While On True / On False 또는, Key script 와 Touch Pushbutton-Action script 의 While Down 을 이용하여 script 를 생성 할 경우에는, 실행되는 빈도를 반드시 명시해야 합니다.

Note : WindowView 는 명시화된 시간만큼 빠른 script-type 의 실행 속도를 보장하기 위해서 가능한 모든 시도를 수행합니다. 그러나 performance 를 보장할 수는 없습니다. 또한, script 는 결코 WindowViewer 속성에서 명시된 설정 Tick Interval 보다 빨리 실행 될 수도 없습니다.

IF-THEN-ELSE and Comparisons

IF-THEN-ELSE 문법은 표현하는 상태에 근거하여, 조건부로 다양한 로직을 실행하기 위해서 사용됩니다.

아래의 비교 연산자들은 IF-THEN-ELSE 문장에서 조건들을 설정하기 위해 사용됩니다.

<	~ 보다 작은
>	~ 보다 큰
<=	~ 보다 작거나 같은
>=	~ 보다 크거나 같은
==	양쪽이 동등한
<>	양쪽이 동등하지 않은

복잡한 script 의 예는 다음과 같습니다.

Note : 각각의 IF 는 ENDIF 와 함께 사용되어야 하며, 세미콜론(;)은 각 문장이 끝날 때마다 반드시 들어가야 합니다.

IF-THEN statement with no ELSE clause

```
IF a <> 0 THEN
    a = a + 100;
ENDIF;
```

IF-THEN-ELSE statement with one ELSE clause

```
IF temp > 500 THEN
    Disc = 1;
    Real = 43.7;
ELSE
    Disc = 0;
    Real = 93.4;
ENDIF;
```

IF-THEN-ELSE statement with one ELSE IF clause and no ELSE clause

```
IF temp > 500 THEN
    Disc = Disc * 10;
ELSE
    IF temp > 250 THEN
        x = y / z;
        a = abc + def;
    ENDIF;
ENDIF;
```

IF-THEN-ELSE statement with one ELSE IF clause and one ELSE clause

```
IF temp > 500 THEN
    Disc = Disc - 10;
ELSE
    IF temp < 250 THEN
        Disc = Disc + 10;
    ELSE
        Disc = Disc + 50;
        Real = 100;
    ENDIF;
ENDIF;
```

IF-THEN-ELSE statement with multiple ELSE IF clauses and one ELSE clause

```
IF temp > 100 THEN
    temphihi = 1
    Disc = 50;
ELSE
    IF temp > 80 THEN
        temphi = 1;
    ELSE
        IF temp < 10 THEN
```

```

templo = 1;
ELSE
    IF temp < 30 THEN
        templolo = 1;
    ELSE
        tempok = l;
    ENDIF;
ENDIF;
ENDIF;
ENDIF;

```

IF-THEN-ELSE statement that tests for Condition 1 or Condition 2

```

IF (pump1 < 50.0) OR (pump2 < 50.0) THEN
    alarm1 = 1;
ELSE
    alarm1 = 0;
ENDIF;

```

IF-THEN-ELSE statement that tests for Condition 1 and Condition 2

```

IF (pump1 < 50.0) AND (pump2 < 50.0) THEN
    alarm2 = 1;
ELSE
    alarm2 = 0;
ENDIF;

```

IF-THEN-ELSE statement that tests for equivalency:

```

IF a > 50 THEN
    IF b == 100 THEN
        c = 0;
    ENDIF;
ENDIF;

```

Notes - Math Functions

Math 함수들은 실수 또는 정수 tag type에 사용됩니다.

Math 함수에서 **ResultNumericTags** 와 **InputNumericTags** 는 실수 또는 정수로 될 수 있으며, 자유로이 변환됩니다.

정수 **tagname**에 정수가 아닌 함수의 결과 값을 반환하게 되면, 결과 값은 틀어지게 됩니다.
바꿔 말하면, 십진수의 오른쪽 부분이 손상되어 정확한 값이 나오지 않게 됩니다.

Notes - System Functions

system 함수는 다른 윈도우들을 활성화시키거나 파일을 복사, 삭제 또는 이동, InTouch **application**과 관련된 정보들을 반환하거나 하는 작업들을 수행하기 위해서 사용됩니다.
System 함수는 세가지 파일 type(File 정보, Application 정보와 System 정보)로부터 데이터를 읽거나 쓸 수 있습니다.

Notes - WW I/O Functions

DDE와 관련된 함수들은 보통의 InTouch DDE 통신을 하기 위한 대체 함수들이 아닙니다. 다만, 언제라도 가능하게, 외부 Application으로부터 데이터를 보내거나 받기 위해서 I/O 형태 tag를 생성 할 수 있습니다.

WW DDE 함수들은 InTouch에 의해 지원되는 DDE Advises를 사용해서 통신할 수 없는 application을 지원하기 위해 사용할 수 있습니다.

WWExecute(), **WWPoke()**와 **WWRequest()** 함수들은 Microsoft Visual Basic (DDEML)처럼 같은 윈도우 함수를 사용합니다.

단일 함수는 실제로 여러 작업들을 수행합니다. 예로, **WWPoke()**는 한 함수에서, DDE 초기화, DDE Poke, DDE 종료가 수행됩니다. 또한, WW DDE 함수들은 자주 발생하는 error들을 비교적 덜 발생시키지만, 많은 DDE 메시지를 보내는 과정에서의 능률은 떨어집니다.
이들 함수의 사용함에 있어서의 일반적인 guide line:

- | 함수들을 반복해서 사용할 수 있습니다.(함수들을 반복적으로 호출할 수 있습니다)
- | 같은 script안에서 혹은 한 개의 열에서 DDE 함수들을 여러 개 호출할 수 있습니다.
- | 다른 DDE Application에서 performance가 큰 작업을 호출하기 위해서 함수들을 사용할 수 있습니다.

DDE 명령어가 다른 Application에서 performance가 큰 작업을 실행하게 되면, 여기에 사용 가능한 모든 processor를 사용하게 될 수도 있습니다. 그에 따른, 통신장애가 발생하게 될지도라도, 데이터의 손실은 발생하지 않습니다. 만일, I/O 서버가 InTouch에 데이터를 보낼 수 없게 된다면, I/O 서버는 계속해서 데이터 전송을 시도 할 것입니다.

Importing QuickScripts

하나의 InTouch Application 으로부터 다른 Application 으로 QuickScripts 를 가져온다는 것은 개발 시간을 줄일 수 있다는 것입니다. 이는 이전에 만들어 놓은 QuickScripts 를 재사용 할 수 있다는 것입니다. QuickScript 를 다른 InTouch Application 으로 이동 시키기 위해서는, **File / Import** 명령을 사용해야 합니다.

- a. 현재 application 에서 모든 윈도우들을 닫습니다.
- b. **File / Import** 를 클릭합니다.

Import from directory 창이 나타나게 됩니다:



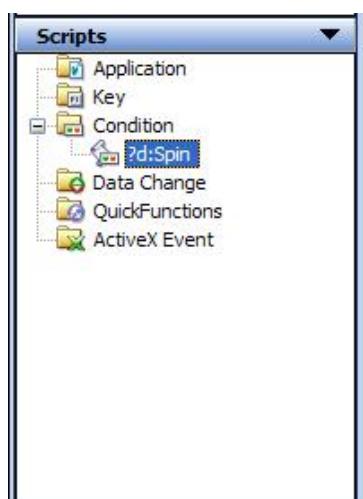
- c. Import 하려는 QuickScript 가 있는 Application directory(폴더)를 선택합니다.
- d. **OK** 버튼을 클릭합니다. **Application Data Import Options** 창이 나타납니다.
- e. **Condition Scripts** 를 선택합니다.



- f. **Select** 를 클릭합니다. **Select a Condition Script** 창이 나타납니다:



- g. Import 하려는 QuickScript 를 선택하고, 창을 닫기 위해서 OK 버튼을 클릭합니다.
h. **Import** 를 클릭합니다. System 은 현재 실행된 application 으로 선택된 QuickScript(s)를 import 하기 위해 자동적으로 시작하게 될 것입니다.



Importing ActiveX Event Scripts

ActiveX Event script 는 ActiveX Control Event 의 결과처럼 어떤 Event 의 발생과 함께 임의의 기능을 실행시키기 위해서 사용이 됩니다. 예로, 클릭, 더블-클릭, 마우스-휠-다운 그리고 키 누름 같은 것들은 많은 ActiveX Control 들 안에서 사용되는 전형적인 Event 들입니다.

InTouch ActiveX Event script 는 Event action 을 지원하기 위해 제공됩니다. 각각의 Event 에 임의의 어떤 ActiveX Event script 를 연결시킬 수 있으며, Runtime 시에는 ActiveX Control Event 를 실행합니다.

InTouch QuickScript 함수들은 ActiveX Control Event 들을 조작하고, Control method 를 호출하며, 속성들을 조정하기 위해서 사용될 수 있습니다. InTouch QuickScript 함수들을 이용하여 ActiveX method 를 실행 시킬 수 있으며, Event 가 발생할 경우, 실행되는 ActiveX Event 에 ActiveXEvent script 를 연결 시킬 수 있습니다.

Runtime 시에, WindowMaker 에서 정의한 tagname 과 QuickScript 로 ActiveX Control 의 실행 기능을 조정할 수 있습니다.

ActiveX Event script 가 하나의 application 에서 다른 application 으로 이동하게 될 경우, 모든 ActiveX Event script 들이 이동하게 됩니다.

또한, 새로운 Application 의 적당한 함수를 import 한 ActiveX Event script 에 의해서, 새로운 Application 에 이전에 생성한 script 와 ActiveX Control, Event 를 사용 할 수 있게 합니다.

만약, ActiveX Control 을 포함하는 윈도우가 닫히면, ActiveX Event script 또는 ActiveX Control 과 연관된 script 함수를 포함하는 어떤 다른 InTouch QuickScript 는 올바르게 실행되지 않을 것입니다.

Converting Placeholder Tagnames

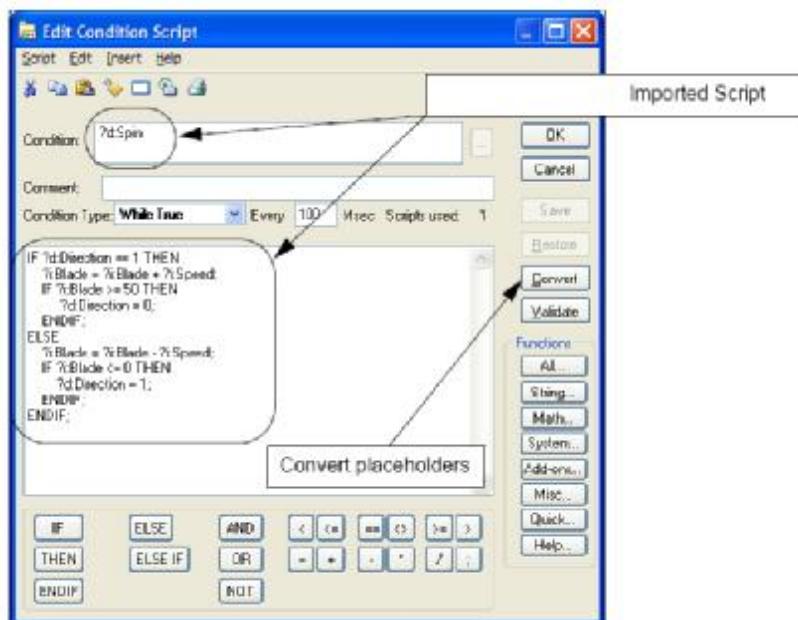
새로운 Application 이 QuickScript 를 import 면, 그 QuickScript 안의 모든 tagname 도 함께 import 하게 됩니다. 그러나, tagname 은 Tagname Dictionary 에 추가 되지 않습니다. 대신에, 그 tagname 은 자동적으로 “placeholder” 로 바뀝니다.

Placeholder tagname 을 tagname 으로 사용하기 위해서는 placeholder tagname 을 바꾸어야 합니다. 만약, 새로운 application 의 Tagname Dictionary 에 tagname 이 정의 되어 있지 않다면, 각각을 정의하기 위한 요청을 받게 될 것 입니다.

Import 한 QuickScript 내의 tagname 들이 placeholder tagname 으로 변경 될 때, 3 가지 문자가 각각의 tagname 앞에 붙여 집니다. 예를 들어, Discrete tagname 을 가져 왔을 경우, 3 개의 문자 ?d:가 접두사로 붙게 됩니다.

문자의 길이가 30, 31, 또는 32 개인 tagname 을 가져오게 될 경우에는, 3 가지 인덱스 문자가 각 tagname 의 앞에 추가 될 것입니다. 그러나 이들 3 가지 문자의 추가로 존재하는 tagname 의 길이를 자르거나 생략 하자는 않습니다. 즉, 32 개 문자의 tagname 은 35 문자로 증가됩니다.

이러한 3 개의 추가공간은 오직 placeholder tagname 에서만 허용됩니다. Tagname 의 길이를 늘리는 것은 기본적은 tagname 에서는 지원되지 않는 사항입니다.



수정하기 위해 script 를 선택할 경우, **Convert** 버튼을 클릭하게 되면, **Convert** 창이 나타납니다.



Tagname Dictionary 에 현재 없는 어떤 태그들을 정의하기 위한 창이 바로 나타나게 될 것입니다.

Note : 다음 랙에서 Tagname 변환을 설명합니다.

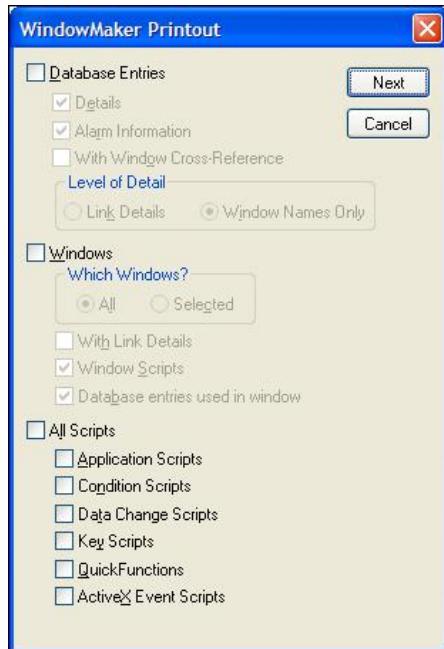
Script Editor Printing

Script editor에서 현재 script를 출력하기 위해서는 프린트 icon 을 사용합니다. 출력을 위해서는 Network에 연결된 프린터를 가지고 있어야 하며, 프린트 함수는 할당된 조건에 맞는 모든 script들을 프린트 할 것입니다.

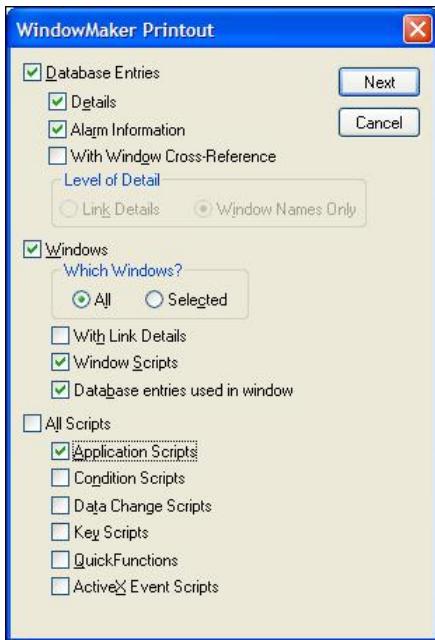
Printing Scripts

각 InTouch QuickScript category에 있는 모든 script는 출력할 수 있습니다.

- a. WindowMaker에서 **File / Print**를 클릭합니다.
WindowMaker Printout 창이 나타납니다.



Database Entries Windows와 출력하려는 script들을 선택합니다.



- b. 윈도우 script 를 출력하기 위해서는, **Windows** 를 선택한 후, **Window Scripts** 를 선택합니다.
- c. Application 안의 모든 윈도우에 있는 script 를 출력하기 위해서는, **Which Windows?** 영역의 **All** 을 선택합니다.



- d. 특정한 윈도우의 script 를 출력하기 위해서는 **Selected** 옵션을 선택합니다.
Windows to Print 창이 나타납니다.



- e. 출력하기 원하는 script 의 윈도우를 선택하고, OK 버튼을 클릭합니다.

Note : 만일, 연결된 script 가 없는 윈도우를 선택한다면, 다음과 같은 제목이 출력물에 인쇄될 것입니다 : **Window Scripts for Window Name: none.**

- f. QuickScript type 의 모든 script 를 출력하기 위해서, QuickScript type 을 선택한 후, Next 버튼을 클릭합니다.

- g. **Select Output Destination** 창에서, **Send output to Text File** 을 선택 합니다.



- h. 출력파일에 이름을 입력하고, **Print** 버튼을 클릭합니다.

출력파일은 기본 파일 저장소에 저장됩니다.

Note : Text 파일 이름을 입력한 후, [...] 버튼을 클릭함으로써 **Save Application Data to File** 윈도우를 볼 수가 있을 것입니다. 그리고, text 파일을 저장하기 위한 위치를 선택할 수 있을 것입니다.

Lab 6 - InTouch QuickScripts 사용

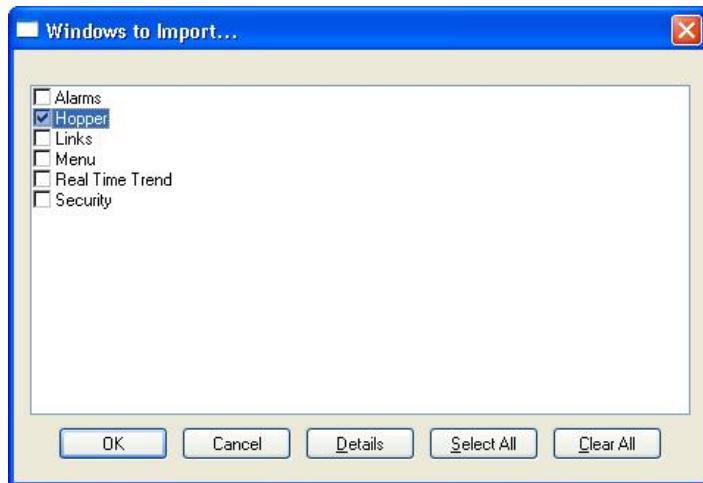
소개

이 랙에서, InTouch Scripting 을 사용하여 자동적인 System 함수들과 다양한 사용자-정의 함수들을 생성할 것입니다.

Lab Instruction

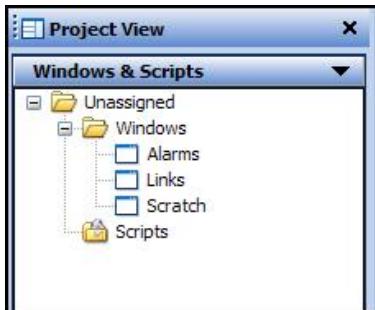
Import the Hopper Window

1. **File / Import** 명령을 사용해서 **Hopper** 원도우를 import 합니다.
2. 적절한 directory(폴더)를 선택합니다. directory 가 선택되면 **Windows to Import** 창이 나타납니다.



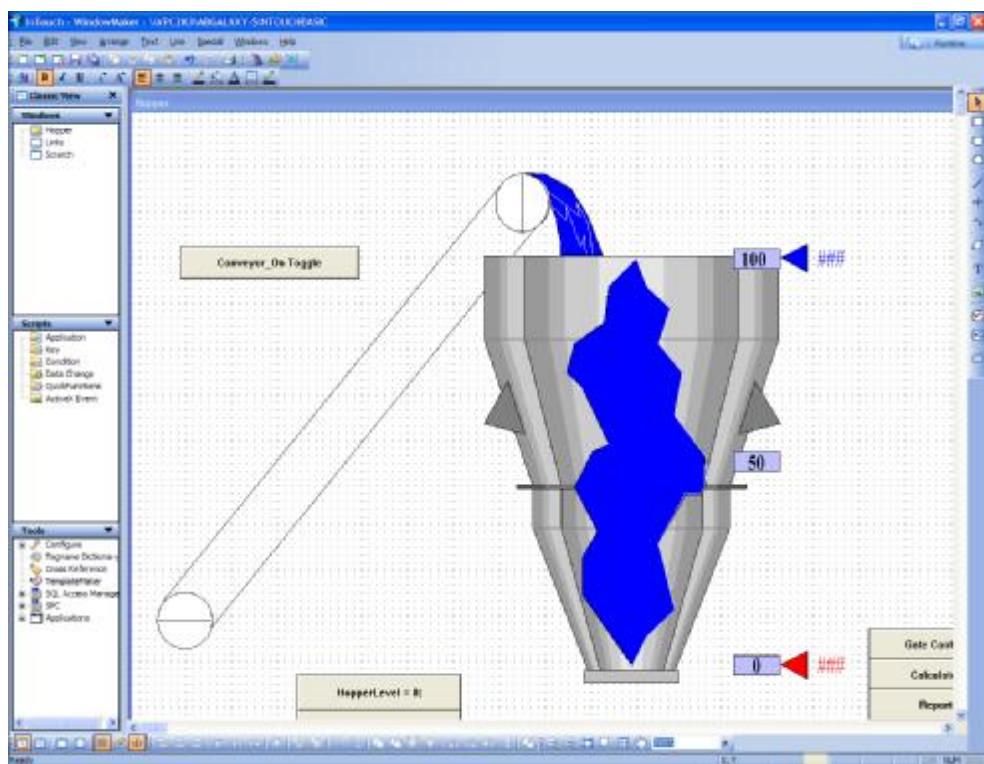
3. **Hopper** 원도우를 선택합니다.
4. **OK** 버튼을 클릭하고 가져옵니다.

Application Explorer 의 원도우 panel 에 Hopper 원도우가 나타납니다.



5. Hopper 윈도우를 더블-클릭합니다.

WindowMaker에 Hopper 윈도우가 나타납니다 :



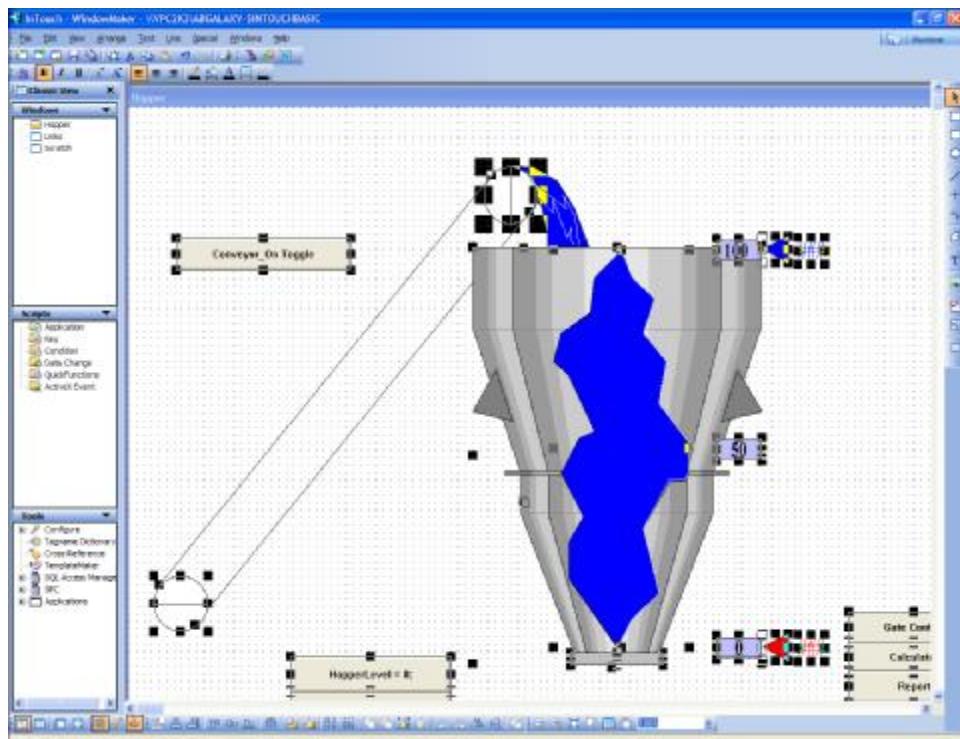
Hopper 윈도우의 그래픽 정보들을 아래에서 설명하고 있습니다 :

- | Hopper 그래픽은 533 픽셀입니다.
- | 파란색 pointer(오른쪽 맨 위)가 Hopper 의 옆 면을 따라서 위, 아래로 움직일 때, **HopperSetPoint tag** 는 0 – 100 까지의 값을 표현하며, 이 pointer 가 value slider 입니다.
- | vertical location 으로 값을 보여주는 animation 은 **HopperSetpoint** 의 현재 값을 보여줍니다.

- | 빨간색 pointer(맨 아래)는 vertical location link이며, pointer와 값을 보여주는 animation 역시 vertical location link입니다. 양쪽 다 **HopperLevel tag**를 할당합니다.
- | hopper 는, **HopperLevel tag** 가 상승하고, 떨어질 경우(0 - 100), 0 - 100%로 색을 채우기 위해 percent fill link를 사용하는 다각형입니다.
 - | Conveyor wheels(,)은 \$Second를 할당 받으며, **Conveyor_on** 이라고 불려지는 **Memory Discrete tag**를 가집니다. Wheels는 **Conveyor_on**이 1이 될 경우에만, 돌아가는 것이 보일 것입니다.
- | Fill material은 conveyor wheels과 같은 방법으로 연결되어 있습니다.
- | 연결된 **Conveyor_on Toggle** pushbutton은 **Conveyor_on tag**를 설정하거나, 재설정하기 위해 사용됩니다.
- | **Gate Control, Calculator** 와 **Report** 버튼과 다른 윈도우 요소들을 scripting 하는 것은 기존에 사용하고 있던 그래픽(예를 들어, 버튼을 클릭함으로써 hopper의 level을 올리거나, hopper gate를 열거나 닫는 action)을 좀 더 유용하게 사용할 수 있다는 것을 뜻합니다.

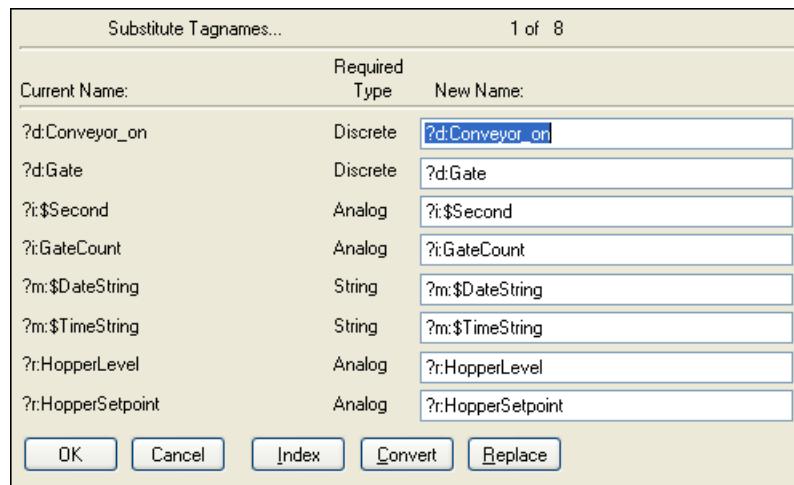
Convert The Hopper Window Tags

6. 모든 객체를 선택하기 위해 F2 키를 누릅니다.



7. **Special / Substitute Tag**를 클릭합니다.

8. **Substitute Tagnames...**창이 나타나며, import 한 tag 들은 InTouch 에 의해 placeholder 로 바뀝니다.

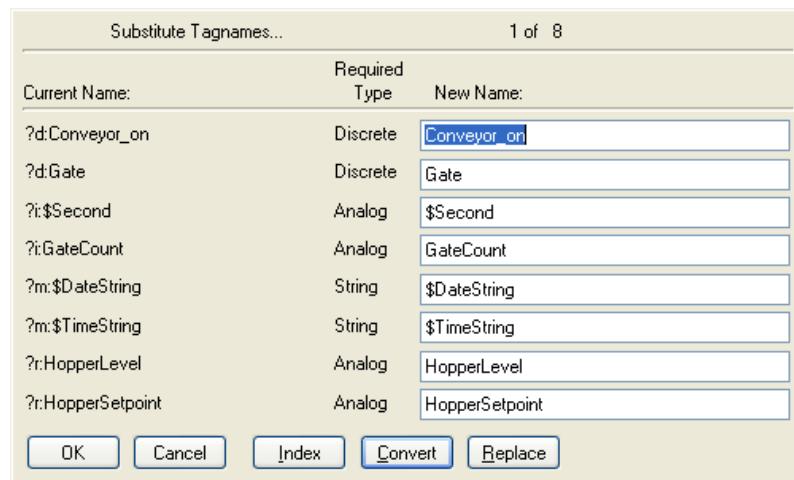


9. **Convert**.버튼을 클릭합니다.

10. **Local** 버튼을 클릭합니다.



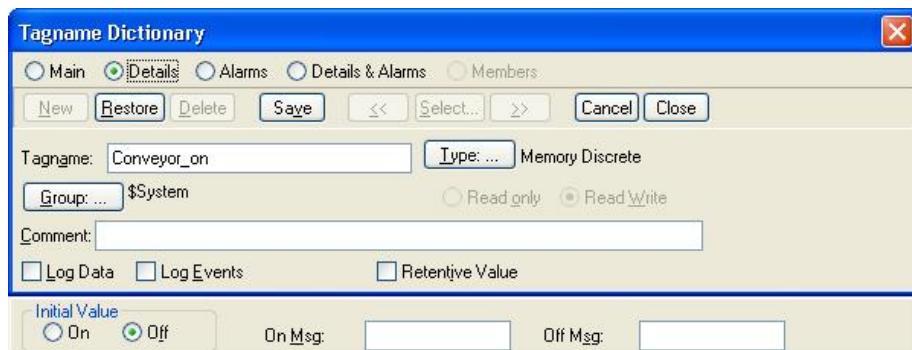
11. tag 의 Placeholder 값이 제거되었다면, **OK** 버튼을 클릭합니다.



12. Tag 들은 아직 존재하지 않으므로, InTouch 는 tag 들을 정의하기 위해 메시지 창을 띄울 것입니다. **OK** 버튼을 클릭합니다.



13. 각 tag 들을 아래와 같이 정의합니다. **Save** 버튼을 클릭하고, 각 tag 의 속성을 설정 한 후, **Close** 버튼을 클릭합니다. Conveyor_on tag 의 정의는 아래의 그림과 유사할 것입니다.



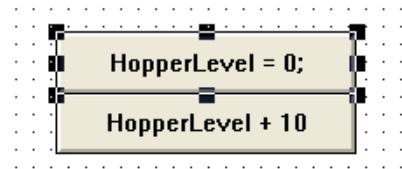
Tag	Type	Initial Value	Min Value	Max Value
Conveyor_on	Memory Discrete	Off	N/A	N/A
Gate	Memory Discrete	Off	N/A	N/A
GateCount	Memory Integer	0	0	100
HopperLevel	Memory Real	0	0	100
HopperSetpoint	Memory Real	0	0	100

새로운 tag 가 모두 정의된 후, Substitute Tags 창이 다시 나타납니다.

14. 창을 종료하고 tag 설정을 저장하기 위해서 **OK** 버튼을 클릭합니다.

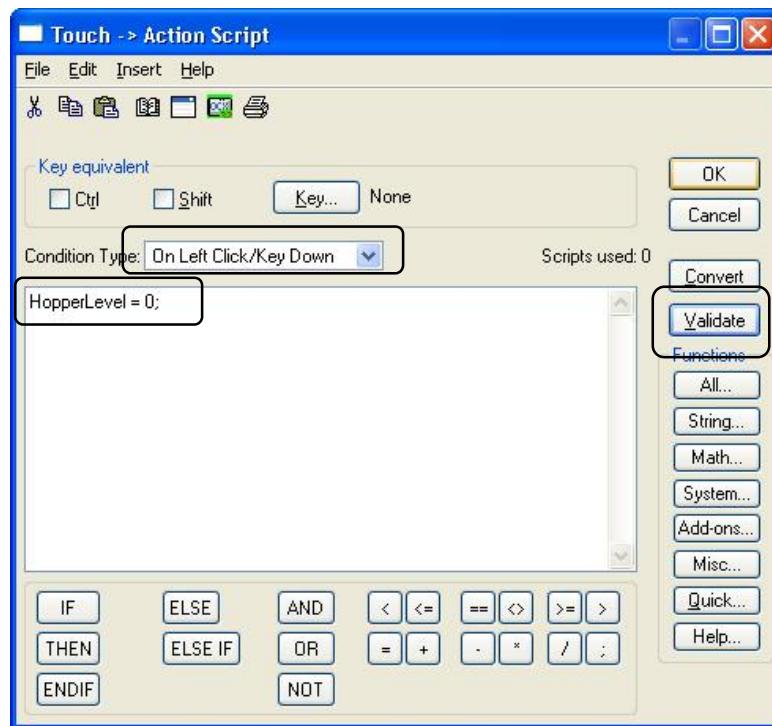
Animate Hopper Window Pushbuttons

15. **HopperLevel = 0;** 버튼을 더블-클릭합니다.

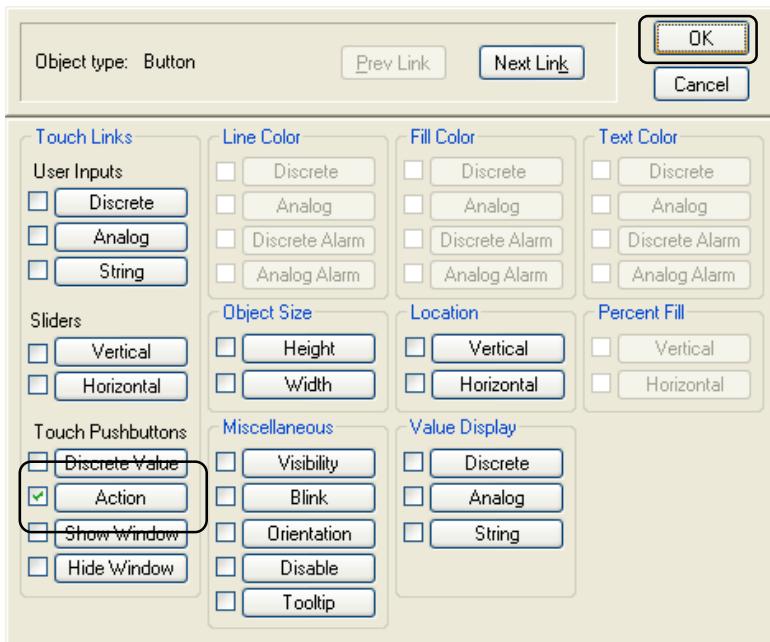


Animation Selection 창이 열립니다.

16. Touchlinks - Touch Pushbutton - Action animation link 를 선택하면, **Touch -> Action Script Editor** 가 보일 것입니다.
17. Condition Type = **On Left Click/Key Down** 가 선택 되었는지를 확인합니다. 이 script 는 버튼을 누른 후 어떤 action 을 제공합니다.
18. Action Script 를 다음과 같이 설정합니다.



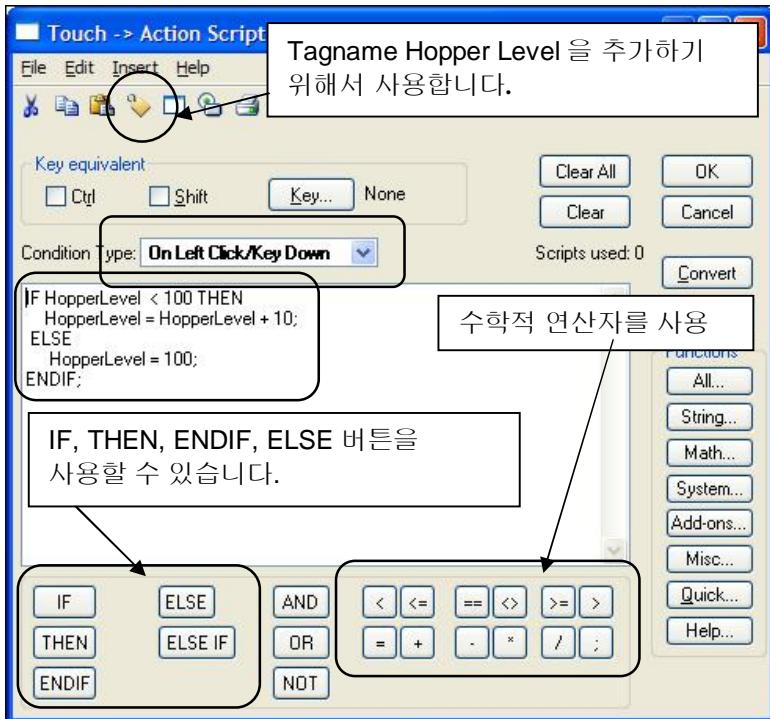
19. Error 를 검사하기 위해서 **Validate** 버튼을 클릭합니다.
20. Script 를 저장하고, script editor 를 종료하기 위해서 **OK** 버튼을 클릭합니다.



21. Animation Selection 대화상자를 닫기 위해 **OK** 버튼을 클릭합니다.
22. **HopperLevel + 10** 버튼을 더블-클릭하면,
Animation Selection 창이 나타납니다.
23. **Touchlinks - Touch Pushbutton – Action** animation link를 선택합니다. **Touch -> Action Script editor** 창이 나타날 것입니다.
24. 이전 script 를 제거하기 위해 **Edit / Clear** 를 사용합니다.

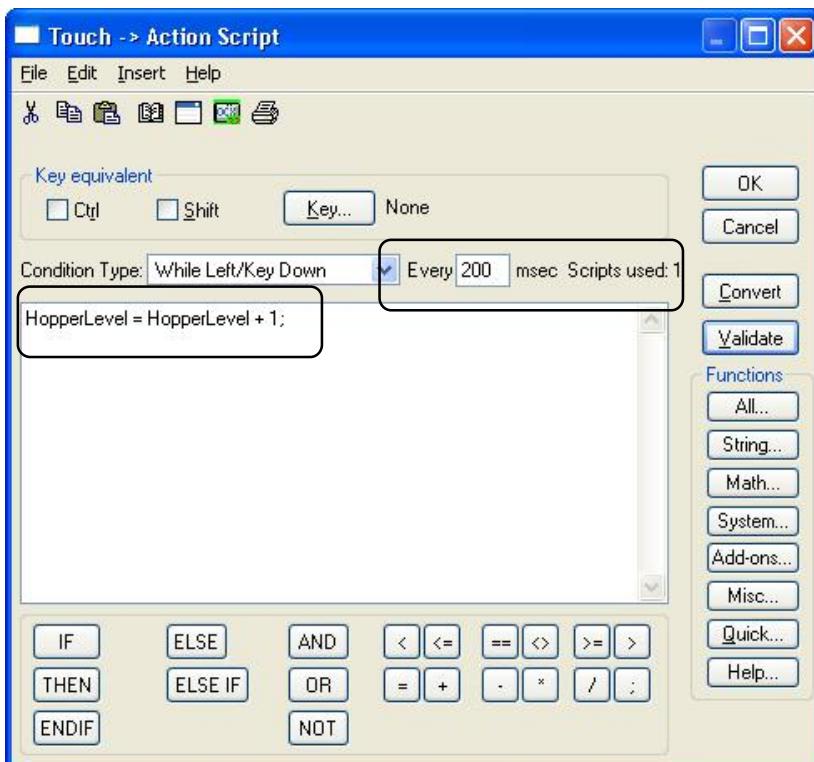


25. 다음 그림에서 보여지는 script 를 입력합니다.
26. **Validate** 버튼을 클릭합니다.

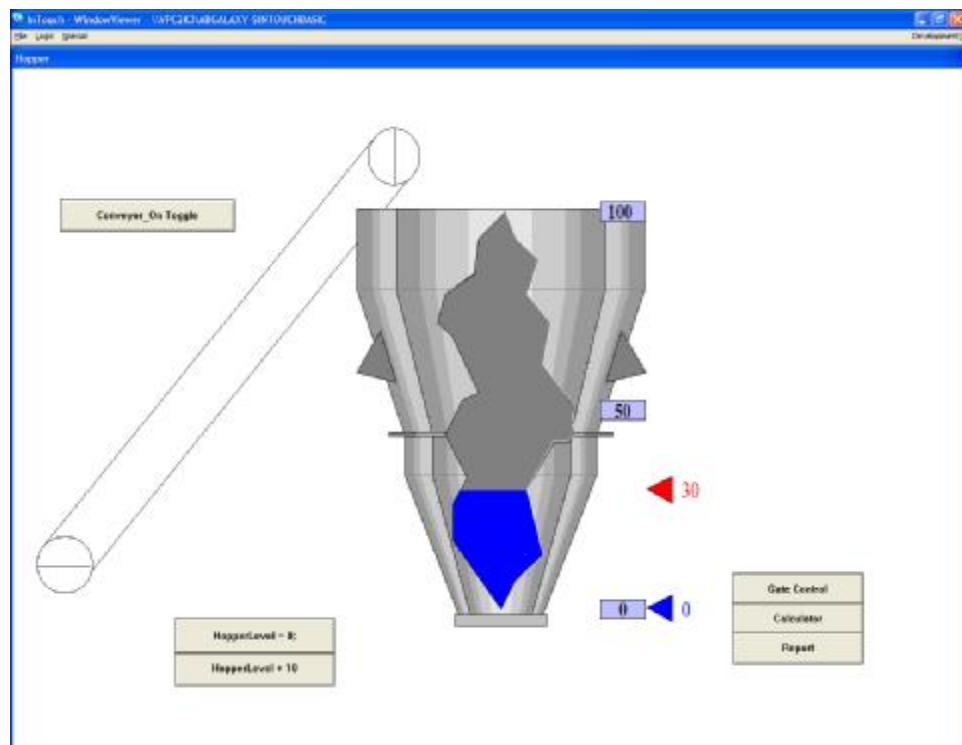


27. Condition type: While Left/Key Down を 선택합니다.

28. 다음 그림에서 보여지는 script 를 입력합니다.



29. Script 입력 후 **Validate** 버튼을 클릭합니다.
30. **Touch -> Action Script editor** 를 종료하기 위해 **OK** 버튼을 클릭합니다.
31. **Animation Links Selection** 창을 종료하기 위해 **OK** 버튼을 클릭합니다.
32. Runtime 모드로 전환합니다.
33. **HopperLevel + 10** 버튼을 클릭합니다.
Level 은 10 씩 증가하며, 빨간색 pointer 와 text display 는 변경될 것입니다.
34. **HopperLevel = 0** 버튼을 클릭하게 되면,
그 버튼은 hopper level을 0으로 재설정하게 됩니다.



35. **Development** 모드로 변환합니다.

Create Condition Scripts

Hopper 의 색상을 채우기 위한 animation 을 설정하기 위해서는, **Hopper_Level** 의 값을 변경시키거나 제어하기 위해서는 Condition script 가 필수적입니다.

첫 번째 Condition script 는 **Conveyor_on tag** 가 참일 때, 100msec 마다 1 씩 **HopperLevel tag**의 값을 증가 시킵니다.

36. **Scripts** 의 앞부분에 **+** 부호를 클릭함으로써 **Scripts** 목록을 펼칩니다.

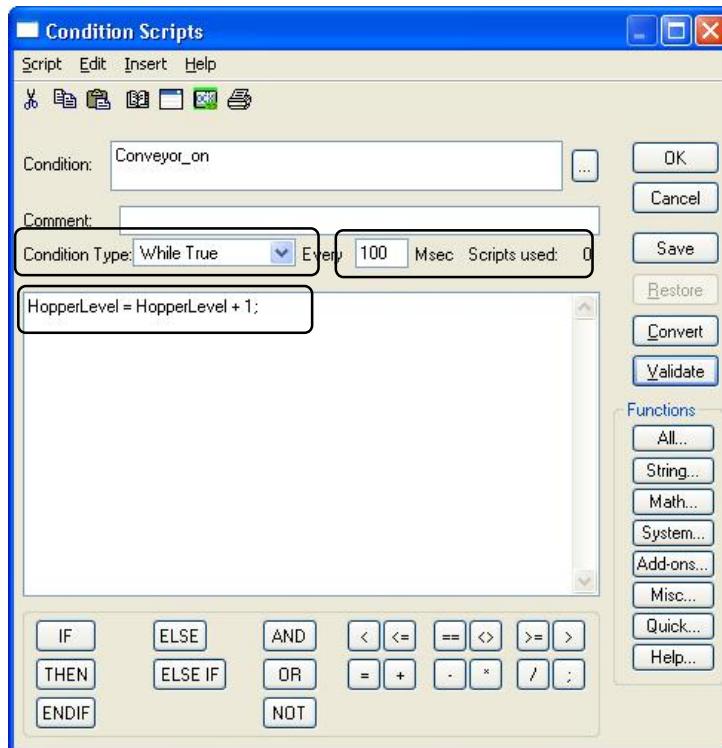
37. Scripts / Condition 항목을 오른쪽 마우스-클릭을 한 뒤, New 버튼을 선택합니다.



Condition Scripts Editor 가 나타납니다.

38. 새로운 조건으로 Conveyor_on 을 입력합니다.

39. 아래 그림과 같이 Condition Script 를 설정합니다.

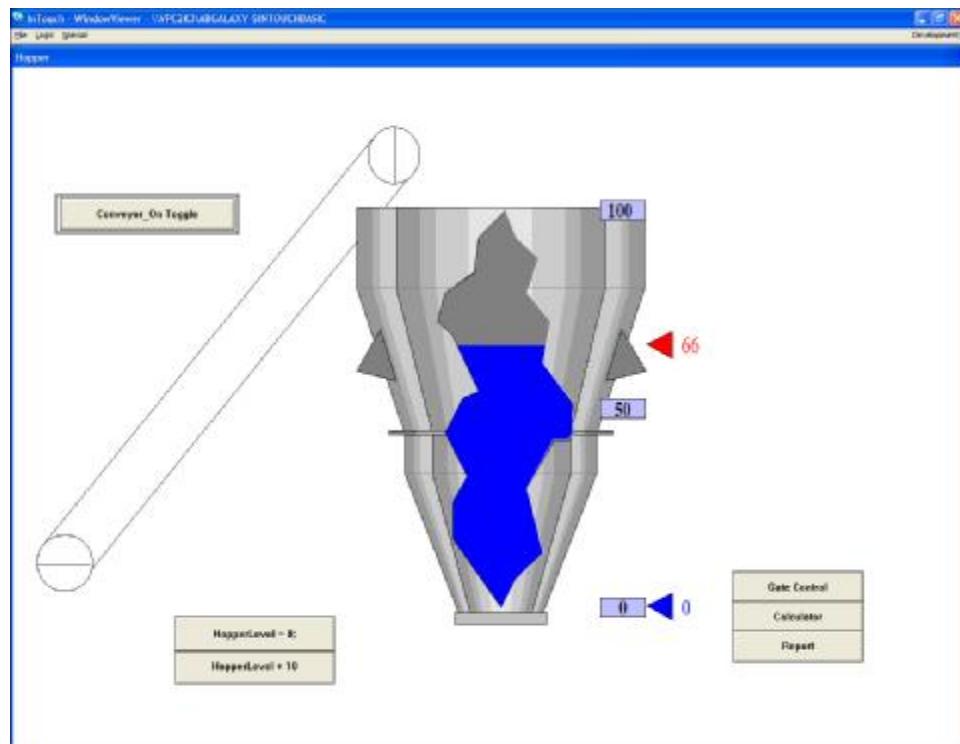


40. Validate 버튼을 클릭합니다.

41. Condition script 를 저장하기 위해 OK 버튼을 클릭합니다.

42. Runtime 모드로 전환하고, script 를 test 하기 위해 Conveyor_On Toggle 버튼을 클릭합니다.

Conveyor_On Toggle 버튼을 클릭 했을 때, conveyor image 는 움직이며, 색상을 채우는 level 은 변경됩니다. 이 색상을 채우는 level 을 재설정하기 위해서는 **HopperLevel = 0;** 버튼을 사용합니다.

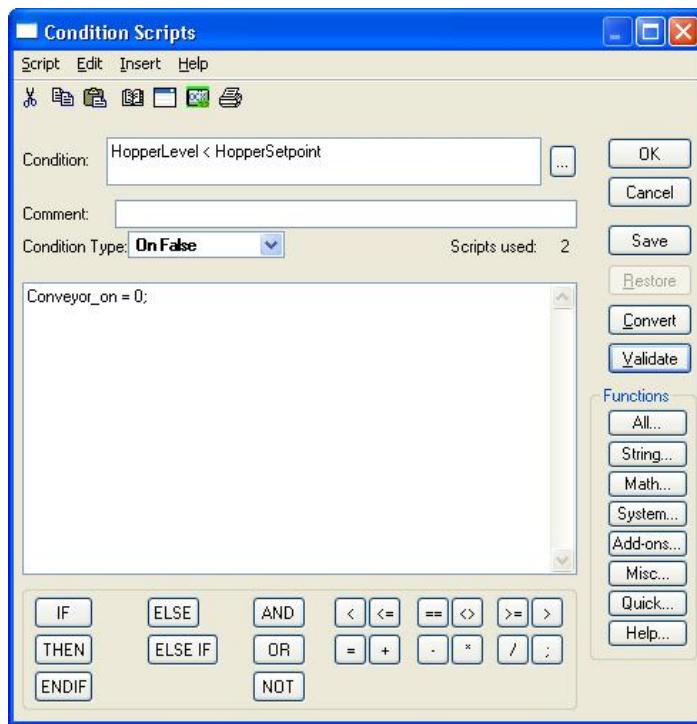
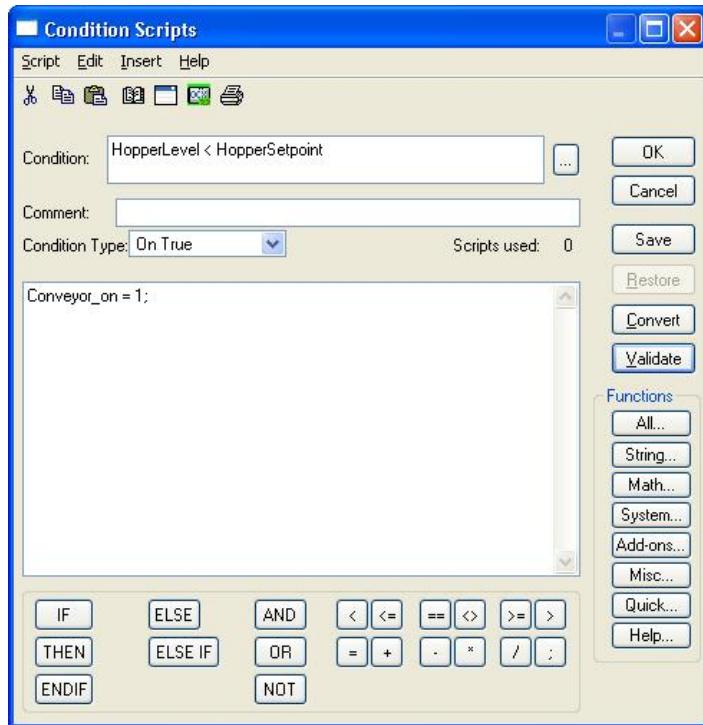


Conveyor 는 HopperLevel 값이 **HopperSetpoint** 의 값보다 적을 때 자동적으로 작동되며, HopperLevel 의 값이 HopperSetpoint 의 값과 같거나 클 경우 정지합니다.

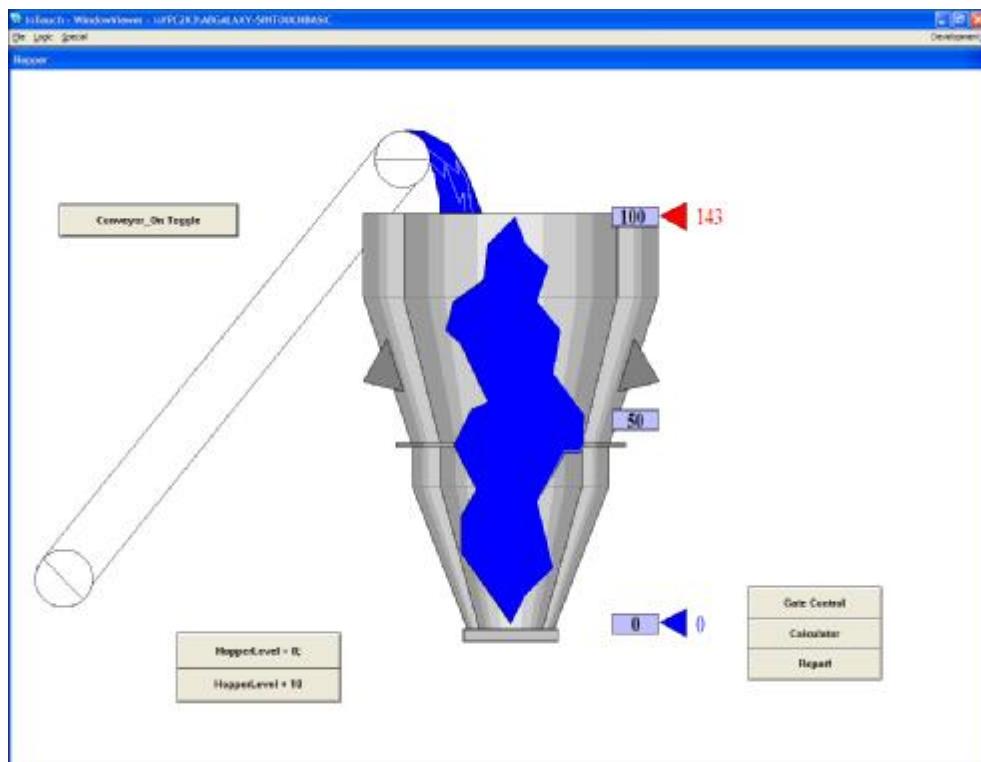
Condition script 의 조건인 **On True** 와 **On False** 설정 :

Conveyor 의 On / Off 상태는 **HopperLevel < HopperSetpoint** 로 조정 할 수 있습니다.

43. 아래의 그림에서 보여지는 것처럼 **On True**, **On False** Condition type 을 새로운 Condition Script 에 입력하고 생성합니다. 설정 후에는 **Validate** 버튼을 클릭합니다.



현재 Conveyor 가 작동하고 있다면, HopperLevel 은 계속 증가할 것이며, level 은 100 을 초과 할 것입니다.



44. 100 을 초과하여 증가하는 **HopperLevel tag** 를 유지하기 위한 condition script 를 생성하거나 수정합니다.

Hint : 새로운 Condition script 를 생성하기 보다는 기존의 script 를 수정합니다.

45. **Runtime** 으로 전환하고 script 를 테스트합니다.

Conveyor 는 **HopperLevel** 이 100 에 도달 했을 때, 멈춰져야 합니다.

Create the GATE CONTROL Window Script

Hopper 윈도우는 **GATE CONTROL** 버튼을 포함하고 있습니다. 버튼이 클릭 되었을 때, **Gate tag** 는 True / False 로 바뀌게 되지만, 현재 이 시점에서는 어떠한 일도 발생하지 않을 것입니다.

Hopper 의 하단에 위치한 Gate 그래픽은 Memory Integer tag 인 **GateCount** 가 할당된 horizontal location link 를 가지고 있습니다.

GateCount 가 0 일 경우, gate 는 닫히게 될 것입니다.

GateCount 가 30 일 경우, gate 는 열리게 될 것입니다.

Gate 가 참일 경우, **GateCount** 는 30 으로 증가시키는 script, **Gate** 가 거짓일 경우, **GateCount** 를 0 으로 감소시키는 script 를 생성하는 것이 필요합니다.

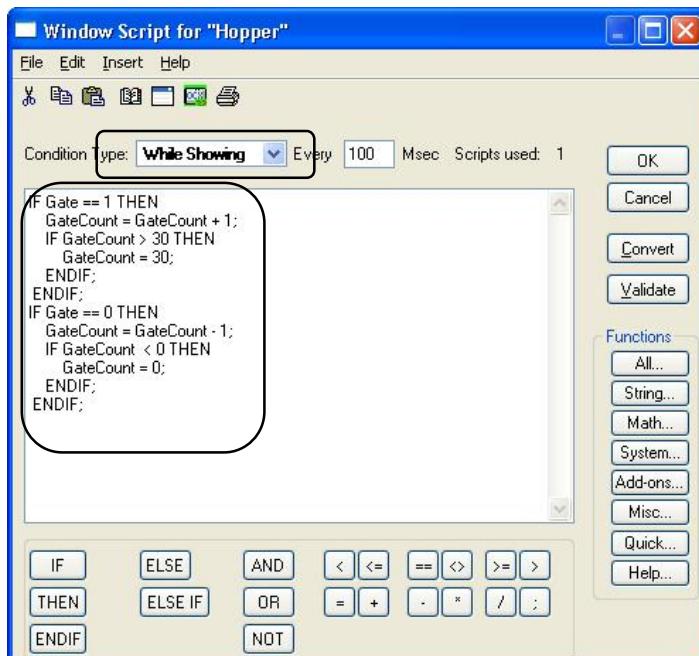
Hopper 윈도우가 열려 있을 경우에만 hopper 를 볼 수 있기 때문에, **While Showing window** script 입력이 필요합니다.

46. Application Explorer에서 **Windows** 아이콘을 펼칩니다.

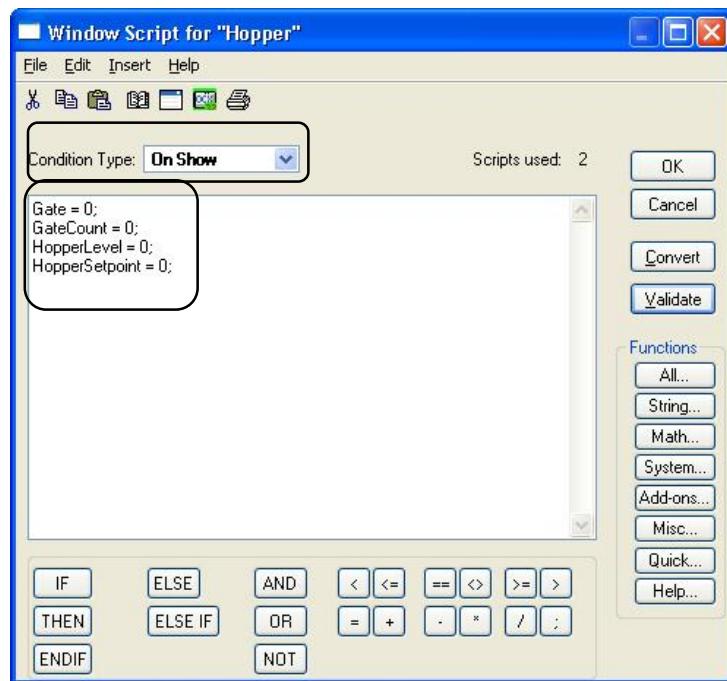
47. **Hopper** 항목을 오른쪽-마우스 클릭을 한 뒤, **Window Scripts** 를 선택합니다.



48. 아래 그림과 같이 script 를 입력합니다.



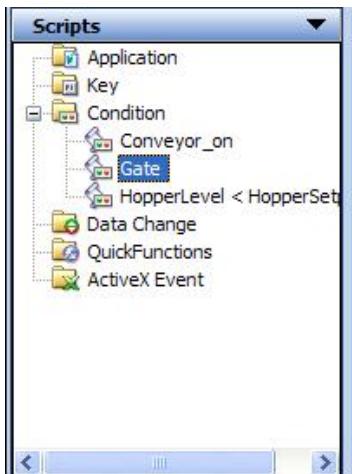
49. **Validate** 버튼을 클릭함으로써, script의 유효성을 검사합니다.
50. 두 번째 condition type을 설정하기 위해서, 아래 그림에서 보여지는 것과 같이 script를 편집합니다.



51. **Validate** 버튼을 클릭하고, 윈도우 script editor를 종료하기 위해 **OK** 버튼을 클릭합니다.
52. **Runtime**으로 전환하고 script를 테스트합니다. 윈도우가 열릴 때, 모든 level들과 값들은 모두 0으로 설정되어야 합니다.
53. **Gate Control** 버튼을 클릭합니다.
 Gate 그래픽에 슬라이드가 열리고 닫히는 것이 보여야 합니다.
HopperLevel은 gate가 열릴 경우에는, 감소하지 않습니다.
54. **Runtime** 모드로 전환합니다.
55. **HopperLevel**의 값을 0으로 감소시키기 위해 script를 사용하고, gate가 열릴 경우에는, 그대로 행동을 멈추고 기다립니다.

Hint : GateCount tag에 script를 사용하지 않습니다.

다음그림은 최적의 해결방법을 위한 hint입니다.



56. **Runtime** 으로 전환하고 script 를 테스트합니다.

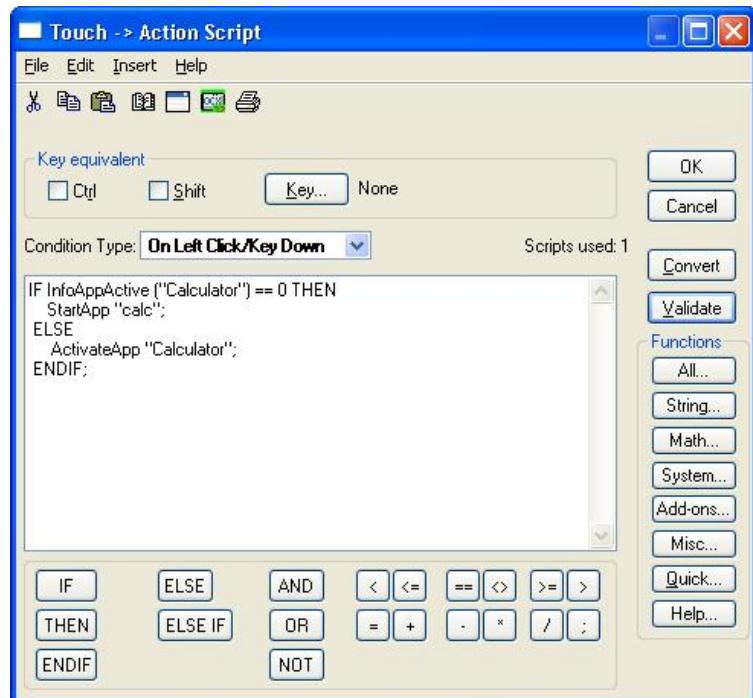
- | Conveyor 는 **HopperLevel** 이 **100** 에 도달했을 때, 멈춰야 합니다.
- | Gate 가 열릴 경우, level 은 감소해야 합니다.
- | Gate 가 왼쪽으로 열릴 경우, **HopperLevel** 은 **0** 으로 감소하고, gate 는 자동적으로 닫혀야 합니다.
- | **HopperLevel** 이 **0** 에 도달하기 전에 gate 를 닫는 것은 level 값의 감소를 멈추는 것입니다.

Use QuickScript Functions

InTouch 는 다른 프로그램들과 함께 작동하기 위한 기능을 가지고 있으며, 프로그램들이 작동하는지 아닌지를 확인 할 수 있는 기능을 가지고 있습니다.

Calculator라는 프로그램이 작동하고 있는지를 확인하기 위해서, 아래의 script 에서는 **InfoAppActive**라는 QuickScript 함수를 사용하고 있습니다. 만일, **Calculator** 프로그램이 실행되고 있지 않다면, **StartApp** 함수가 프로그램을 시작하게 할 것입니다. 반대로 **Calculator** 프로그램이 실행하고 있다면, **ActivateApp** 함수가 프로그램을 화면의 최상위에 불러 올 것입니다.

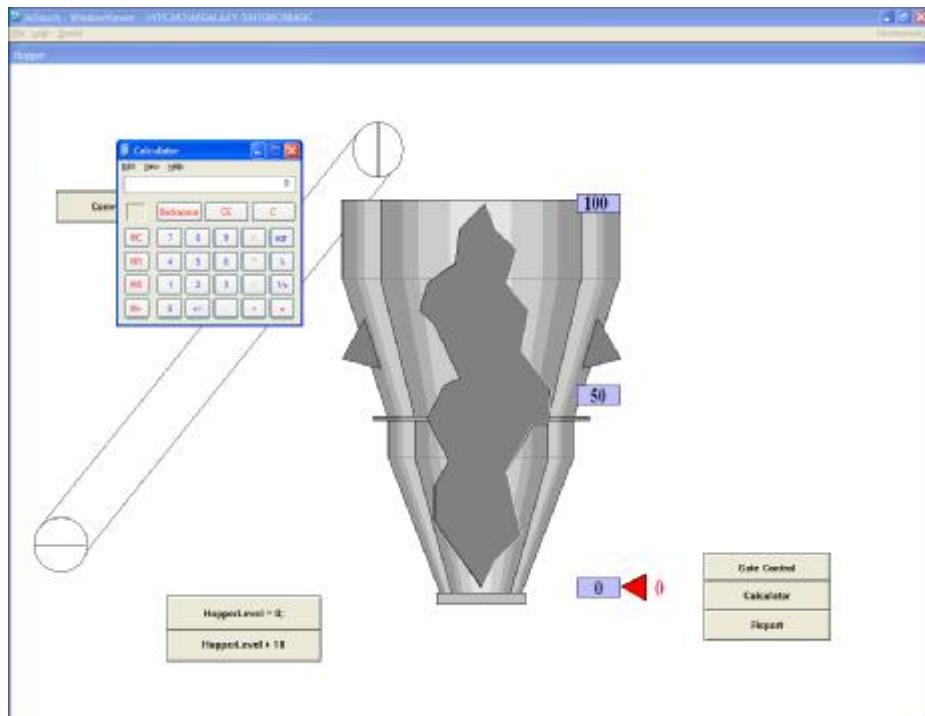
57. Hopper 윈도우에서 Calculator 버튼을 선택하고, 다음에 그림에서 보여지는 것처럼 Touch Action Script 를 생성합니다.



58. Script의 유효성을 검사하고, **Runtime**으로 전환합니다.

59. Calculator 버튼을 클릭합니다.

만일, Windows Calculator 가 실행하고 있지 않다면, 화면 위에 나타나게 될 것입니다.
그러나 만일, Windows Calculator 가 실행은 되고 있지만, 화면의 최상위에서 실행되고
있지 않을 경우에는, 화면의 최상위에 나타나게 됩니다.



60. Calculator 를 닫고, **Development** 모드로 전환합니다.

Write a Custom Report Script

REPORT 버튼이 클릭 될 경우, HopperLevel 의 상태 보고서(기록)를 볼 수 있습니다.

이 보고서는 다음과 같은 것들이 필요합니다 :

- | 버튼을 클릭된 날짜와 시간
- | Conveyor 와 gate 의 상태
- | 현재의 HopperLevel 과 HopperSetpoint 의 흐름

61. **REPORT** 라는 버튼의 **ACTION** script 에 다음의 내용을 입력합니다.

DIM ConveyorStatus AS MESSAGE;

```
IF Conveyor_on == 1 THEN
    ConveyorStatus = "Conveyor Running";
ELSE
    ConveyorStatus = "Conveyor Stopped";
ENDIF;
```

DIM GateStatus AS MESSAGE;

```
IF Gate == 1 THEN
    GateStatus = "Gate is Open";
ELSE
    GateStatus = "Gate is Closed";
ENDIF;

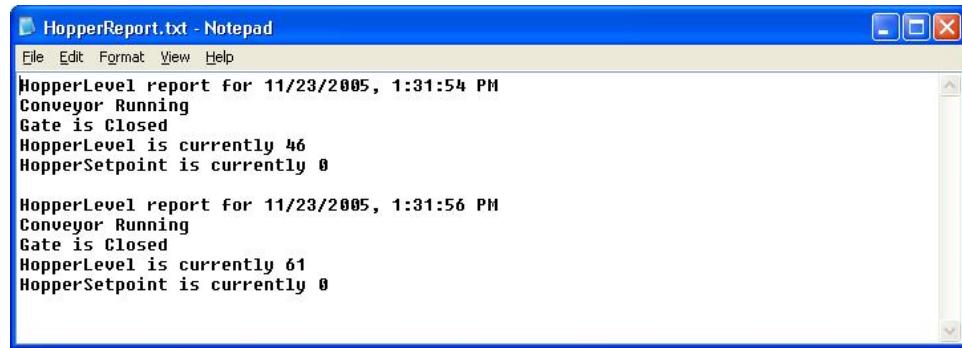
DIM ReportHeader AS MESSAGE;
ReportHeader = "HopperLevel report for " + $DateString + ", " + $TimeString;

DIM ReportLine1 AS MESSAGE;
ReportLine1 = Text(HopperLevel, "HopperLevel is currently #");

DIM ReportLine2 AS MESSAGE;
ReportLine2 = Text(HopperSetpoint, "HopperSetpoint is currently #");

FileWriteMessage( "C:\HopperReport.txt", -1, ReportHeader, 1);
FileWriteMessage( "C:\HopperReport.txt", -1, ConveyorStatus, 1);
FileWriteMessage( "C:\HopperReport.txt", -1, GateStatus, 1);
FileWriteMessage( "C:\HopperReport.txt", -1, ReportLine1, 1);
FileWriteMessage( "C:\HopperReport.txt", -1, ReportLine2, 1);
FileWriteMessage( "C:\HopperReport.txt", -1, " ", 1);
```

62. **Runtime** 으로 전환하고, **Report** 버튼을 클릭합니다.
63. 메모장을 사용하여 특정한 위치에 있는 **HopperReport** 파일을 엽니다.

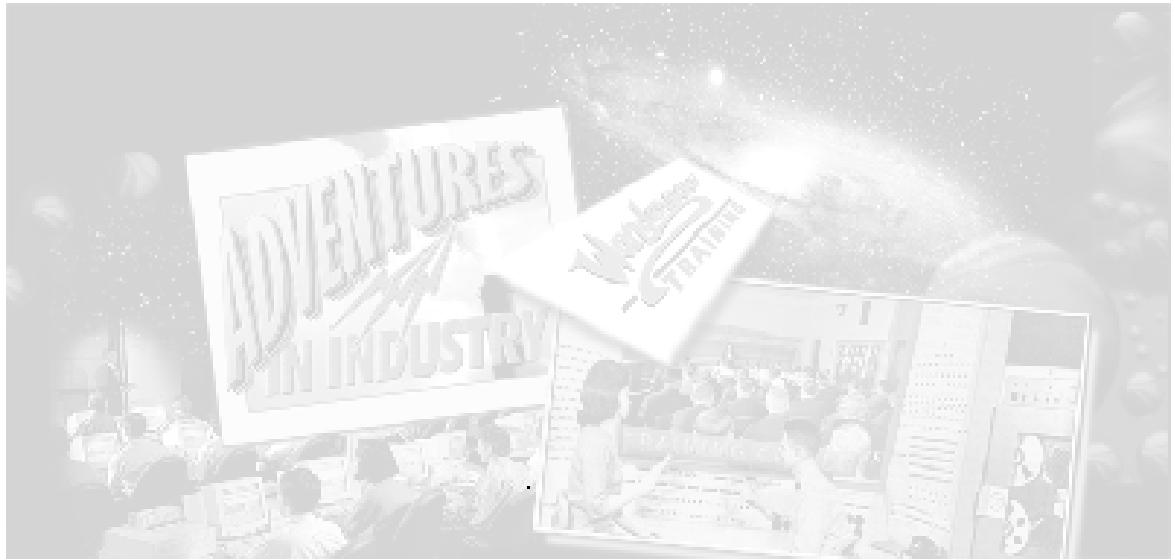


The screenshot shows a Windows Notepad window titled "HopperReport.txt - Notepad". The window contains two sets of text logs. The first log is dated 11/23/2005, 1:31:54 PM, and the second is dated 11/23/2005, 1:31:56 PM. Both logs report on conveyor status, gate status, current hopper level, and current hopper setpoint.

```
HopperLevel report for 11/23/2005, 1:31:54 PM
Conveyor Running
Gate is Closed
HopperLevel is currently 46
HopperSetpoint is currently 0

HopperLevel report for 11/23/2005, 1:31:56 PM
Conveyor Running
Gate is Closed
HopperLevel is currently 61
HopperSetpoint is currently 0
```

64. 파일을 닫고, Development 모드로 전환합니다.



Module 6

Alarms

Section 1 – Alarms and Events

Section 2 – The Distributed Alarm Objective

Lab 7 – Creating Alarms and Events

Section 1 – Alarms and Events

소개

InTouch는 HMI 프로세스의 상황을 운전자에게 알려 주기 위해 알람과 이벤트, 2 가지 방법을 가지고 있습니다.

알람은 공정중에 발생되는 이상 원인이 될 수 있는 프로세스들의 경고를 말하며 Operotor의 응답을 필요로 합니다.

아날로그 값 같은 데이터값이 사용자가 정의한 최대 한계 경계를 초과 했을 때 알람을 발생 시킵니다. 이 기능은 발생한 문제를 Operotor에게 통지하기 위해 현재 상태를 알람 **비확인** 상태로 변경 합니다. 이렇게 상태가 변경이 되면 InTouch는 이 알람을 데이터베이스와 파일에 기록하고 프린터로 인쇄할 수 있습니다. 운전자가 알람에 응답하면, 시스템은 알람 확인 상태로 돌아갑니다. 만일, 알람이 발생된 상황이 자체 해제된다 해도 InTouch는 알람 확인 상태를 계속 요구하게 됩니다.

태그가 Tagname dictionary에 정의 되어있는 동안, 이벤트를 모니터링 할 수 있도록 구성 되어 질 수 있습니다. 이벤트 메시지는 태그명의 값이 바뀔 때 마다, 알람 시스템에 기록 되어 집니다. 그 이벤트 메시지는 어떻게 값이 바뀌었고, 값의 변화가 운전자에 의해서, I/O, 스크립트 또는 시스템에 의해서 시작 되었는지를 기록 합니다.

알람구성

특정한 시스템이나 사용되거나 시작하는 방법에 상관없이, 일반적으로 적용되는 알람의 용어나 개념이 몇 가지 있습니다.

Alarms : 일반적으로 알람이란 임의의 상황에서의 특정한 유형입니다. 특히, 알람은 이상 있는 상황을 말하는 것입니다.

알람은 어떠한 상황이 잘못 되었거나, 프로세스가 어떠한 특별한 단계에 이르렀음을 알리는 하나의 신호입니다. 예를 들면, 보일러가 안전한 온도한계치 이상을 넘겼을 때 알람이 발생합니다.

Priority : 알람과 연관된 상황이나 환경이 얼마나 중요한지를 표현하는 우선순위를 말합니다.

보일러 온도 한계치의 경우, 아주 심각한 상황에서는 생명이나 재산을 지키기 위해 알람에 대한 즉각적인 대처가 필요로 하게 되며, 반대로 그다지 중요하지 않은 상황에서는 최소의 주의만 요구되기도 합니다.

알람의 경고는 상황에 따라 달라집니다. 이를테면, 공장 어플리케이션, 장비의 성능, 안정성, 백업시스템의 가용성, 손실에 대한 잠재적 비용, 작업 중단 시간 등등에 의해 설정이 달라집니다.

Sub-states : 알람 조건은 **sub-conditions**를 가지고 있습니다. 이는 **multi-state** 알람이라고도 합니다.

예를 들면, 아날로그 알람은 여러 개의 한계치를 가지고 있습니다. **High**나 **Low**한계치 같은 일반적인 운전상의 범위 뿐만 아니라 **HiHi**나 **LoLo**한계치 같이 좀 더 강도 높은 단계를 가지고 있습니다.

위에서 언급된 보일러 온도 단계는 이러한 **sub-state**들 중 하나의 조건에 만족될 시에 알람이 발생하게 됩니다. 또한, 전체적인 알람 상황이 확인되지 않은 상황에서, 그대로 알람이 남아있다면, 두개의 **sub-state**들 사이의 이동 역시 가능합니다.

Event : 이벤트는 시스템이 감시할 수 있는 사건발생이라고 말 할 수 있는데, 그 사건은 알람 발생과 상관이 있을 수도, 없을 수도 있습니다.

알람 상태로 변하거나 반대의 경우 역시 이벤트의 한 종류라고 할 수 있습니다. 또한, 이벤트는 운전자의 조치나 시스템 설정의 변경사항, 시스템 오류를 기록합니다.

Acknowledgment : 알람의 주된 목적은 알람 발생 상황을 누군가가 수정하게 하기 위함입니다. 각 개인이나 시스템은 알람을 확인하고, 평소 상태로 유지시켜야 합니다.

알람확인은 지금 당장 발생하지 않을지도 모르는 상황에 대해서 정확한 조치를 취하는 것과는 다릅니다.. 또한, 알람 발생 상황이 외부에서의 조정이나 시스템 자체적으로 평소 상태로 복귀하게 하는 것과도 다릅니다.

알람확인은 단순하게 누군가가 알람 발생 사실을 인지했다는 것을 나타낼 뿐입니다. 상위의, 혹은 중간 정도의 우선순위 알람은 대개는 알람확인을 요구합니다. 하지만, 아주 낮은 우선순위의 알람은 그렇지 않습니다.

비록 알람이 해제되어도, 알람이 확인될 때까지 그 알람은 자신이 해제되었다고 인식하지 않습니다.

Groups : 알람들은 관리면에서나 알람 주제를 용이하게 하기 위해서 그룹으로 구성할 수 있습니다. 이런 그룹들은 공장의 각기 다른 공정이나 장비의 부품, 운전자의 책임 구역, 특정한 기능별로 나눌 수가 있습니다. 또한, 각 그룹들은 계층적으로 부모 그룹과 자식 그룹으로 구성할 수도 있습니다.

Areas : 공장들은 일반적으로 물리적인 지역이나 운전자의 책임구역, 공정 처리의 단계, 장비의 종류 등과 같이 편성되어 있습니다. 그룹과 마찬가지로 각 구역들은 계층적으로 부모구역, 자식구역으로 나눌 수 있습니다.

구역이 그룹과 개별적으로 언급된 이유는 어떤 공장들은 공장 사이간에 구분을 하기 때문입니다.

Note : 분산 알람 시스템은 구역들에 대한 뚜렷한 지원기능을 볼 수 없습니다. 그러나 알람 그룹들은 구역에 해당하는 집합들로 알람들을 분산시킬 수가 있습니다.

알람유형

InTouch는 그 특성에 따라 몇 가지 일반적인 분류로 알람을 나눕니다. 이러한 분류는 클래스(Class)와 유형(Type)으로 나눌 수 있습니다.

표준 알람 시스템은 모든 알람을 이산, 편차, 변경율, 값, 그리고 SPC 등 다섯 개의 일반 유형들로 나눕니다.

Alarm Condition	Distributed Class	Distributed Type
-----------------	-------------------	------------------

Discrete	DSC	DSC
Deviation – Major	DEV	MAJDEV
Deviation – Minor	DEV	MINDEV
Rate-of-Change	ROC	ROC
SPC	SPC	SPC
Value – LoLo	VALUE	LOLO
Value – Low	VALUE	LO
Value – High	VALUE	HI
Value – HiHi	VALUE	HIHI

각 알람은 **InTouch** 태그와 연관이 있습니다. 태그의 유형에 따라 한 개 이상의 알람 종류나 유형들이 구성될 수 있습니다.

알람 조건들은 태그명 사전에서 정의 되어 집니다.

이벤트 유형

InTouch는 그 특성에 따라 몇 가지 일반적인 분류로 이벤트를 나눕니다. 이러한 분류는 이벤트 유형(Event Type)으로 나눌 수 있습니다.

아래 테이블은 이벤트의 특성으로 나눈 목록입니다.

Event	Condition
ACK	알람이 확인되었습니다.
ALM	알람이 발생했습니다.
EVT	알람 이벤트가 발생했습니다.
RTN	알람상태에서 정상상태로 복귀되었습니다.
SYS	시스템 이벤트가 발생했습니다.
USER	Operator가 바뀌었습니다.
I/O	DDE client로부터 값이 들어왔습니다.
LGC	QuickScript에 의해 태그값이 변경되었습니다.
OPR	Operator가 입력을 통해 태그 값을 변경시켰습니다.

위에서 6개의 이벤트는 이벤트 로그가 가능한 상태에서는 자동으로 구성됩니다. 나머지 3개는 Tagname dictionary에서 각각의 태그이름에 따라 사용자가 정의해야 됩니다.

태그 알람 설정

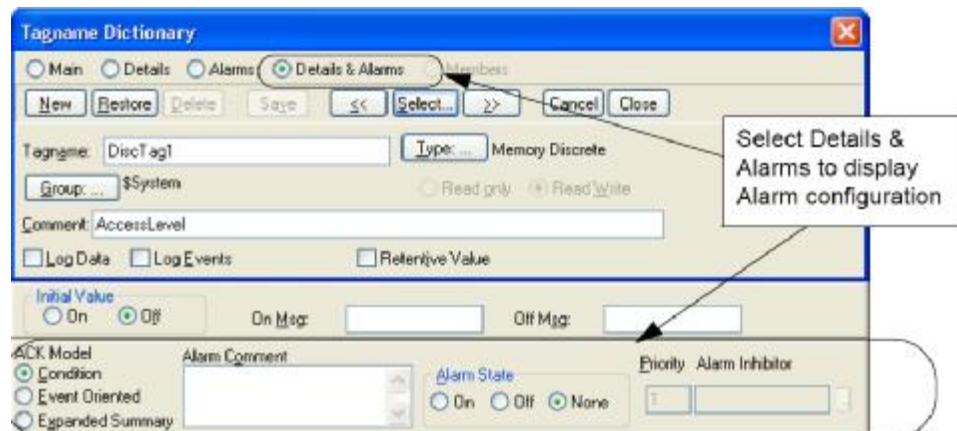
Tagname dictionary에 정의된 각 태그에 InTouch는 알람 정의와 설정을 제공합니다. 기본적인 개념은 어떤 태그든지 **알람 가능상태**로 구성할 수 있으며, 사용자가 지정한 경계값과 알람의 탑입을 설정할 수 있습니다. 이런 태그들은 태그의 데이터값이 바뀔 때마다 **alarm logic**을 실행 시킵니다.

Alarm logic은 InTouch 내부 서브루틴으로 존재함으로써, 알람의 유형을 체크하며, 갱신된 값을 경계치와 비교하고, 태그들에 알람이 발생된 상황인지 아닌지를 판단하는 역할을 합니다. 어떠한 상태 변화든 분산 알람 시스템에 기록되어 남게 됩니다.

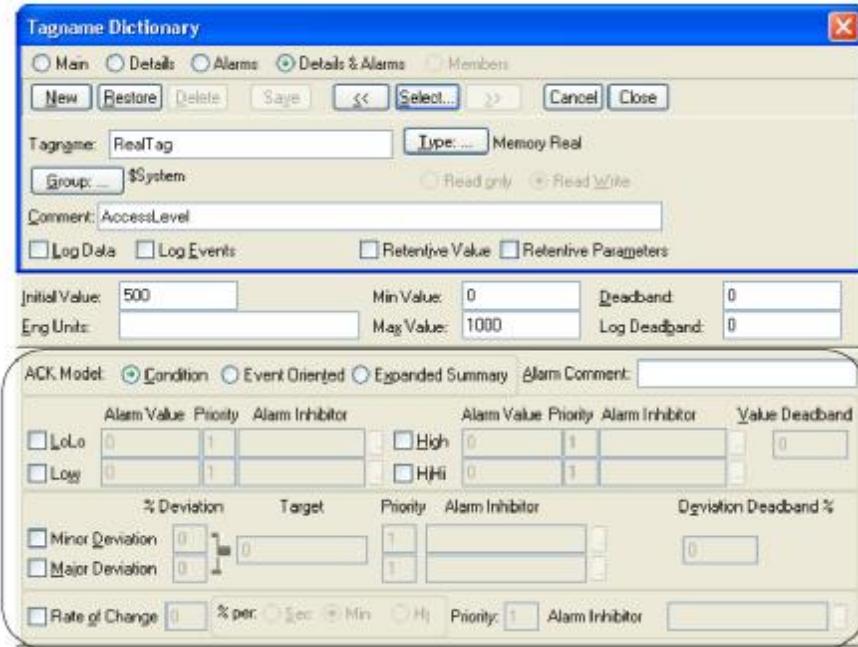
InTouch는 기본적으로 두 개의 알람 탑입으로 정의 하고 있으며, 각각 여러 개의 하위 탑입으로 세분화하고 있습니다.:

이산형: 이산형 알람은 이산형 태그에 사용됩니다. 알람 조건은 이산형 태그의 값이 참 혹은 거짓인 상태에 따라 달라지며, 알람의 우선순위를 적용합니다.

이력 알람 설정 창은 아래 그림과 같습니다:



아래의 그림은 아날로그 알람 태그의 설정 창입니다.



- | **아날로그형:** 아날로그 알람은 정수형 과 실수형(부동 소수점) 태그에 적용됩니다.
아날로그 유형에는 몇몇의 알람 부류가 있습니다.
- | **Value:** 현재 값은 하나 혹은 그 이상의 경계값과 비교 됩니다. 만약 경계값을 초과했다면 알람이 발생하며, 경계값과 우선순위, 알람 경계값의 사용여부를 개별("LoLo" limit, "Lo" limit, "Hi" limit and "HiHi" limit)적으로 구성할 수 있습니다.
- | **Deviation:** 현재값과 목표값을 비교합니다. 그 후, 그 편차값은 하나 혹은 그 이상의 경계값과 비교(태그값 범위에서 퍼센트로 표현)를 하게 됩니다. - 즉, 설정한 허용범위 안에서의 총량의 편차. "Major Deviation"과 "Minor Deviation"의 경계값, 우선순위, 알람 허용 여부 등을 개별적으로 구성할 수 있으며, 태그값 범위에서 비율로 표시되는 "Deviation Deadband"도 구성할 수 있습니다. 알람 상태에서 정상상태로 복귀 하기 위해서는, 여기서 정한 비율 아래로 떨어져야만 합니다.
- | **Rate-of-Change:** 현재 값과 이전 값이 현재시각과 그 이전에 갱신된 시각과 함께 계산됩니다. 만약 절대값의 변화율이 경계값을 초과하면 알람 상태가 됩니다. ROC의 경계값과 우선순위 알람 발생의 사용 여부를 설정 할 수 있습니다.

경계값은 태그 값의 범위에서 비율로 표현됩니다. – 시간값은 초마다, 분마다, 시간마다 설정 가능합니다.

알람 우선순위

InTouch에서 구성된 각각 알람은 우선 순위 값을 반드시 하나씩은 가져야 합니다. 이 값은 일반적으로 알람의 중요도를 나타내며 1을 최대치로 하고 1부터 999 까지 1000 개의 레벨 범위를 가지게 됩니다.

이 우선권을 이용해서 알람 범위를 작성하고 각각에 알람을 지정해서 각 알람의 종류와 위험도를 쉽게 구분할 수 있으며, 애니메이션 링크, 스크립트, 그리고 태그 필터와 인쇄 순서를 우선 순위 를 이용하여 작성할 수 있습니다.

예를 들어, 임의의 프로세스를 가지는 공장에서는 아래와 같이 4개의 단계로 범위를 정하고 우선순위를 정하여 사용 할 수도 있습니다.

Alarm Severity	중요도 범위
Critical	1 – 249
Major	250 – 499
Minor	500 – 749
Informational	750 – 999

공장의 엔지니어들이 **InTouch**에서 태그와 알람 조건들을 작성 할 때 각 알람은 그 우선 순위의 범위 안에서 순위를 결정함으로, 각 알람은 이 범위의 레벨을 표시 할 수 있습니다.

구성된 이 범위를 참조하여 공장 operator는 수비개 알람을 인지 할 수 있고 특정한 레벨의 알람만을 인쇄 할 수 있습니다.

알람 그룹

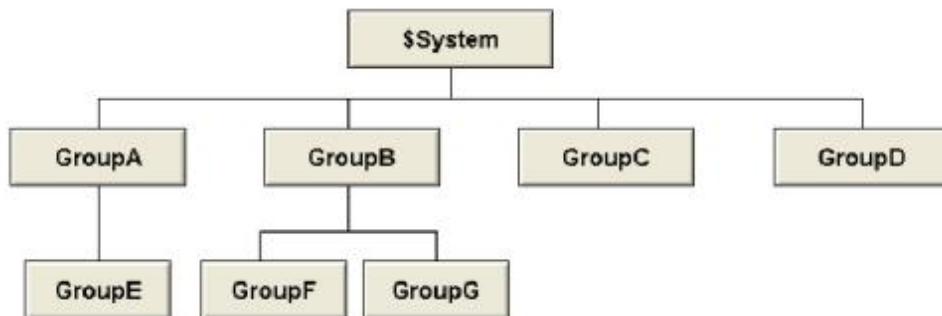
각 **InTouch** 알람은 논리 알람 그룹을 정의 할 수 있습니다. 이 알람 그룹은 사용자가 유일한 이름을 정의할 수 있으며, 레벨당 총 32개의 그룹을 지원합니다. 이 그룹들은 조직, 공정 라인, 혹은 사용자가 생각하는 그룹 단위로 알람을 분류하는 방법을 제공합니다. 알람 그룹들은 알람 디스플레이, 알람 인쇄, 확인 스크립트를 필터링하기 용이합니다.

모든 태그는 알람 그룹과 관련이 있습니다. 만일 알람 그룹명을 지정하지 않으면 태그는 자동적으로 최상위 그룹 "\$System"이 입력 됩니다. 어떤 알람 그룹이든 태그명과 알람 그룹명 두 가지를 가질 수 있습니다.

알람 그룹은 트리의 맨 위쪽에 있는 루트 그룹인 \$System이 있는 순위적 트리 구조로 조직되어 있으며, 정의된 모든 알람 그룹은 자동적으로 루트 그룹 밑에 놓이게 됩니다.

이 트리는 최고 32 레벨까지 있습니다. 각 알람 그룹은 최대 32개의 하위 그룹을 보유 할 수 있으며, 각 하위 그룹은 최대 32개의 하위 그룹이 있는 등, 32 단계에 도달 할 때까지 계속 이어집니다.

이 그림은 단지 알람 그룹을 보여줄 뿐 각 그룹에 할당된 태그이름은 아닙니다. 이 트리 개념은 윈도우의 폴더 디렉토리 구조와 흡사합니다. 하나의 디렉토리가 다른 하위 디렉토리를 가질 수 있듯이, 그룹 역시 마찬가지이며, 파일이름은 태그명이라고 생각하면 될 것입니다.



알람 그룹의 이름은 알파벳 (A-Z, a-z)으로 시작하는 최대 31개의 문자까지 사용할 수 있습니다. 그 그룹의 이름의 문자 중간에 빈칸을 넣을 수 없습니다. 예를 들면, “AlarmGroup#1-23”d은 사용할 수 있는 그룹명이지만, “Alarm Group # 1 – 23”이라는 이름은 사용할 수 없습니다.

분산 알람 시스템은 또한 이 그룹들을 알람 그룹 목록을 기초로 합니다.

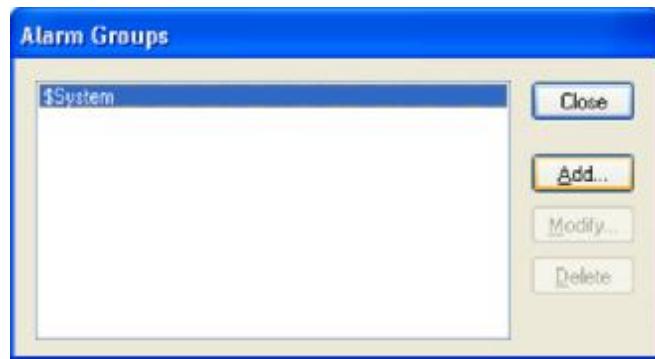
Note : InTouch 라이센스로는 알람 그룹은 태그의 개수로 세지 않습니다. 하지만 데이터베이스에서는 태그로 간주합니다. 그래서 알람 그룹과 실제 태그 수를 합쳐 61,405를 초과 할 수 없습니다.

또한, View가 실행되는 동안에는 그룹을 생성할 수 없습니다.

Warning : 많은 수의 알람 그룹은 (100's/1000's) 시스템 성능을 떨어뜨릴 수 있습니다.

알람 그룹 생성

- Windowmaker 메인메뉴의 Special / Alarm Groups를 클릭
Alarm Groups 창이 나타납니다.



Modify, Delete버튼은 알람그룹이 정의 되어 있지 않으면 활성화 되지 않습니다.
\$System 알람그룹은 수정하거나 삭제할 수 없다.

Note : Tagname dictionary에서 태그를 정의 할 때, 알람 그룹을 생성할 수 있으며, 그 태그와 알람그룹을 연계시킬 수 있습니다.

b. Add 버튼 클릭

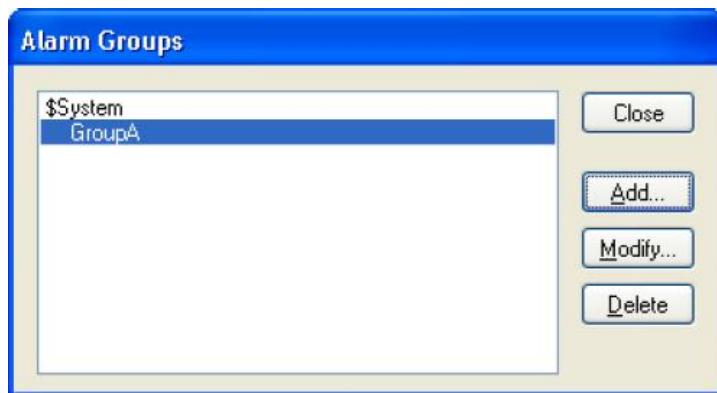
Add Alarm Group 창이 열립니다.



c. Group Name 상자에 새 알람 그룹 이름을 입력합니다. 처음 알람그룹은 자동적으로 Partner Group으로 \$System이 할당됩니다.

d. OK 버튼을 클릭

Alarm Groups 창이 열립니다:



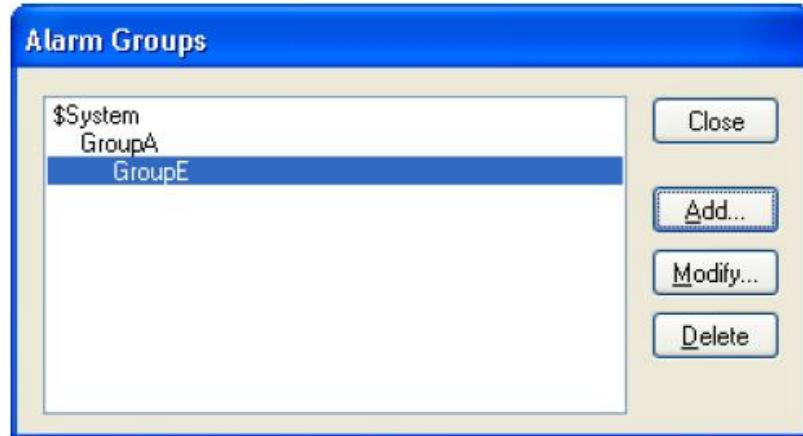
- e. 방금 생성한 알람 그룹을 선택하고, 두번째 그룹을 만들기 위해서 Add버튼을 클릭하면, 새로운 알람 그룹의 Partner Group에 방금 선택한 그룹이 입력되어 있는 것을 볼 수 있습니다.



- f. Comment 상자에는 생성할 알람그룹의 주석을 달아 놓습니다.

- g. OK 버튼을 클릭

Alarm Group 창에 알람그룹 계층이 생긴 것을 볼 수 있을 것입니다.



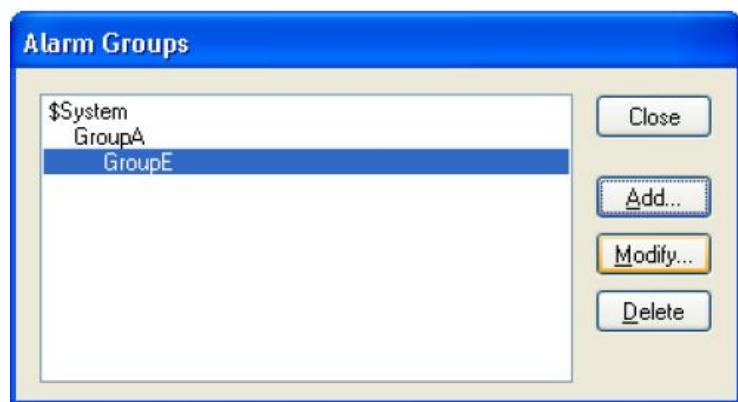
- h. Close 버튼 클릭

알람 그룹 설정

- a. Special / Alarm Group 선택합니다.
Alarm Group 창이 보일 것입니다

Note : Tagname dictionary에서 태그를 정의 할 경우에도, 알람그룹 설정은 가능합니다.

- b. 목록에서 수정하려는 알람 그룹을 선택하고, Modify버튼을 클릭합니다.

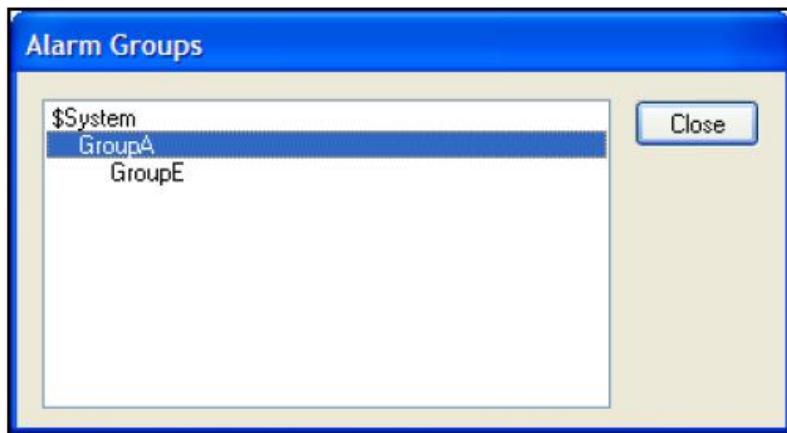


Modify Alarm Group 창이 나타납니다:



- c. 알람그룹의 partner group을 수정 하려면, Partner Group을 클릭합니다.

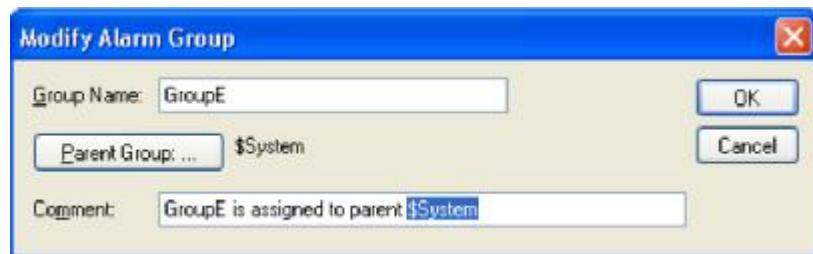
Alarm Group창이 나타납니다:



- d. 새 partner 그룹으로 \$System을 선택합니다.

Close를 클릭

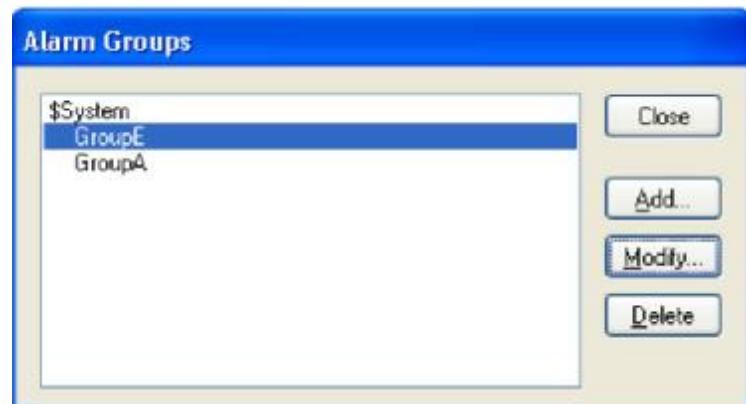
- e. Modify Alarm Group창이 나타나면, 새로운 partner 그룹이 보일 것이다.



- f. 원한다면, Comment를 입력합니다.

- g. OK를 클릭

알람 그룹들이 계층적으로 보이는 것을 알 수 있습니다:



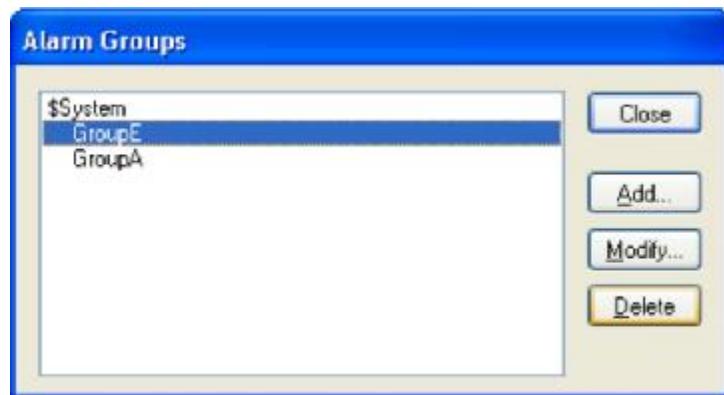
- h. Close 클릭.

알람 그룹 삭제

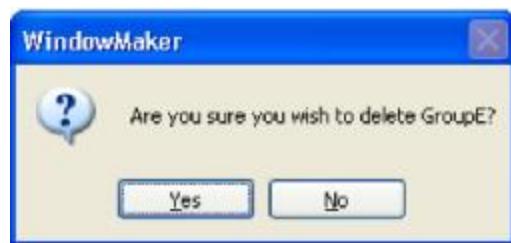
- a. Special / Alarm Groups을 선택

Alarm Group 창이 나타납니다.

- b. 삭제하려는 알람그룹을 선택합니다.



- c. Delete 버튼 클릭



- d. 확인 창에서 Yes버튼을 클릭

Note: Tagname dictionary에서 태그를 정의할 때, 알람그룹을 삭제할 수 있습니다.

알람 그룹들은 삭제하려는 그룹에서 사용되는 태그명하고는 상관없이 삭제할 수 있습니다. 삭제하려는 그룹의 태그들은 파트너 알람그룹으로 이동하게 됩니다.

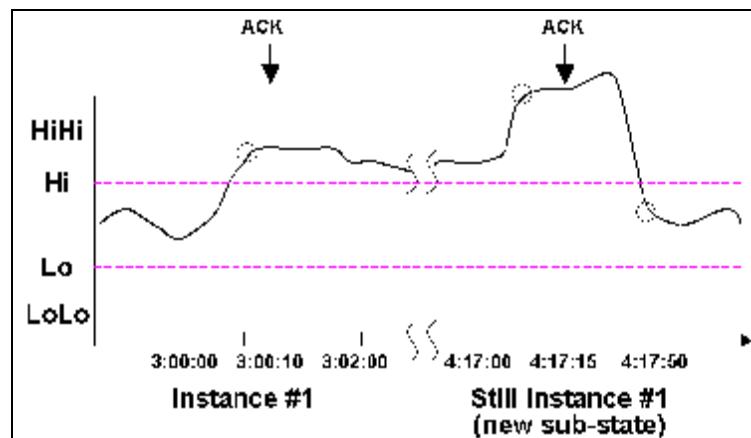
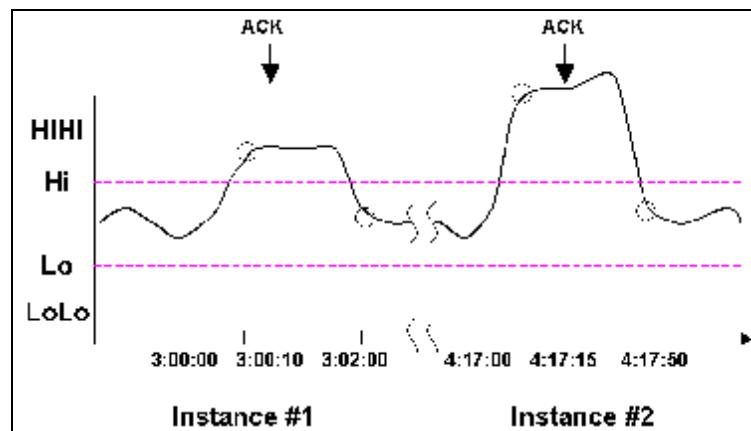
알람 확인 모델(Alarm Acknowledgment Models)

이번에는 알람이 어떻게 기록, 추적, 확인 되는지를 설명 할 것입니다.

Normal 상태에서 **Alarmed** 상태가 될 때마다, 새로운 알람이 생성됩니다.

만약 multi-state 알람인 경우, 이미 알람이 발생한 상태에서 새로운 **sub-state**로의 변경은 동일한 알람 요구(instance)의 변화로 취급합니다.

알람 지속시간(*lifetime*)은 알람이 정상상태로 돌아오면 멈춥니다. – 뒤따른 알람 상태로의 변경은 새로운 알람 상태를 발생시킵니다.



InTouch 분산 알람 시스템은 각각의 알람 요구를 추적합니다. – 처음으로 알람상태에 돌입했을 때, 하부 상태의 변화가 일어났을 때, 정상 상태로 되었으나 확인을 기다리고 있을 때, 마지막으로 확인되었을 때

- I **Alarms as conditions:** 알람 instance에 의해 ‘ACK 요구’가 생성됩니다. 처음 알람 상태가 되면 알람 instance는 ACK를 기다리기 시작합니다. 만약 ACK된 후 새로운 알람이 발생하면(예를 들어 “Hi”에서 “HiHi”로 변화), 또 다른 **ACK**를 기다리게 됩니다.

ACK를 받으면 지금까지 확인되지 않은 모든 Instance에 대해 **ACKed**된 것으로 간주합니다.

- I **Alarms as events:** 알람 instance에 의해 가장 최근의 알람 상태 혹은 하부 알람의 'ACK 요구'가 생성됩니다. 처음 알람 상태가 되면 알람 instance는 ACK를 기다리기 시작합니다. **ACKed** 한 후 변화에 의해 새로운 알람 하부 상태가 되면, 또 다른 ACK를 기다리게 됩니다. 연속된 변화에 각각의 일련번호가 부여되고, 응답 ACK 역시 일련번호가 부여됩니다. 거절된 ACK는 진단 용도로 기록됩니다. 다른 방법으로 추적은 불가능합니다. **ACK**가 받아들여지면 지금까지 **ACK**되지 않은 모든 변화가 **ACKed** 됩니다. **Event-oriented** 모델에서는 알람이 다른 상태로 바뀌더라도 **ACK**의 날짜 정보가 일치합니다. 대기 시간이 짧은 시스템에서는 **Condition-oriented** 알람과 비슷하게 보입니다. – 하지만 인터넷과 같이 대기 시간이 긴 시스템에서는 중요합니다.
- I **Expanded Summary Alarms:** 알람의 각 변동마다 '확인 요구'가 필요합니다. 처음 알람 상태가 되었을 때 반드시 **ACKed**되어야 합니다. 그리고 정상 상태로 복귀 되었을 때도 별도의 **ACKed**가 있어야 합니다.
- I 새로운 알람 하부 상태로의 어떠한 변화든 반드시 **ACKed**되어야 하며, **RTN** 역시 **ACKed**되어야 합니다. 이전 정상상태에서 처음 알람에 들어간 뒤의 하부상태 변화는 "**RTN group**,"에 속 한 걸로 간주합니다. 만약 정상으로 돌아오고 뒤이어 알람 상태로 다시 들어가게 되면 새로운 **RTN group**가 생성됩니다. 각각의 변화는 반드시 개별적으로 확인을 해야 합니다. 알람은 정상으로 돌아오고 모든 **RTN 그룹**이 확인되어야지만 **ACK**된것이 인정됩니다.

Note: 여기서 "summary"의 의미는 "확인을 기다림"이라는 뜻입니다. 이 알람 모델은 "ring-back alarms."으로도 불립니다.

요약하면, **condition-oriented** 알람은 확인이 발생한 이전 시간의 모든 알람을 확인합니다. **Event-oriented** 알람 (OPC의 경우와 비슷)의 경우 확인은 가장 최근에 "활성화"되어 있는 이벤트에 대해서 인용될 때 받아들여집니다. 그리고, **Expanded Summary** 알람의 경우 모든 이벤트 – "활성", "비활성", 다른 하부상태 – 가 각각 확인이 되어야지만 모든 알람이 확인된 것으로 인정합니다.

Expanded Summary 알람

알람이 발생하면 알람 디스플레이 개체에 알람 상태와 발생시간이 하나의 레코드로 나타납니다. 이 레코드는 조작자가 확인을 하고 정상상태로 돌아가기 전에는 없어지지 않습니다. 만약 조작자의 확인 전 정상상태로 돌아가면 두 줄의 레코드로 나타나게 됩니다.

예를 들어 보일러 온도가 Hi 경계값을 초과하면 알람이 발생하게 됩니다. 하지만, 조작자의 확인 전 정상온도로 돌아가게 되면 알람이 발생했다는 레코드와 알람이 확인되지 않은 상태에서 정상으로 돌아갔다는 레코드가 생기게 됩니다.

알람상태, 하부상태, 정상으로 돌아가는 등의 개별적 변화에 따라 각각 확인이 필요합니다. 정상상태에서의 각 변화는 새로운 RTN그룹을 만들게 된다. 하나의 RTN그룹이 확인상태가 되려면 RTN그룹에 속한 개별적인 알람이 모두 확인되어야 합니다.

Expanded Summary 알람 사용하기

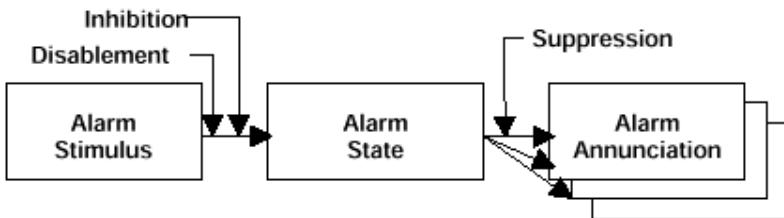
태그를 정의할 때 ACK 모델을 **Expanded Summary**로 선택하면 알람이 정상으로 바뀌더라도 조작자의 확인이 필요합니다. 알람을 확인하면 알람이 표시되는 색깔은 바뀌지만 발생시간표시는 바뀌지 않습니다. 알람은 확인되고 정상상태로 돌아갔을 때에만 디스플레이에서 사라집니다.

Note: 만약 **Expanded Summary** ACK 모델로 태그를 정의하면, **Alarm Properties** 대화상자의 **RTN Implies Ack** 옵션은 사용할 수 없습니다.

Alarm Visibility Controls

여러 이유로 어떤 알람을 구성변경 없이 ‘off’ 해야만 할 때가 있습니다. InTouch는 세 가지 기본적인 알람 인지 형태를 지원합니다: **disablement**, **inhibition**, and **suppression**. 이것을 세단계 알람 모델을 통해 설명하겠습니다..

- I **Alarm stimulus:** 알람가능한 아이템의 값 또는 상태를 추적합니다.
- I **Alarm state:** 알람가능한 아이템을 경계값과 상태와 비교하여 만약 알람 상태이면 상태를 변화시키고 그를 추적 저장합니다.
- I **Alarm annunciation:** 디스플레이 혹은 기록되고 있는 하나 혹은 여러 개의 알람 Client의 알람 상태를 저장 혹은 업데이트 시킵니다.



이 모델의 개념은 알람 공급자에서 보여지는 알람을 조정할 수 있습니다.:

- I **Alarm Disablement:** 플래그로 알람 불가를 표시할 수 있습니다. 알람 불가 상태에서는 알람 구성과 관련된 어떠한 변화도 있을 수 없습니다. 조건에 따라 알람 상태로 간주할 수 있겠지만 알람 불가 상태에서는 알람상태로 할 수 없습니다. 다시 말해 특정 아이템을 “강제 정상” 상태로 만드는 것입니다. **Disablement**는 **alarm stimulus** 와 **alarm state**의 연결을 끊는다.

알람 불가 상태에선 알람 상태가 될 수 없고, 그러므로 알람 이력도 남기지 않는다.

- I **Alarm Inhibition:** **inhibitor tag** 라 불리는 태그 값에 의해 특정 태그의 알람 금지가 결정됩니다. Inhibitor 태그 값이 FALSE (or zero)인 경우 알람이 정상적으로 동작합니다. 하지만, TRUE (or non-zero)인 경우에는 알람이 발생 할 수 없습니다. 본질적으로는 알람 **disabled**과 같습니다. **Alarm Inhibition** 는 두 단계의 과정을 가집니다:

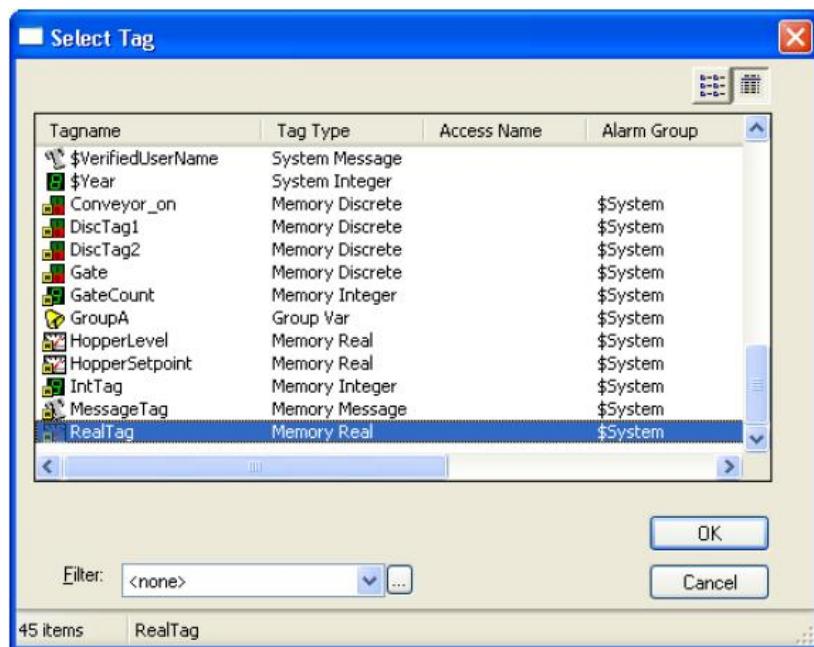
1. Inhibitor tag를 할당합니다.
2. FALSE에서 TRUE로 혹은 그 반대로 inhibitor 태그를 변화시킵니다.

Disablement가 그렇듯, **inhibition**도 **alarm stimulus** 와 **alarm state**의 연결을 끊습니다.

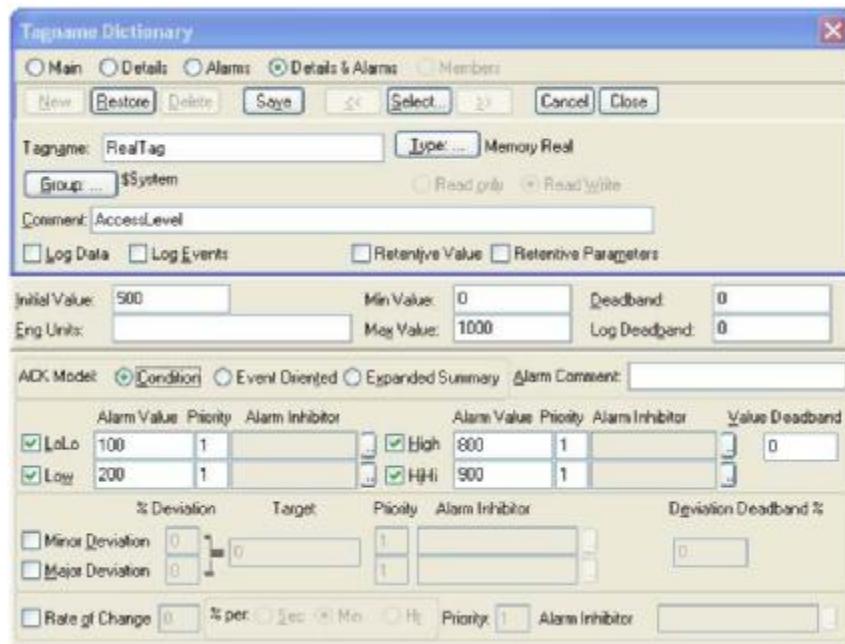
- I **Alarm Suppression:** 위 알람 모델에서 **alarm state**와 **alarm annunciation**의 연결을 끊는다. **Alarm Suppression**에 대해서는 InTouch Advanced course에서 설명드립니다.

알람 Inhibition 설정 구성

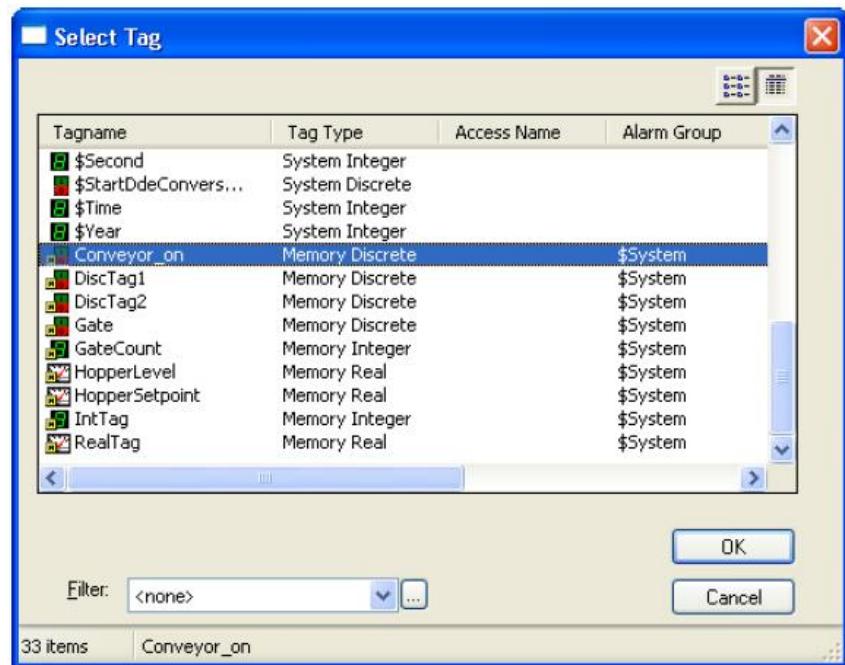
- Special / Tagname Dictionary를 엽니다.
- 기준에 정의된 태그를 하나 선택합니다.



- OK버튼을 클릭
- Details&Alarms 옵션을 선택합니다.



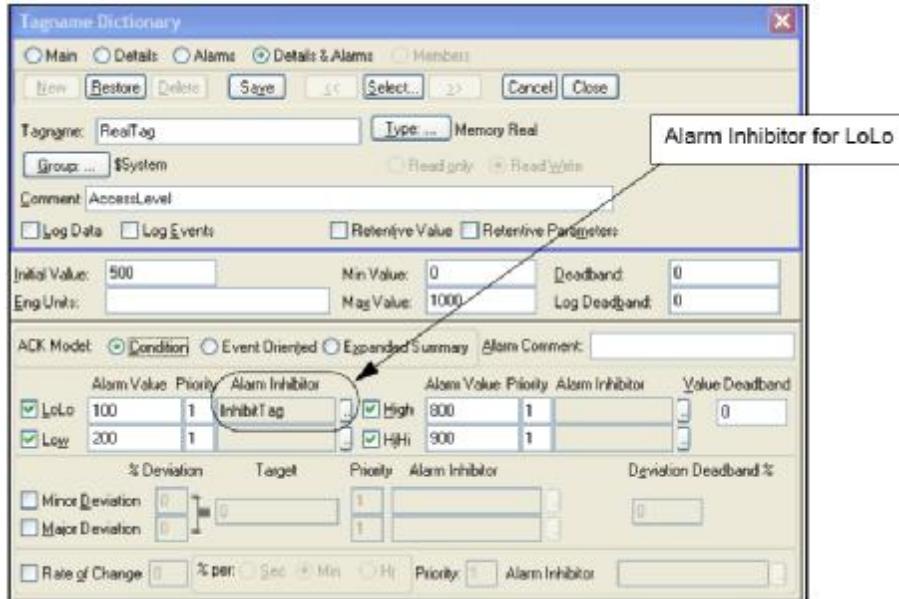
- e. 알람 데이터값 옆에 Alarm Inhibitor의  버튼을 클릭합니다.



알람 Inhibition에 Memory-Discrete 태그 유형은 사용될 수 있지만, Message 태그는 사용할 수 없습니다. 태그 값이 참일 경우, inhibition은 작동됩니다.

f. 태그를 선택한 후, OK 버튼을 클릭합니다.

Tagname dictionary에서 새로운 Inbition이 태그 공란에 생긴 것을 볼 수 있습니다:



g. Save버튼을 클릭하고, Close버튼을 클릭합니다.

알람 제거

- a. Tagname dictionary의 alarms 뷔는 태그의 .alarmEnabled dot field에 접근 하기 위해 사용 할 수 있습니다. 이 Dot field를 이용하여, 알람 그룹이나 각각의 태그에서 발생하는 알람을 제거 할 수 있습니다.

Note : 알람그룹의 알람을 제거하게 되면, 그룹에 속한 모든 태그들의 알람 또한 제거됩니다.

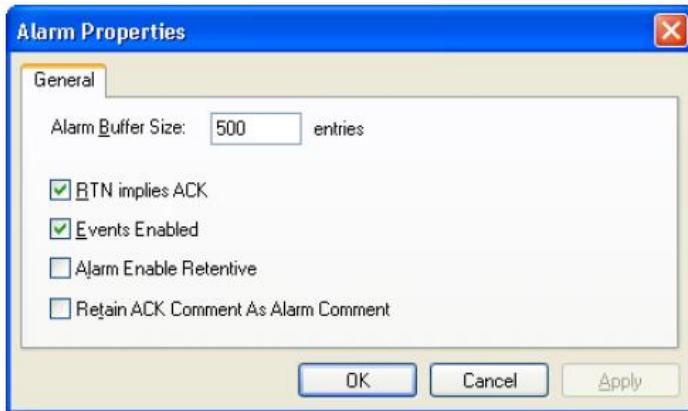
Note : 추가적으로, 다른 알람 Dot field를 사용하여 알람 설정을 변경할 수 있습니다.

알람 시스템 구성

이벤트 가능여부, 알람이 정상상태로 복귀했을 때 자동인지 여부 등 알람시스템의 여러가지 설정들을 구성할 수 있습니다.

알람/이벤트의 일반적 속성 구성

- a. **Special** 메뉴에서 **Configure / Alarms** 을 선택합니다. 또는 Application 탐색기의 **Configure**에서 **Alarms** 을 더블 클릭하면, **Alarm Properties** 창이 나타납니다.:



구성 옵션:

- | **Alarm Buffer Size :** 상자안의 숫자는 windowviewer에서 유지되길 원하는 내부 메모리에 저장되는 알람 이벤트 수입니다.
- | 이 숫자는 노드가 요약 또는 이력 모드로 저장할 수 있는 알람의 최대개수입니다.

Note: 내부 메모리에 저장된 알람 이벤트만 알람 디스플레이 객체로 볼 수 있습니다. 만약 알람을 사용하지 않는다면, 최소값은 (1)로 두는 것이 좋습니다.
이 값을 너무 높게 잡게 되면, 시스템성능이 떨어집니다. 분산 알람 시스템을 사용하기 위해선 기본값으로 500을 추천합니다.

- | **RTN implies ACK :** 알람 발생된 태그가 “정상”상태로 돌아올 때 자동으로 확인 상태가 됩니다. 만약 이 옵션을 선택하지 않을 경우, 알람 발생 후 운전자의 확인이 있어야 “정상”상태가 됩니다.

Note: **Expanded Summary** 확인 모델에서는 **RTN Implies ACK** 옵션이 적용되지 않습니다.

- | **Events Enabled :** 운전자, I/O, QuickScripts, 시스템 등에 의해 변경된 모든 데이터의 변경사항들이 이벤트로 저장이 됩니다. (단, **Log Events** 옵션이 선택된 태그만 저장됩니다.)
- | **Alarm Enable Retentive:** Windowviewer 가 닫히거나 재시작 되어도 **.AlarmEnabled** 변수값을 유지합니다.

- I **Retain ACK Comment as Alarm Comment** : 사용자 정의 주석이 tagname dictionary에서 정의된 알람 주석에 우선합니다.

Section 2 - the Distributed Alarm Objective

분산 알람 객체는 로컬이나 원격에서 발생하는 알람을 보여줍니다. 이 **display** 객체는 스크롤 바, 각각의 **column** 수정, 알람들 다중선택, 갱신 상태 바, 퀼리 상태 바, 오른쪽 마우스 활성화 메뉴, 알람의 우선순위에 의한 색상 변경, 같은 기능을 가지고 있습니다.

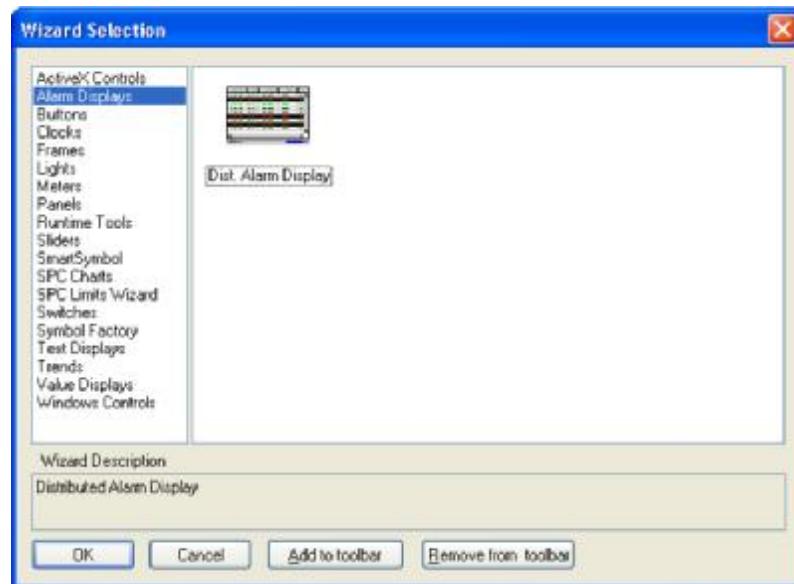
소개

InTOuch에서는 **alarm display**에서 표현되는 형식 및 이 객체에서 보여줄 수 있는 정보를 수정할 수가 있고, 다양한 알람 상황들이 색상별로 표현할 수 있으며, 알람그룹과 알람 우선순위를 표현할 수 있습니다.

Date	Time	State	Class	Type	P...	Name	Group	Provider	V...	Limit
28 Nov	15:43	UNACK	Value	HIHI	1	Alarm1	GroupName	Provider	15	14
28 Nov	15:43	UNACK	Value	HI	250	Alarm2	GroupName	Provider	15	14
28 Nov	15:43	UNACK	Value	LO	500	Alarm3	GroupName	Provider	15	14
28 Nov	15:43	UNACK	Value	LOLO	750	Alarm4	GroupName	Provider	15	14
28 Nov	15:43	ACK	Dev	Minor	1	Alarm5	GroupName	Provider	15	14
28 Nov	15:43	ACK	Dev	Major	250	Alarm6	GroupName	Provider	15	14
28 Nov	15:43	ACK	RDC	1	500	Alarm7	GroupName	Provider	15	14
28 Nov	15:43	ACK	Custom	1	750	Alarm8	GroupName	Provider	15	14

분산 알람 디스플레이 작성하기

- Wizard/ActiveX** 도구상자에 있는 마법사 도구  를 클릭합니다.
- Wizard Selection 창이 나타납니다.



- c. Wizard 목록에서 **Alarm Displays** 를 선택하고, Dist.Alarm Display 를 선택합니다.
- d. OK 버튼을 클릭하면, Wizard Selection 창이 닫되고, 원도우가 다시 나타나면, 붙여넣기를 합니다.
- e. Alarm Display 를 붙여넣을 곳을 지정한 후, 클릭합니다.

Date	Time	State	Class	Type	Priority	Name
28 Nov	16:21	UNACK	Value	HIHI	1	Alarm1
28 Nov	16:21	UNACK	Value	HI	250	Alarm2
28 Nov	16:21	UNACK	Value	LO	500	Alarm3
28 Nov	16:21	UNACK	Value	LOLO	750	Alarm4
28 Nov	16:21	ACK	Dev	Minor	1	Alarm5
28 Nov	16:21	ACK	Dev	Major	250	Alarm6
28 Nov	16:21	ACK	ROC	1	500	Alarm7
28 Nov	16:21	ACK	Custom	1	750	Alarm8

이제 이 객체를 설정 할 수 있는 준비는 끝난 것입니다.

알람 뷰어 기능들

스크롤바

분산 알람 객체는 상하좌우 스크롤바가 내장되어있어, 나열된 알람들을 볼 수가 있으며, 이 스크롤바에 대한 설정을 할 수 있습니다.

컬럼 크기변화

분산 알람 객체는 Grid(격자무늬)를 사용하여 알람 메시지를 보여줍니다. 사용자는 column 을 선택하거나 드래그함으로써, 손쉽게 column 의 폭을 조정할 수 있습니다. 이 기능은 오직 Runtime 시에만 사용할 수 있으며, Grid 의 사용여부 역시 정할 수 있습니다.

Grid column 변화는 저장되지 않으므로, 원도우를 다시 열었을 때에는 기본 폭으로 되돌아 가기 때문에 수직 Grid 라인을 더블 클릭함으로써 컬럼 크기를 자동 설정 할 수 있습니다.

다중 선택

Grid 는 하나 혹은 그 이상의 알람 선택이 가능합니다. 선택된 알람들은 나중에 설명될 **almAckSelect()**라는 QuickScript function 을 사용하여 확인 할 수 있습니다. 분산 알람 객체를 설정 할 경우, toggle (item by item)이나 multiple (CTRL 또는 SHIFT 를 누른 채 다수의

알람을 마우스로 클릭하여 선택하는 방법) 옵션을 선택, 지정 해야 합니다. Runtime 선택을 해제할 수도 있습니다.

알람 메시지 색상

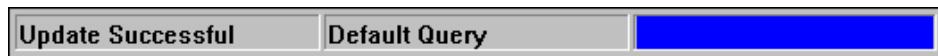
알람의 중요도, 확인 여부에 따라 8 가지의 다른 색상으로 디스플레이 되는 알람 메시지를 설정할 수 있습니다.

업데이트 상태 바

분산 알람 객체는 상태 메시지, 현재 알람 쿼리, 진행의 세가지 상태 바를 가지고 있습니다. 이 상태 바들은 보여지는 쿼리의 현 상태에 대한 전반적인 내용 및 분산 알람 객체에서의 suppression 의 상세한 내용을 제공합니다.

상태 바의 오른쪽 팬(pane)은 freeze 상태일 때 빨간색을, 상태 바의 왼쪽 팬은 suppression 상태일 경우 빨간색으로 바뀐다. suppression 상태일 때는 “suppression”이라는 메시지가 왼쪽 팬에 나타납니다.

Runtime 시에는 상태 바를 끌 수도 있습니다.



기능	설명
Status	상태 바의 가장 왼쪽 끝에 나타나는 상태 메시지는 현재 쿼리 상태의
Message	보다 자세한 설명을 제공합니다.
Progress Bar	상태 바 가장 오른쪽 끝의 업데이트 진행 바는 현재 쿼리 진행에 대한 가시적인 표시를 제공합니다.
Alarm Query	Alarm Query 는 현재 알람 쿼리에 대한 표시를 제공합니다.

Note : Freeze 된 상태에서 새로운 알람이 발생하면, ““New Alarm (s)”라는 메시지가 상태 바에 나타나게 됩니다.

Status Message	상태/표시	상태 바
None	No Query	None
Update Incomplete	Query 미완료	Blue/Green
Update Successful	Query 완료	Solid Blue

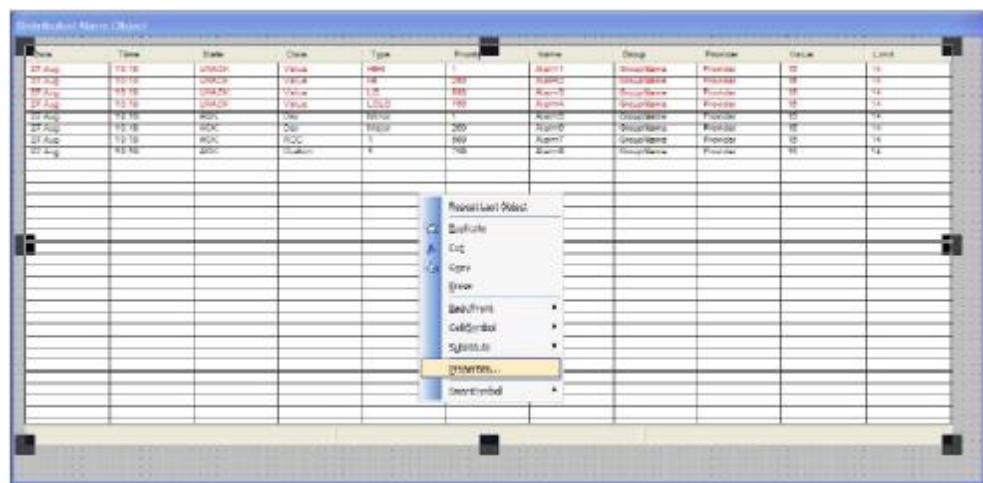
Suppression	Query name	Solid Blue
Freeze	Query name	Red
New Alarm	Query name	Red

마우스 오른쪽 버튼 클릭 메뉴

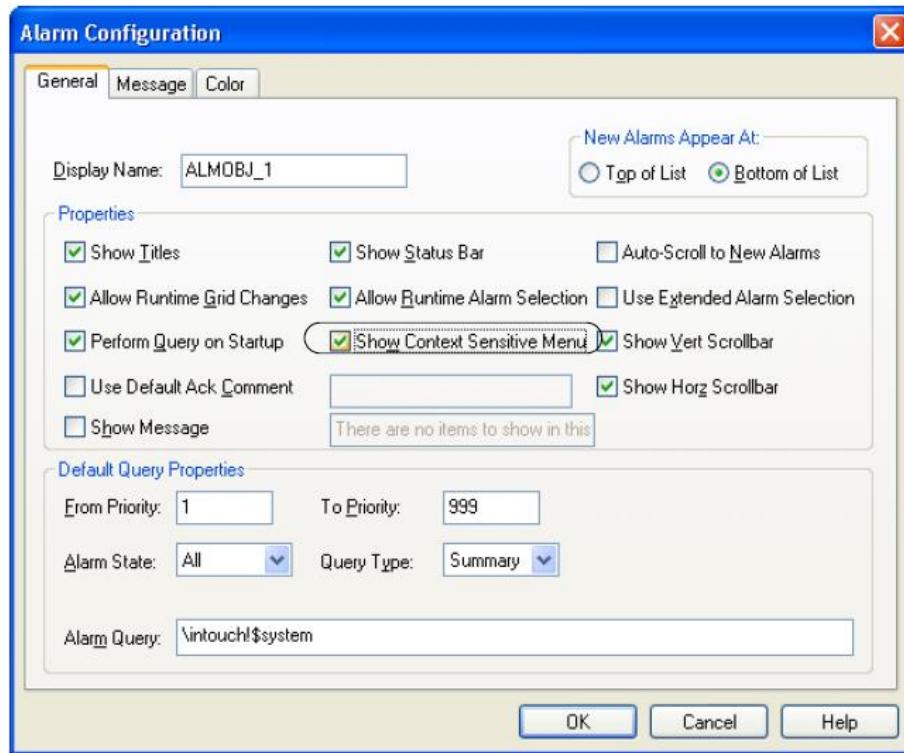
분산 알람 디스플레이 객체는 마우스 오른쪽 버튼 클릭 메뉴에서 보여지는 객체나 알람, 알람 그룹, 태그, 우선순위 등에 빠르게 적용할 수 있는 명령을 Runtime 상에서 제공합니다.

이 메뉴 기능은 기본적으로 비활성화되어 있으며, **Property sheets**에서 적절한 값을 조정하여 활성화 시킬 수 있습니다. 또한, 이 메뉴의 모든 기능들이 스크립트를 사용하여 구현이 가능합니다.

- a. RunTime 시 이 기능을 활성화 시키려면, 개발모드상에서 분산 알람 디스플레이 객체의 오른쪽 버튼을 눌러 **Properties**를 선택하거나 더블 클릭 합니다.



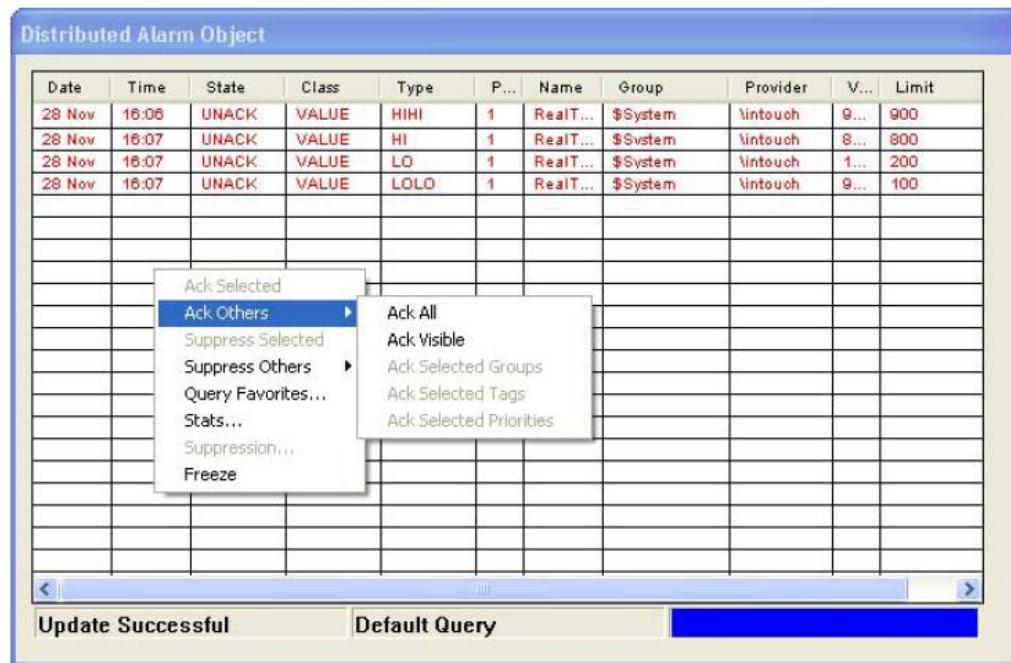
- b. **Alarm Configuration** 대화 상자가 나타납니다. **General tab**에서 **Show Context Sensitive Menu**를 선택합니다.



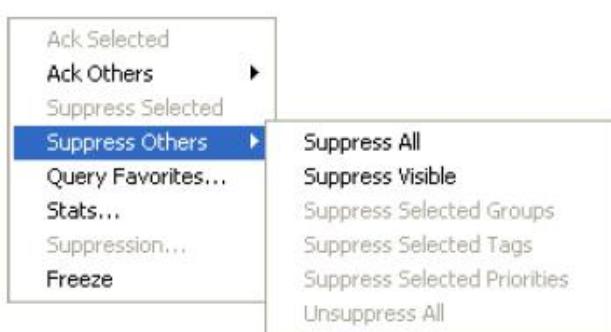
c. **OK** 버튼을 클릭합니다.

이제, Runtime 시 오른쪽 마우스 클릭 기능을 활용할 수 있게 되었습니다. 여러 가지 부가적인 기능들을 알아보도록 합니다.

- I **Ack Selected** – Runtime 시 운전자가 디스플레이 창안에서 알람을 선택한 후, 오른쪽 클릭을 하고 **Ack Selected**를 선택하면, 선택된 알람을 빠르게 확인할 수 있습니다.
- I **Ack Others** – 운전자가 디스플레이 창안에서 오른쪽 클릭을 하고 **Ack Others**를 선택하면, acknowledgment command 들을 포함한 sub-menu 가 열립니다. Runtime 시, 운전자는 이 기능을 사용하여 디스플레이 상의 모든 알람, 보이는 알람 (visible alarms), 알람그룹, 태그 또는 중요도에 따른 알람을 빠르게 확인할 수 있습니다.



- I **Suppress Selected** – Runtime 시 운전자가 디스플레이 창안에서 오른쪽 클릭을 하고 **Suppress Selected**를 선택하면, 선택된 알람을 Suppress 시킬 수 있습니다.
- I **Suppress Others** – 운전자가 디스플레이에 창안에서 오른쪽 클릭을 하고 **Suppress Others**를 선택하면, suppression command 들을 포함한 sub-menu 가 열립니다. Runtime 시 사용자는 이 기능을 사용하여 디스플레이 상의 모든 알람, 보이는 알람 (visible alarms), 그룹, 태그 또는 중요도에 따른 알람을 빠르게 suppress 시킬 수 있습니다.



- I **Query Favorites** – 운전자가 디스플레이 창안에서 오른쪽 클릭을 하고 **Query Favorites**를 선택하면, **Alarm Query** 대화상자가 나타납니다.

- | **Stats** – alarm statistics 대화상자를 엽니다.
- | **Suppression** – Alarm Suppression 대화상자를 엽니다.
- | **Freeze** – 현재 디스플레이를 Freeze 시킵니다.

알람 뷰어 컨트롤 디스플레이 Guidelines

알람뷰어 컨트롤은 윈도우 컨트롤 Wizard 와 비슷한 구조를 가지고 있으며, 이 구조는 분산 알람 같은 객체를 사용할 경우에 아래와 같은 가이드라인을 따라서 사용해야 합니다.

- | 각각의 디스플레이는 구별할 수 있는 식별자를 가지고 있기 때문에, QuickScript function 으로 어떤 디스플레이가 변경되는지를 알 수가 있습니다. 이 식별자는 **Alarm Configuration** 창안에서 **Display Name** 에 입력해야 하며, 각각의 디스플레이에 고유한 이름으로 입력해야 합니다.
- | 디스플레이들은 윈도우 컨트롤 혹은 그래픽 같은 InTouch 객체들과 겹쳐서는 안됩니다. 이는 windowmaker 에서 분산알람 객체를 클릭한 뒤, “handles”를 선택함으로써 확인할 수 있습니다. 이 기능은 화면 위의 다른 그래픽 객체들과 겹치지 않도록 작동합니다.
- | 다수의 디스플레이들이 하나의 화면에 있다면, 시스템 성능을 감소 시키는 결과를 가져오게 되므로 디스플레이들은 적당히 사용해야 합니다.
가능한, 화면 위에 보여줄 디스플레이의 수를 제한을 해야 하며, 추가적인 디스플레이들을 포함하고 있는 화면을 불러들이는 방법을 사용해야 합니다.

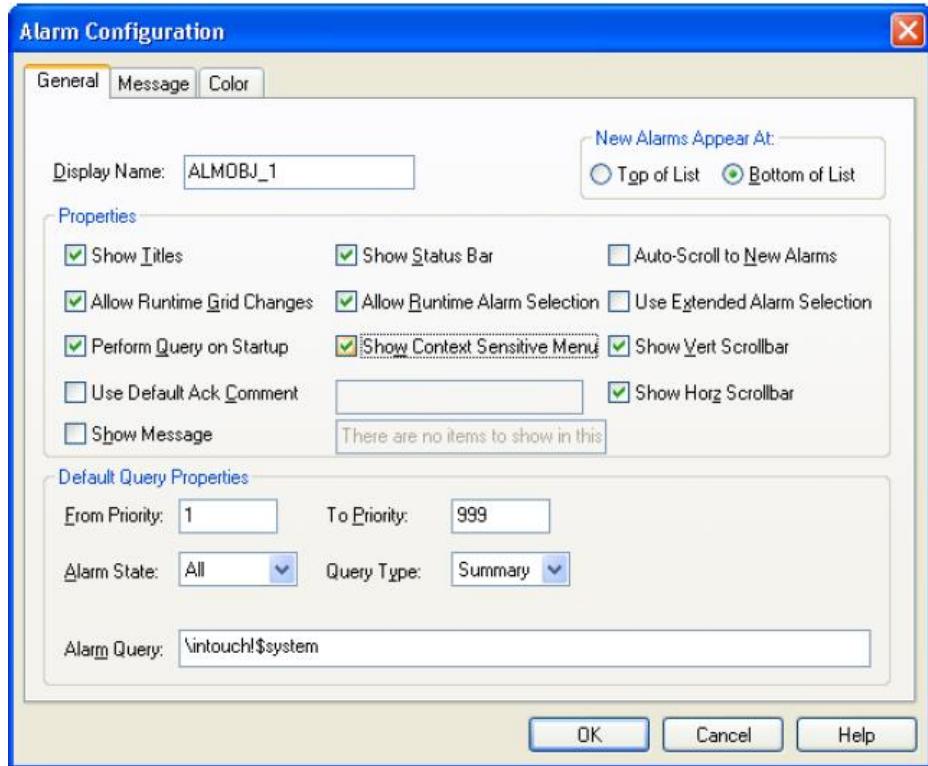
Note : 추가적으로 알람 디스플레이들은 ActiveX Alarm Object”처럼 이용할 수 있습니다.

알람 디스플레이 구성 옵션

Alarm Configuration 창 안에는 **General, Message, Color** 이 세 가지의 구성 옵션으로 구성되어있습니다.

Note : 이 구성 대화상자는 일반 Windows property sheet 와 마찬가지로 **OK** 버튼을 누를 때까지 어떠한 설정도 저장되지 않습니다. 또한 적절하지 않은 값이 입력되었을 때는 해당 에러 메시지가 나타납니다. **Cancel** 버튼을 누르면 변경한 모든 설정 값을 취소되며, 창은 닫히게 됩니다.

- a. 분산 알람 객체를 더블 클릭합니다.



설정 옵션

Display Name : 알람 디스플레이의 이름을 입력합니다. 이 이름에는 각각의 고유한 알람 디스플레이 이름이 들어가야 합니다.

Note: 여기서 입력된 이름은 시스템 전체에서 발생하는 알람에 대한 확인이나 쿼리의 실행 같은 작업을 수행하는데 있어서 사용하게 됩니다.

New Alarms Appear At : 새로운 알람이 발생하게 되면, 원하는 곳에 그 알람을 보여주게 됩니다.

- | **Top of List :** 최근 발생한 알람을 리스트 최상단에 표시.
- | **Bottom of List :** 최근 발생한 알람을 리스트 최하단에 표시

Property Setting

Property	설명
Show Titles	알람 메시지 타이틀 바에 디스플레이되는 내용.
Allow Runtime Grid Changes	Runtime 상에서 컬럼 길이를 바꿀수 있게 함.
Perform Query on Startup	체크시, Default query properties 를 이용하여 디스플레이를 자동으로 업데이트 함. 미체크시는 디스플레이를 업데이트하기 전에 almDefQuery 나 almQuery 를 실행하여야 함.
Use Default Ack Comment	특정 comment 없는 알람을 확인할 때, 입력한 스트링이 기본 comment로 나타나게 함.
Show Status Bar	상태 바 디스플레이.
Allow Runtime Alarm Selection	Runtime 상에서 알람을 선택할 수 있음.
Show Context Sensitive Menu	오른쪽 마우스버튼 팝업 메뉴를 활성화 함.
Auto-Scroll to New Alarms	사용자가 리스트의 처음에서 스크롤하면 자동으로 현재 디스플레이 되지 않고있는 새로운 알람으로 스크롤 함.
Use Extended Alarm Selection	마우스와 Ctrl 또는 Shift 버튼을 동시에 눌러 다수의 알람을 선택하게 함. 기본적으로는 각각의 알람을 클릭하면 해당 알람만 선택 함. (Allow Runtime Alarm Selection 을 선택했을 시만 보임).

Default Query Properties

TIP: Default Query Properties 는 Perform Query on Startup 옵션이 선택되거나 almDefQuery QuickScript function 이 실행될 때 사용 할 수 있습니다.

Property	설명
From Priority	가장 낮은 알람 우선순위.
To Priority	가장 높은 알람 우선순위. 알람 우선순위에 관한 자세한 정보는 본 섹션 "Alarm Priorities"를 참조.
Alarm State	Query 할 알람 상태 (All, UnAck, Ack).
Query Type	Summary(요약) 나 Historical(이력) 디스플레이 설정.
Alarm Query	초기 알람 쿼리 설정. 텍스트만 지원; Tag는 지원하지 않음. 적절한 문법은 다음과 같습니다: ¥¥Node¥InTouch!Group : 알람 그룹의 경로

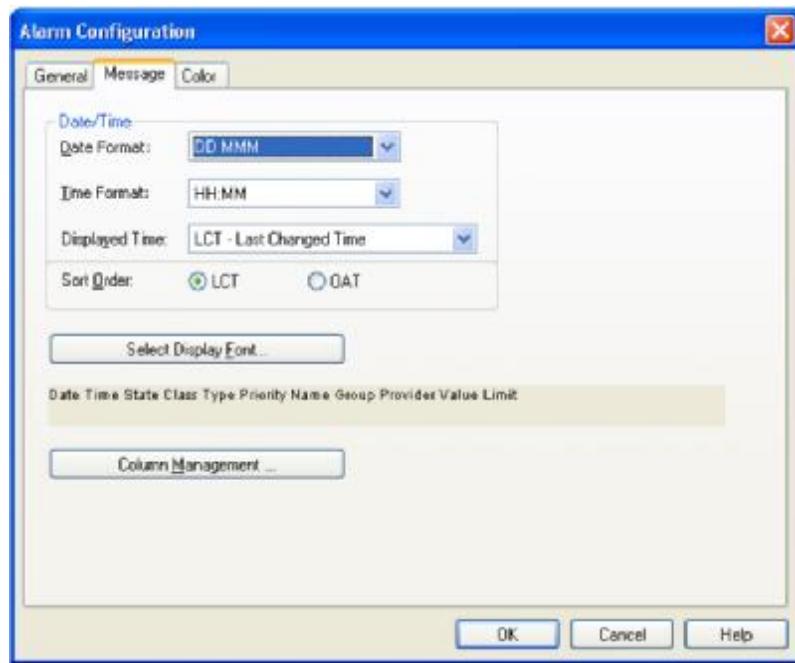
¥InTouch!Group : 로컬 알람 그룹의 경로

GroupList : 다른 그룹 리스트

Note: 다수의 쿼리를 실행하기 위해서는, 빈 칸으로 각 쿼리들을 구분해야 합니다. **Alarm Providers** 를 사용하는 부분은 **InTouch Advanced** 과정에서 자세히 다루게 됩니다.

분산 알람 메시지 형식

- a. Message 탭을 클릭합니다.



Note: 알람 구성 창안에서 텍스트박스를 오른쪽-클릭을 할 경우, 선택한 문자에 적용되는 명령을 보여주는 메뉴가 열리게 됩니다.

Preview area (dialog 의 하단부) 는 현재 설정된 알람 메시지의 형식을 보여주데, 이는 선택된 폰트에 적용은 되지만 색상은 적용되지 않습니다.

설정 옵션

Date Format : 사용할 수 있는 날짜 형식은 다음과 같습니다:

Selection	Display	Selection	Display
DD MMM	28 Feb	MM/DD	02/28
DD MM YYYY	28 Feb 2002	MM/DD/YY	02/28/02
DD/MM	28/02	MMM DD	Feb 28
DD/MM/YY	28/02/02	MMM DD YYYY	Feb 28 2002
YY/MM/DD	02/02/28	YYYY/MM/DD	2002/02/28

Time Format : 이 필드의 값들은 시간형식을 명세화하는 템플릿으로 사용됩니다. 예를 들어, 10:24:30 AM 의 시간형식은 HH:MM:SS AP 로 명세화 할 수가 있습니다.

Template 의 문자들 명세는 다음과 같습니다:

AP	AM/PM 형식을 선택. 예를 들어, 오후 3 시는 3:00 PM 으로 디스플레이. 별다른 설정이 없을 경우 24 시간 주기로 설정됨. 예를 들어, 오후 3 시는 15:00 로 디스플레이.
HH	Alarm/event 가 발생된 시를 디스플레이
MM	Alarm/event 가 발생된 분을 디스플레이
SS	Alarm/event 가 발생된 초를 디스플레이
sss	Alarm/event 가 발생된 1/1000 초를 디스플레이

Displayed Time : 알람 상태에 따른 디스플레이 타입을 설정.

사용 가능한 형식은 다음과 같습니다:

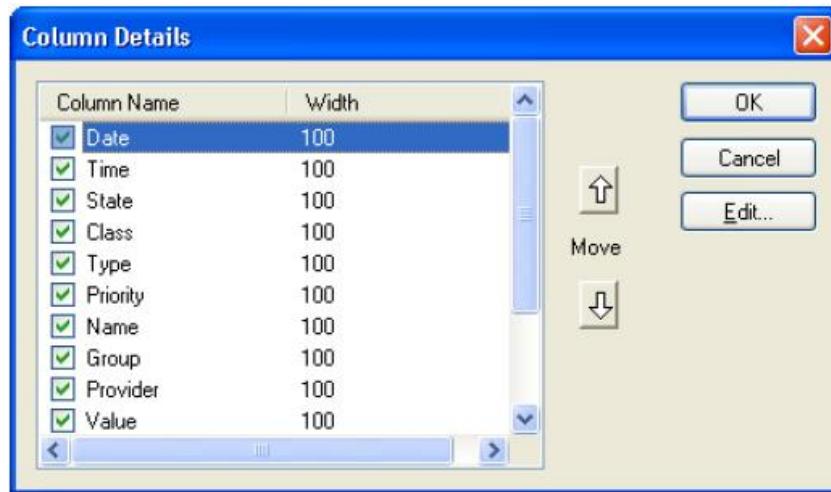
LCT	Last Changed Time – 알람의 가장 최근의 상태 변화에 대한 date/time 스탬프: 알람 설정, sub-state 의 변화, return to normal, 또는 알람 확인.
LCT But OAT on ACK	Last Changed Time, but Original Alarm Time on ACK – 알람 미확인시는 LCT 가 사용되고, 확인시는 OAT 가 사용됨.
OAT	Original Alarm Time – 알람 설정 시의 date/time 스탬프.

Sort order : 알람 객체 안에서 정렬되는 알람의 순서를 원하는 방식(LCT or OAT)으로 선택할 수 있습니다.

Select Display Font : 알람 디스플레이안에서의 폰트, 크기, 스타일을 변경할 수 있습니다.

Column Management : Column Detailsダイ얼로그 박스에서 컬럼의 순서나 해당 컬럼의 폭을 정할 수 있습니다.

상, 하 화살표를 사용하여 디스플레이 되는 순서를 정렬하고, 원하는 **Column Names** 을 선택합니다. **Edit** 버튼을 사용하여 선택된 컬럼의 이름과 사이즈를 변경할 수 있습니다.



Date : 날짜를 디스플레이 합니다.

Time : 시간을 디스플레이 합니다.

State(UnAck, Ack) : 알람 상태를 디스플레이 합니다.

Class(VALUE.DEV.ROC...) : 알람 카테고리를 디스플레이 합니다.

Type(HIHI,LO,MAJDEV,...) : 알람의 유형을 디스플레이 합니다.

Priority : 알람의 우선순위를 디스플레이 합니다.

Name : 알람/태그명을 디스플레이 합니다. **Length** 박스에서는, 디스플레이를 원하는 글자수를 입력한다. (최고 64 자)

Name : 알람 그룹의 이름을 디스플레이 합니다. **Length** 박스에서는, 디스플레이를 원하는 글자수를 입력한다. (최고 64 자)

Provider : 알람 제공자를 디스플레이 합니다.

Value : 알람 발생시, 태그의 데이터 값을 디스플레이 합니다.

Note : Value 는 정확도를 높이기 위해서 적당한 사이즈가 필요합니다.(최고 32 자)

Limit : 태그의 알람 제한값을 디스플레이 합니다.

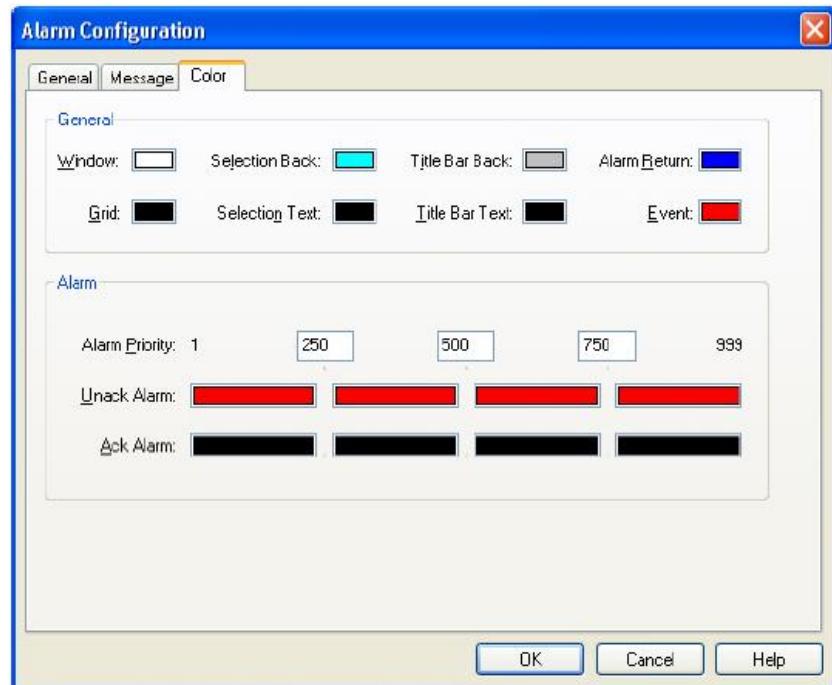
Note : 이 필드의 사이즈는 정확도를 높이기 위해서 적당한 사이즈가 필요합니다.(최고 32 자)

Operator : 알람 상태와 연관된, 로그인 한 운전자의 ID 를 보여줍니다.

Comment : 태그의 주석을 보여줍니다. 태그의 알람이 데이터베이스에 저장 될 때, Alarm Comment 상자에 입력했던 주석부분입니다.

분산된 알람 객체의 Color 속성

- 분산 알람 객체를 더블 클릭 하거나 마우스 오른쪽 버튼을 눌러 **Alarm Configuration** 창을 연 후, **Color** 탭을 선택합니다.



General 상자 : 각각의 칼라 상자는 InTouch Palette 를 열고, 다음 옵션에 따라 원하는 색상을 선택합니다.

Option	설명
Window	디스플레이의 바탕 색상 지정.
Grid	디스플레이의 Grid 색상 지정.
Selection Back	선택된 글자의 바탕 색상 지정.
Selection Text	선택된 글자의 색상 지정.
Title Bar Back	Title bar 의 바탕 색상 지정 (Show Titles option 이 선택되었을 때만 보임).

Title Bar Text	Title bar 의 글자 색상 지정 (Show Titles option 이 선택되었을 때만 보임).
Alarm Return	복구된 알람의 색상 지정 (확인되지 않고 원상태로 복귀된 알람).
Event	Sets color of Event alarms.

Alarm Priority : 알람 디스플레이의 breakpoint 값을 입력합니다.

UnAck Alarm 과 **Ack Alarm** : InTouch palette에서 각각의 breakpoint 값에 원하는 색상을 지정합니다.

알람 확인

Action이나 Key QuickScript의 Ack 함수를 사용하거나, 분산알람객제의 context sensitive 오른쪽-클릭 메뉴를 사용하여 알람들을 확인 할 수 있습니다. Ack 함수는 로컬에서 발생하는 모든 알람의 유형들의 알람 확인 상태를 감시하거나 제어할 수 있습니다.

Action 또는 Key QuickScript 와 link 할 3-D 버튼이나 다른 object를 만든 후, QuickScript Action Script에 다음과 같은 수식을 입력합니다.

Statement	설명
Ack \$System;	시스템의 모든 local 알람을 확인합니다.
Ack Group Name;	특정 알람 그룹의 모든 local 알람을 확인합니다.
Ack Tagname;	특정 태그명의 알람을 확인합니다.
\$System.Ack=1;	시스템의 모든 local 알람을 확인합니다.
Group Name.Ack=1;	특정 알람 그룹의 모든 local 알람을 확인합니다.
Tagname.Ack=1;	특정 태그명의 알람을 확인합니다.
Tagname.AckDev=1;	특정 태그명의 편차 알람을 확인합니다.
Tagname.AckROC=1;	특정 태그명의 rate-of-change 알람을 확인합니다.
Tagname.AckValue=1;	특정 태그명의 Value 알람을 확인합니다.

알람 확인 QuickScript Functions

분산 알람 시스템은 쿼리할 수 있는 어떠한 알람에 대해서도 확인이 가능합니다 (summary display only). 이 기능을 제공하기 위해, 분산 알람 시스템은 알람 확인 QuickScript function 들을 가지고 있습니다. 이러한 function 들은 InTouch 알람 시스템에서 local 알람 확인과 알람 그룹을 위해 사용되는 .Ack dot field 를 보완합니다. 이 ALM QuickScript function의 특정 syntax는 “Script Function”에 자세히 설명되어 있습니다.

Alarm Ack Function 와 Comment 넣기

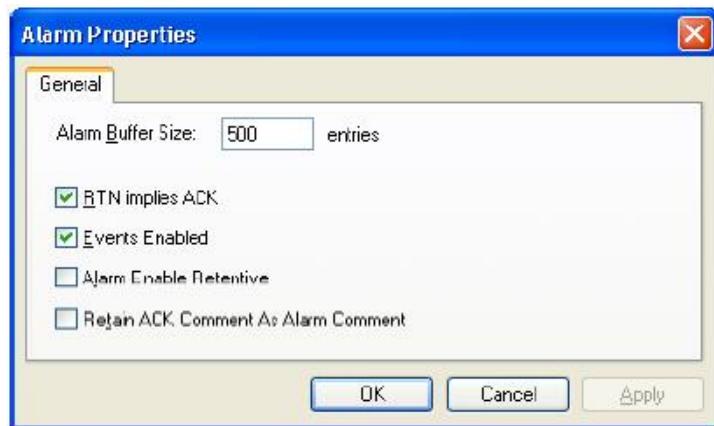
각각의 알람 확인 function에는 comment를 적용할 수가 있습니다. 알람을 확인하는 운전자는 이 comment를 사용하여 해당 알람에 대한 정보를 추가할 수 있으며, InTouch는 이 알람을 알람 데이터베이스에 저장하거나 프린트할 수 있습니다.

분산 알람 시스템은 **Alarm Comment**라는 데이터베이스 내의 태그의 알람 설정 상자를 사용하여 이러한 comment들을 보내줍니다.

사용자는 ACK comment들을 위해 tagname **Alarm Comment** 박스의 사용여부를 선택할 수 있습니다. Tagname **Alarm Comment** 상자 사용시, 이 field의 모든 comment들은 Runtime 상에서 덮어써지게 됩니다. (Tagname dictionary에는 영향을 끼치지 않습니다.)

- Special에 Configure에 Alarms를 클릭합니다.

Alarm Properties 창이 나타나게 됩니다:



- 다양한 almAck() 함수들을 이용한 사용자 정의 주석을 사용하기 원할 경우, Alarm Comment로써 Retain ACK Comment를 선택합니다. 미체크시, Tagname dictionary의 alarm comment를 사용하게 됩니다.
- OK 버튼 클릭.

Tip: 알람 구성 창의 텍스트 상자에서 오른쪽-클릭을 하게 되면 해당 텍스트상자에 적용 가능한 형식을 보실 수 있습니다.

Lab 7 – Creating Alarms and Events

이번 랙에서는 알람 확인 및 이벤트 발생을 위한 알람 윈도우 생성 및 설정 방법을 다루게 됩니다.

단계별 개요

좀 더 수월한 랩 실습을 위해서, 이번 랙을 다섯 가지 세부 항목으로 이해하기 쉽도록 나누어 놓았습니다.

각각의 단계들은 다시 몇 가지 세부적인 항목으로 다시 나누어 볼 수가 있습니다.

Lab 6a – 알람 확인

- a. 알람 윈도우를 Import 합니다.
- b. Dist.Alarm Display 를 조절하고, 배열시킵니다.
- c. 윈도우의 태그명을 변환 시키고, 태그의 알람들을 설정합니다.
- d. 확인 버튼을 설정함으로써 Touch Pushbuttons – 액션 스크립트에서 사용합니다.

Lab 6b – 알람 Inhibition

Note: 다음 각각의 항목들을 완성하기 위해서, 그 결과들을 Runtime 상에서 테스트 해 볼 것을 추천합니다.

- e. 윈도우에 Knob Switch 를 삽입합니다.
- f. Discrete Switch Wizard 를 설정합니다.

Lab 6c – 알람 dot field

- g. \$System.Alarm 결과를 활용하기 위한 Symbol 을 선택합니다.
- h. Value Display – Discrete 애니메이션 링크에 태그를 할당합니다.
- i. Output – Discrete Expression 를 설정하고, On, Off 메시지를 입력합니다.
- j. \$System.Ack 결과를 위해 사용되는 Symbol 을 선택합니다.
- k. Touch Link / User Input – Discrete 애니메이션 링크에 태그를 할당합니다.
- l. Input – Discrete Tagname 의 Expression 에 설정을 하고, Msg to User, Set / Rest Prompts 및 On / Off 메시지를 입력합니다.

- m. 이전 단계처럼, Value Display – Discrete 애니메이션 링크에서 RealTag.HiLimit 를 할당하고 설정합니다.
- n. Min / Max Values 및 Msg to User 를 설정합니다.
- o. 이전 단계처럼, Value Display – Discrete 애니메이션 링크에서 RealTag.HiStatus 를 할당하고 설정합니다.
- p. On, Off 메시지를 정의합니다.

Lab 6d – 알람 예고

한계치에 접근하면, 경고 및 주의의 기능을 하는 윈도우를 생성하고 설정합니다.

- q. 알람 확인 윈도우를 생성하고 설정합니다.
- r. 깜빡거림을 실행시키는 Touch PushButton – 액션 컨디션 스크립트를 생성하고, Object Blinking – Discrete Value 애니메이션을 설정한 후 Symbol 을 선택합니다.

Lab 6e – 이벤트 모니터링 생성

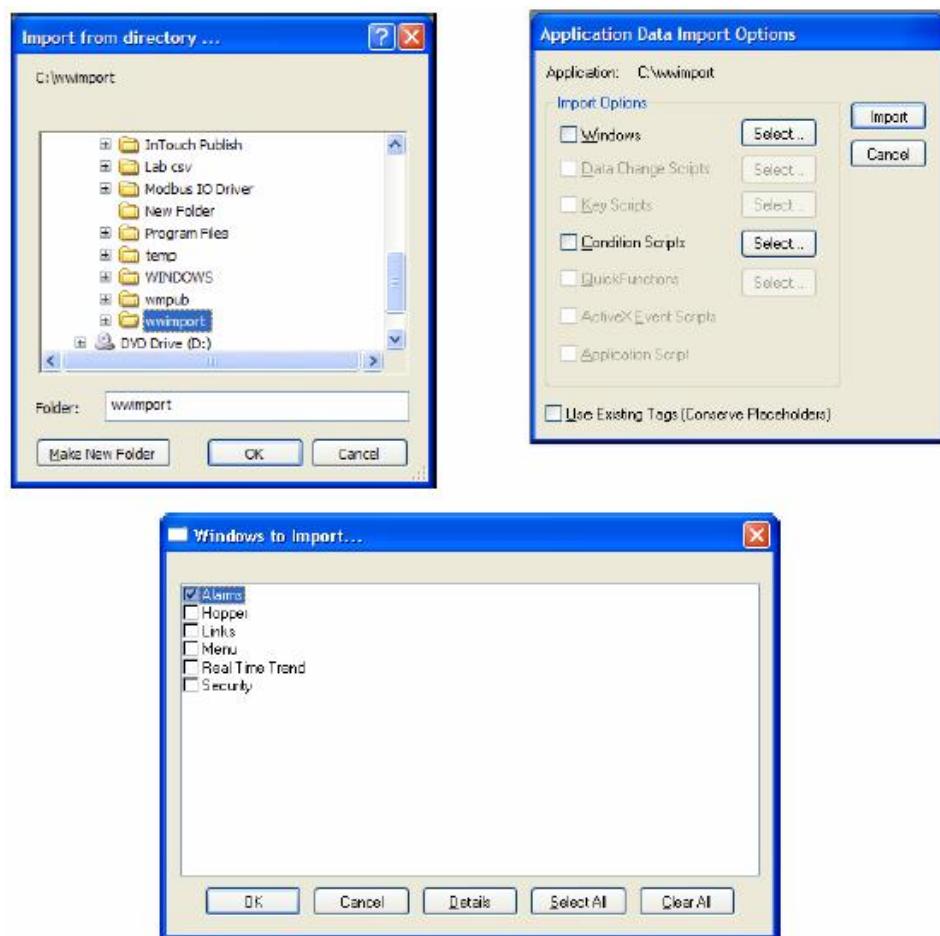
- s. 알람 설정 창에서, Historical 로써 알람 객체의 Query Type 을 설정합니다.
- t. Log Events 에 DiscTag1 을 할당하고, 우선순위를 998 로 입력합니다.

Lab Instructions

Lab 6a – 알람 확인

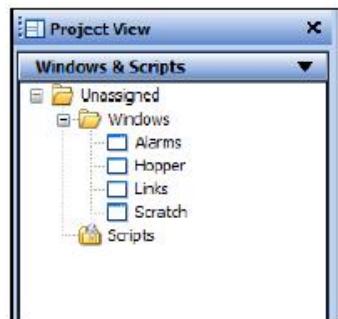
알람 윈도우 Import

1. Windowviewer 를 닫습니다.
2. Windowmaker 에서 열려있는 윈도우를 모두 닫습니다.
3. Windowmaker 메뉴상에서 File / Import 를 선택한 후, (이전 랙에서 사용된)wwimport 폴더를 선택합니다.



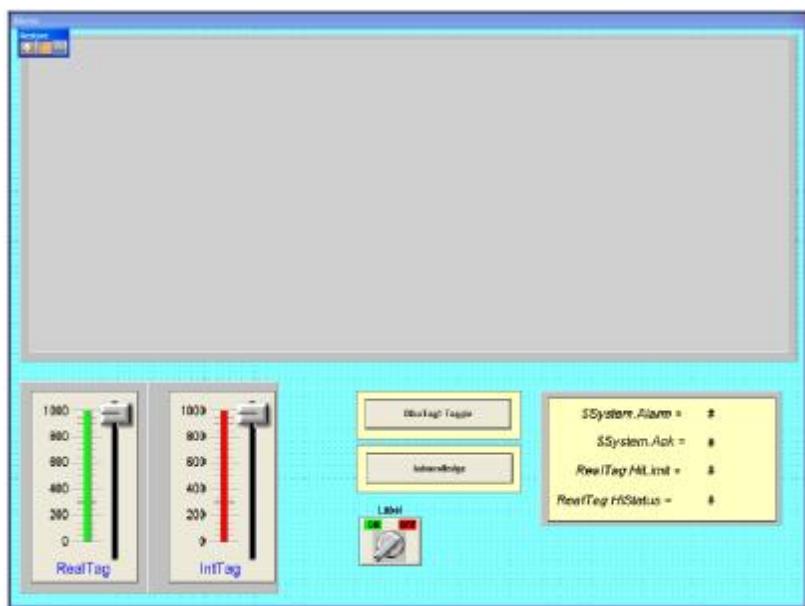
4. Alarms 윈도우를 선택합니다.
5. OK 버튼을 클릭합니다.
6. Import 를 클릭합니다.

Application Explorer Project View 에서 다음처럼 알람 윈도우가 보일 것입니다.



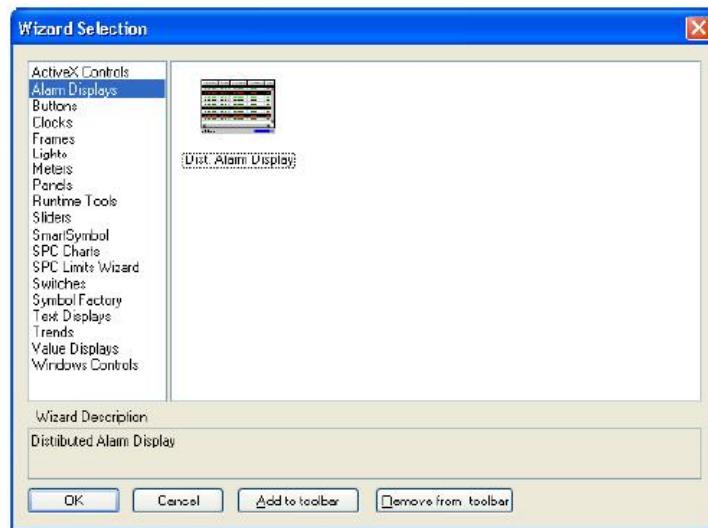
알람 윈도우 설정

7. Alarms 윈도우를 엽니다.



8. Wizard / ActiveX Toolbar에 위치해 있는 Wizard 를 클릭합니다.
9. Wizard 선택을 하게 되면, 다음 다이얼로그 창이 나타납니다:

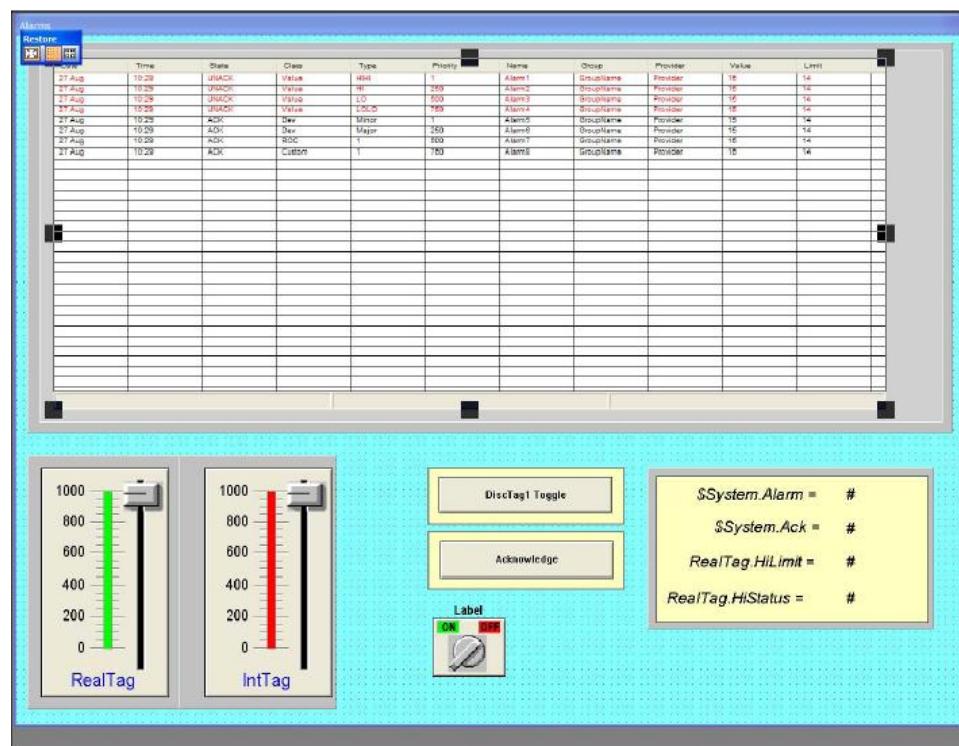




10. Alarm Displays 를 선택합니다
11. Dist.Alarm Display 컨트롤 박스를 선택합니다.
12. OK 버튼을 클릭합니다.

Wizard Selection 창을 닫고, “붙여넣기”모드 에서 커서를 원하는 윈도우의 위치에서 클릭을 합니다.

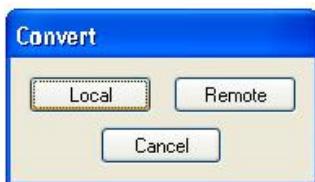
13. 윈도우안에서 Alarm Display 의 사이즈를 적당히 맞춥니다.



14. F2 키를 누릅니다.
15. Special / Substitute Tags 를 선택합니다.
16. Conver 항목을 클릭합니다.



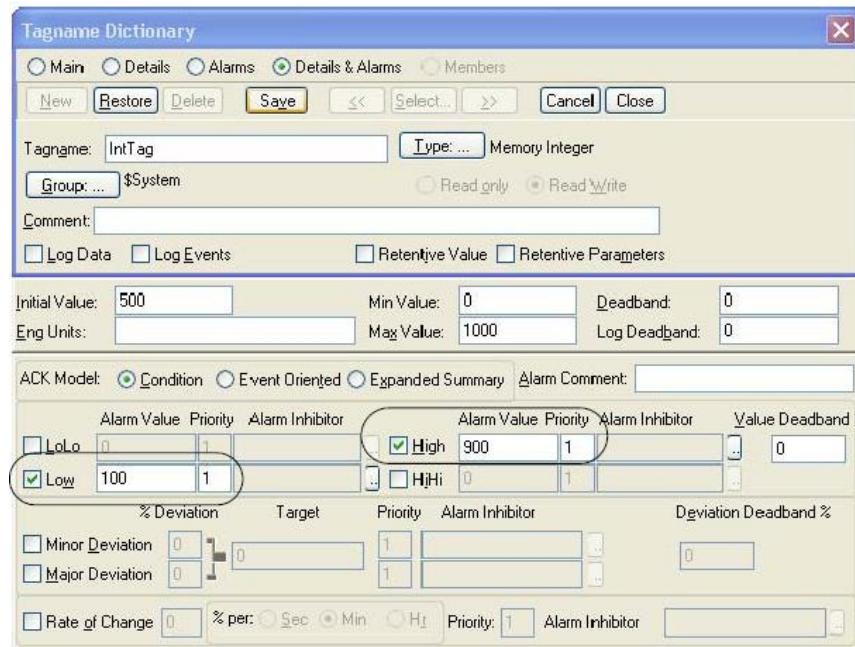
17. Convert 창이 나타나면:



18. OK 버튼을 클릭합니다.

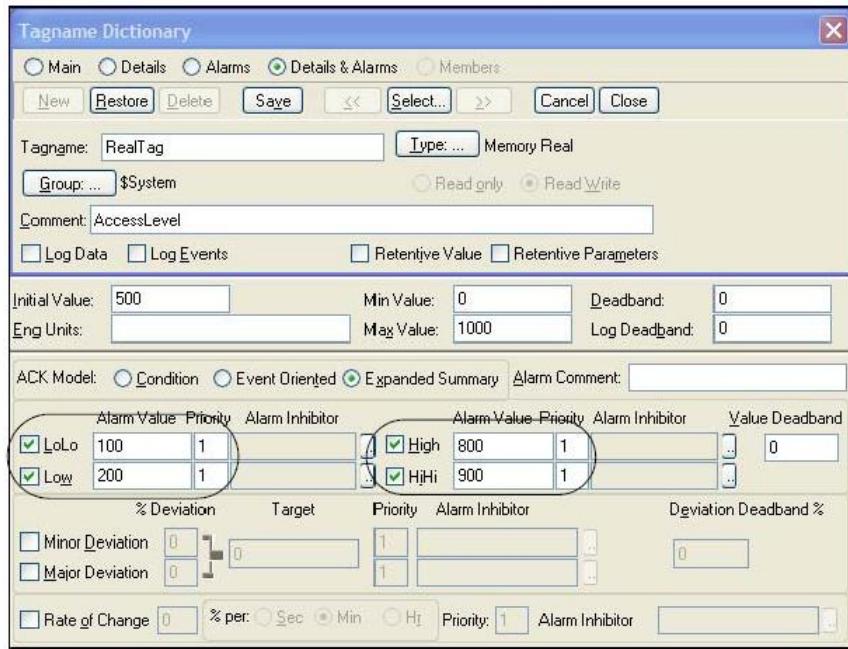
알람 태그들의 설정

19. Tagname dictionary 에서 IntTag 를 선택합니다.
20. Details &Alarm 옵션을 선택합니다.
21. 다음 그림처럼 알람을 설정한 뒤, Save 버튼을 클릭합니다.



22. Tagname dictionary 에서 RealTag 를 선택합니다.
23. 다음 그림처럼 알람을 설정합니다.

24. Save 버튼을 클릭한 후, Close 버튼을 클릭합니다.



25. Runtime 을 실행시킵니다.

26. RealTag 및 IntTag 의 슬라이더를 이동시켜, Alarm Display 를 테스트 합니다. Alarm

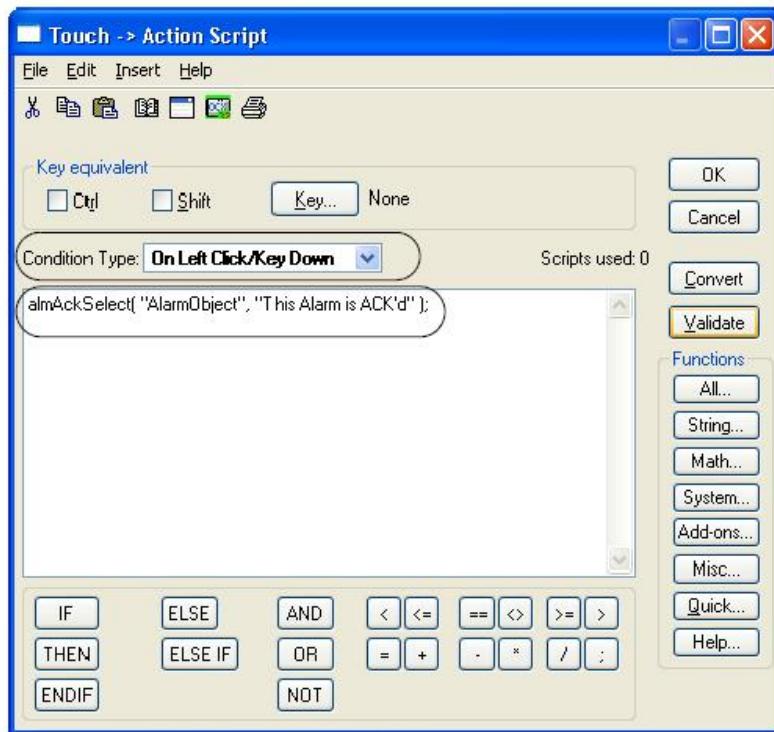
Display 는 슬라이더가 이동 할 때마다, 갱신되는 것을 볼 수 있습니다.

27. 개발모드로 전환합니다.

Note: Tagname dictionary 에서 ACK Model 을 변경해도, Windowviewer 를 종료시킨 후 재시작하기 전까지 변경 사항이 적용되지 않습니다.

확인 버튼

28. Acknowledge 버튼을 더블-클릭 한 후, Touch Pushbuttons – Action 을 선택합니다.



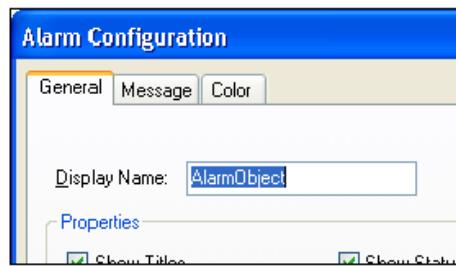
29. 이전에 입력된 스크립트를 지웁니다.

30. 위의 그림에서 보이는 스크립트를 Action Script 에서 동일하게 입력 및 설정을 합니다.

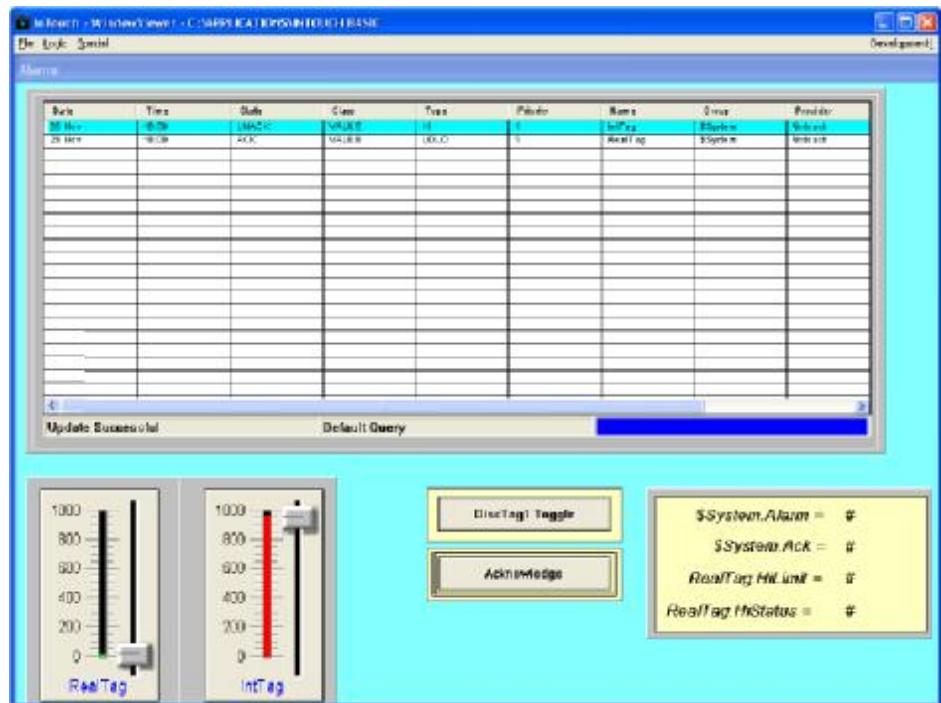
31. 스크립트가 올바르게 입력되었는지, Validate 버튼을 클릭해서 확인합니다.

이 스크립트는 Distributed Alarm 객체에 almAckSelect 함수를 이용함으로써, Acknowledge 버튼이 알람 객체를 확인 이 가능하게 합니다.

Note : almAck 함수의 모든 알람객체의 인수들의 값은 알람 객체 설정의 name 속성에서 할당 받습니다. 알람 객체의 Display name 을 일치시키려면, 스크립트를 수정해야 합니다.



32. OK 버튼을 클릭, 다시 한번 OK 버튼을 클릭합니다.
33. Runtime 으로 전환한 후, RealTag 와 IntTag 의 알람을 초기화합니다.
34. 알람들 중 하나를 선택합니다.
35. Acknowledge 버튼을 클릭합니다. 알람의 색상은 확인 작업이 정상적으로 이루어지면 바뀌게 됩니다.

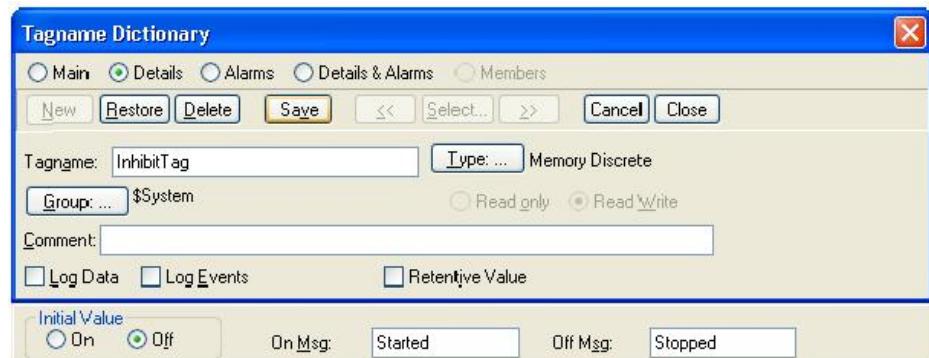


Lab 6b – 알람 Inhibition

Inhibit 태그 생성

Inhibition 태그는 알람의 세부적인 속성을 비활성화 시키는 기능을 합니다.

36. Tagname dictionary 를 열고, New 버튼을 클릭합니다.



37. InhibitTag 를 Memory Discrete 타입으로 생성합니다.

38. Save 버튼을 클릭합니다.

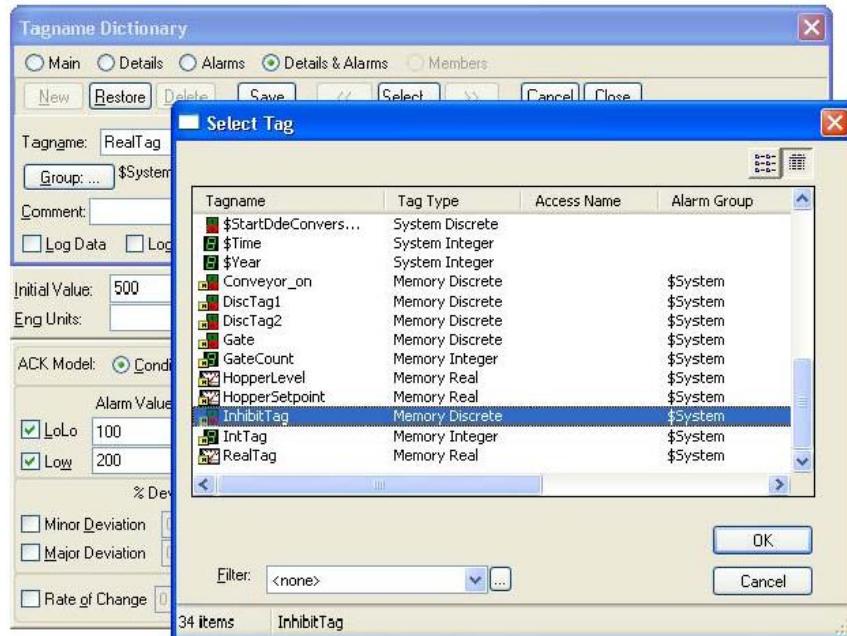
39. Select 버튼을 클릭합니다.

40. Tagname dictionary 에서 RealTag 를 선택합니다.

41. OK 버튼을 클릭합니다.

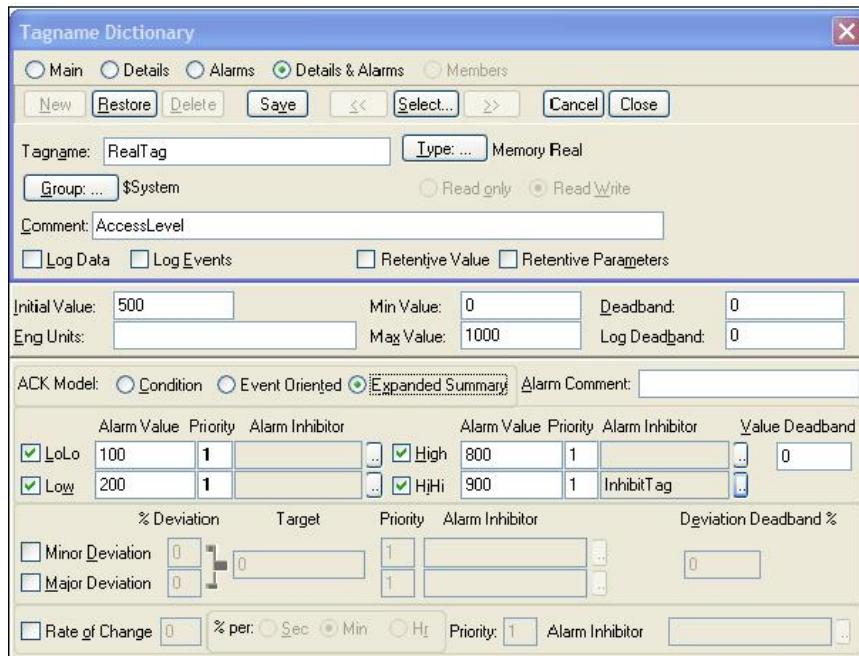
42. Details & Alarams 옵션을 선택합니다.

43. HiHi 알람 inhibitor 를 설정하기 위해서, [...] 버튼을 클릭합니다.



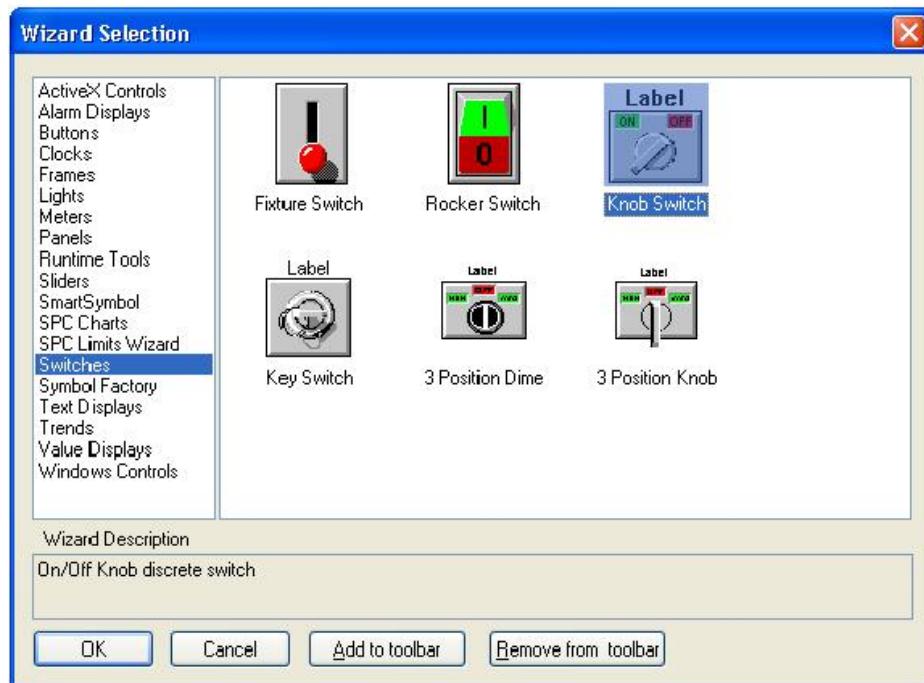
44. Tagname dictionary 으로 InhibitTag 를 선택한 후, OK 버튼을 클릭합니다.

45. Save 버튼을 클릭한 후, Close 버튼을 클릭합니다.



Inhibitir 스위치 생성

46. Wizard Hat 을 툴 박스의 아이콘 을 선택한다.



47. Switches 를 선택합니다.

48. Knob Switch 를 선택합니다.

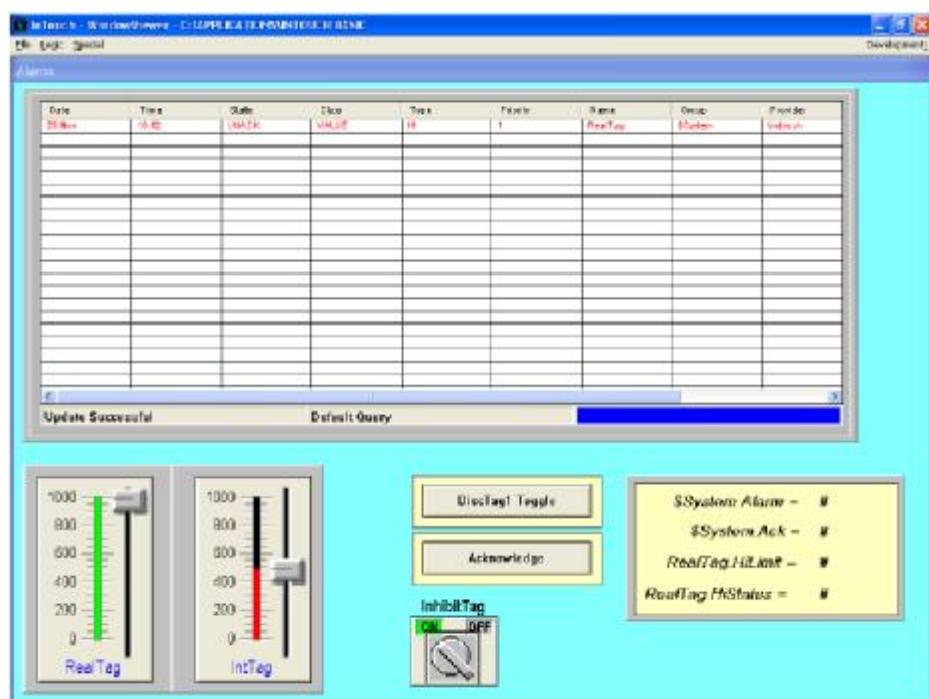
49. Wizard Selection 창에서 Knob Switch 를 더블-클릭 한 후, 어플리케이션의 원하는 위치에 붙여 넣습니다.

50. 붙여 넣은 Knob Switch 를 더블-클릭합니다.

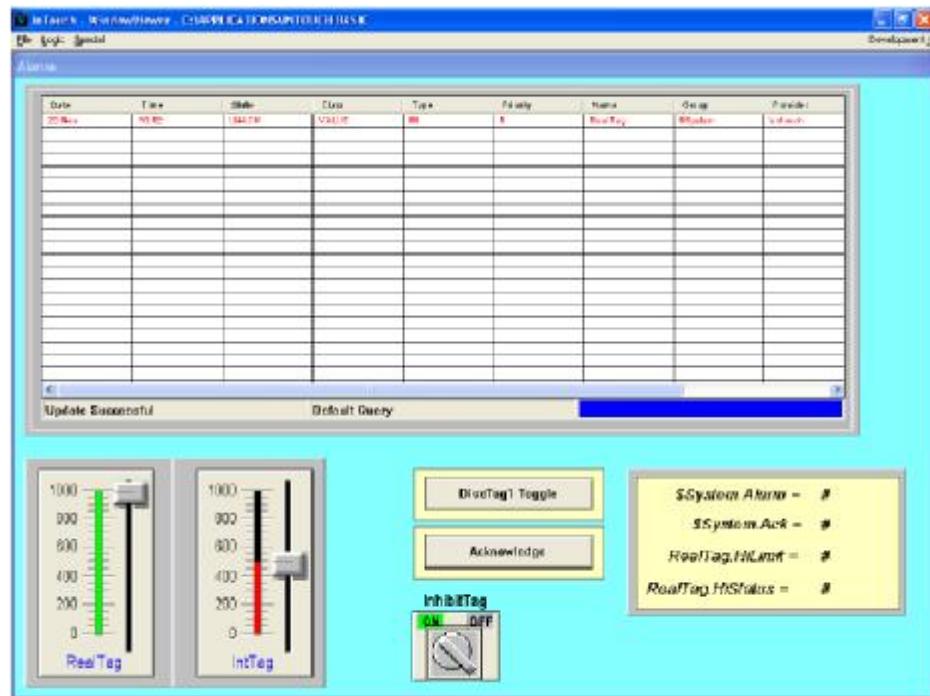
Discrete Switch Wizard 창을 보면 다음과 같습니다:



51. Tagname 의 공란을 더블-클릭을 한 후, Tagname dictionary 에서 InhibitTag 를 선택합니다.
52. OK 버튼을 클릭 한 후, 다시 한 번 클릭합니다.
53. Runtime 으로 전환합니다.



54. knob Switch 를 클릭함으로써, On 상태로 셋팅합니다.
55. RealTag 슬라이더를 최상위로 이동시킵니다. Aalarm display 에는 Hi Alarm 상태만 나타납니다. 현재 HiHi 알람은 inhibit 되어 있는 상태이므로, 오직 Hi 알람만 보이는 것입니다.
56. 스위치를 Off 상태로 셋팅한 후, 이전 단계를 다시 반복합니다. HiHi 알람이 지금은 보일 것입니다.

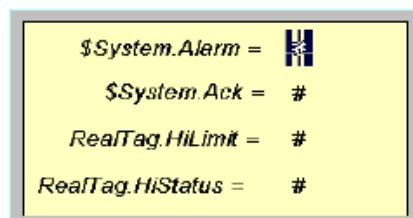


57. Development 모드로 다시 전환합니다.

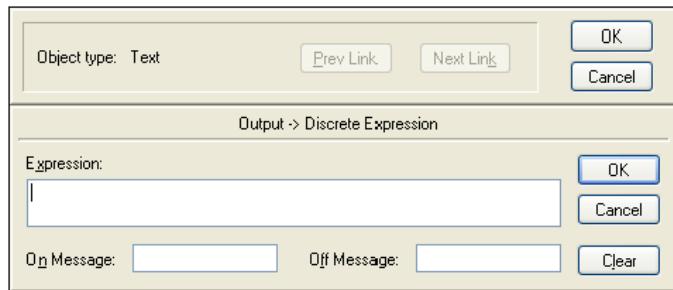
Lab 6c – 알람 Dot Fields

알람 Dot Fields 설정

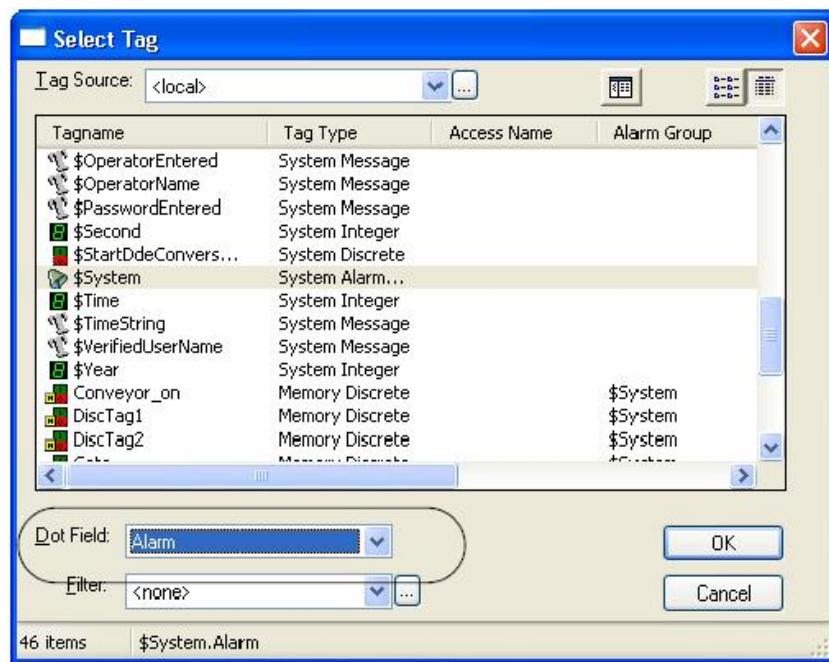
58. \$system.Alarm fields 의 # Symbol 을 더블-클릭합니다.



59. 애니메이션 링크 창에서 Value Display – Discrete 를 선택하면, Output Discrete Expression 창이 보이게 될 것입니다.



60. (Expression 문자란에 이전에 설정된 태그가 있다면 지운 후) Expression의 문자부분을 더블 클릭한 후, Select Tag 창을 엽니다.



61. \$system 태그를 선택합니다.

62. Dot-Field 부분에서 drop-down 리스트를 엽니다. 이 리스트에는 모든 dot-field 들이 보일 것입니다.

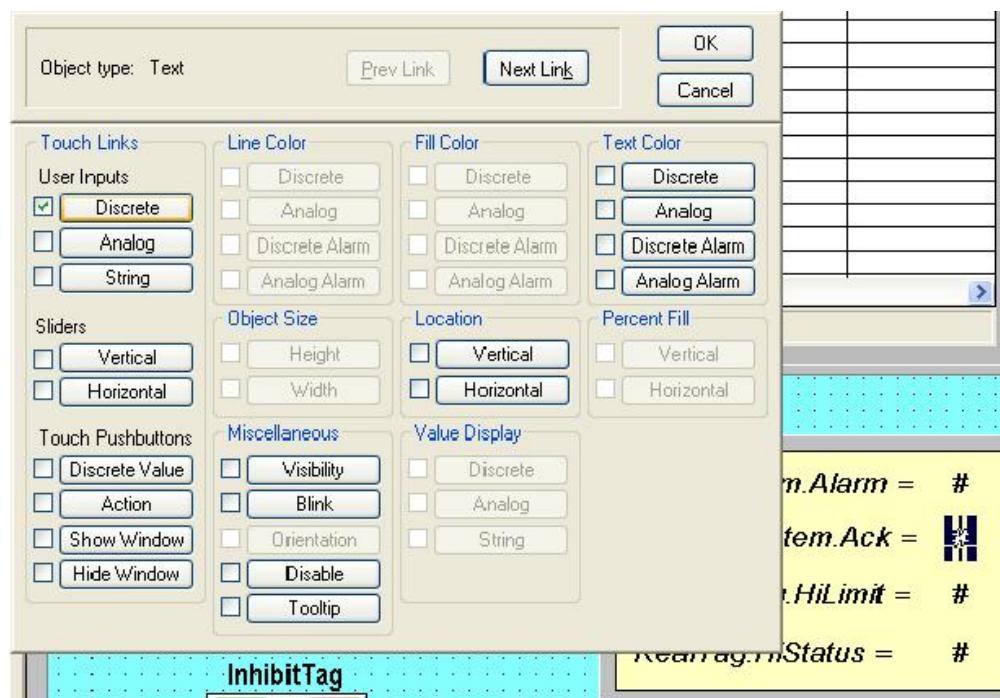
63. Alarm 이라는 Dot Field 를 선택합니다. Select Tag 창은 위에 보이는 그림과 같이 설정이 되었을 것입니다.

64. Ok 버튼을 클릭합니다. .Alarm 은 \$System 태그에 할당 된 것입니다.

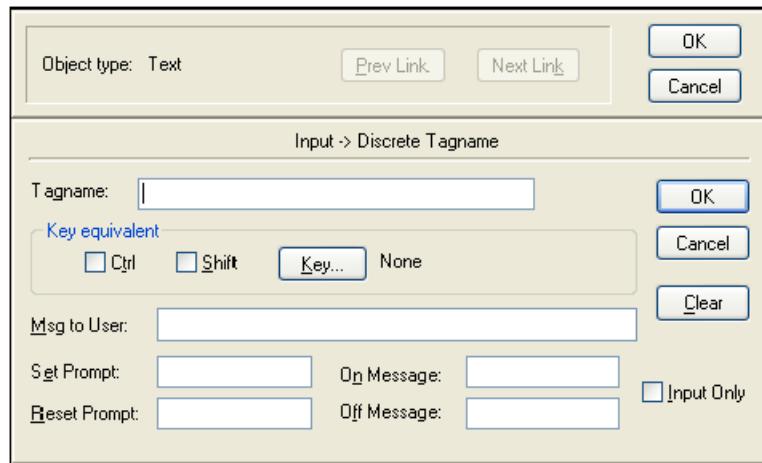
65. On Message 에는 Yes 를, Off Message 에는 No 를 입력합니다.



66. OK 버튼을 클릭합니다.
67. 윈도우에서 \$System.Ack에 인접한 # symbol을 더블-클릭합니다.
68. 애니메이션 링크 창에서 Toch Links / User Inputs – Discrete를 선택합니다.
- 69.

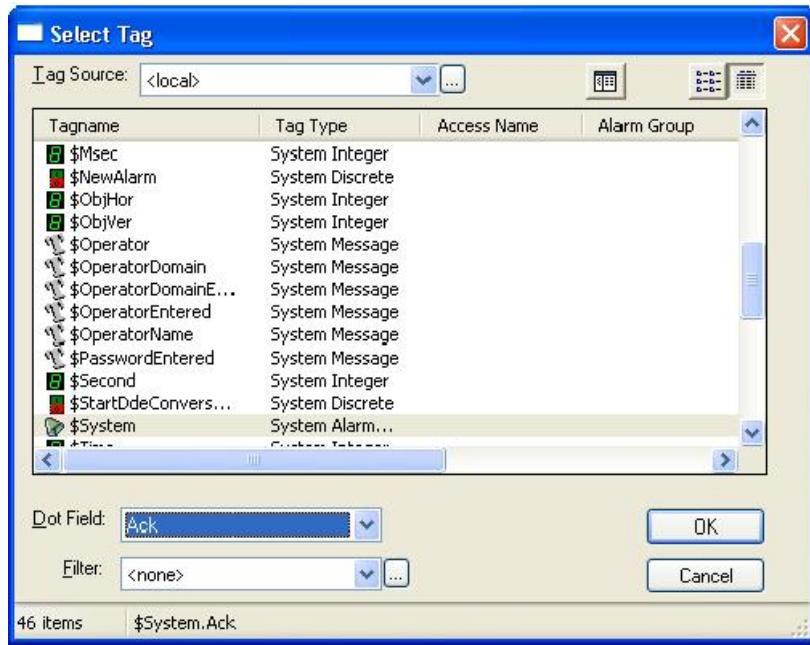


70. 이전에 설정된 태그명과 설정 값을 모두 지웁니다.



71. Tagname 의 빈영역을 더블-클릭합니다.

Select Tag 창이 보일 것입니다:

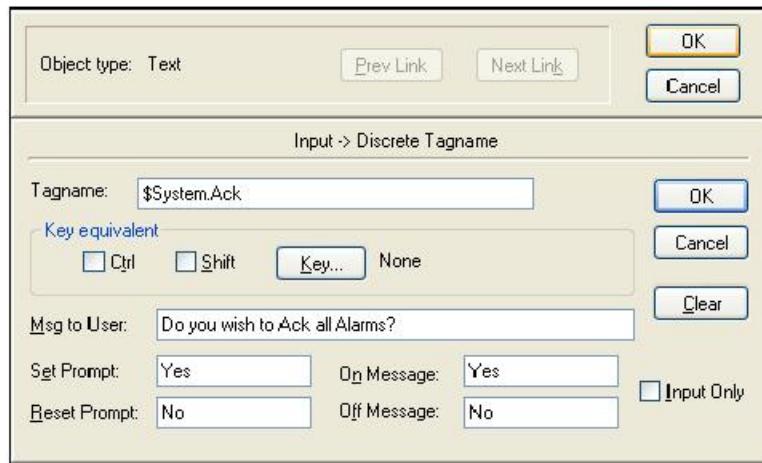


72. \$system 태그를 선택합니다.

73. Dot Field 의 drop-down 리스트를 열어서 ACK 를 선택합니다.

74. OK 버튼을 클릭합니다.

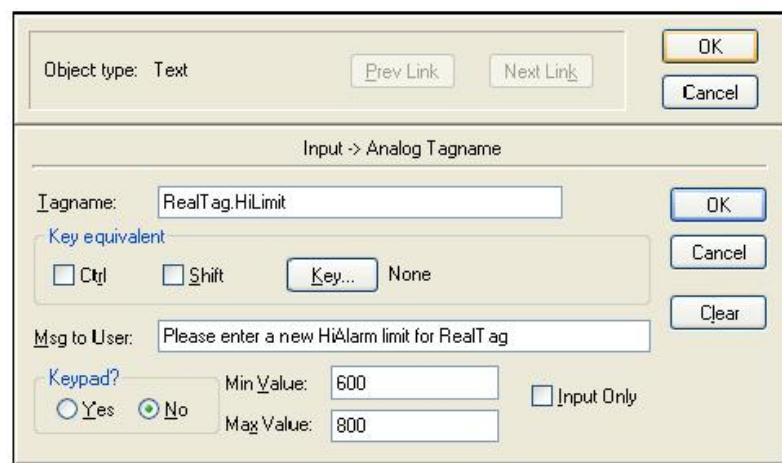
75. 다음 그림과 동일하게 애니메이션 링크의 설정을 합니다.



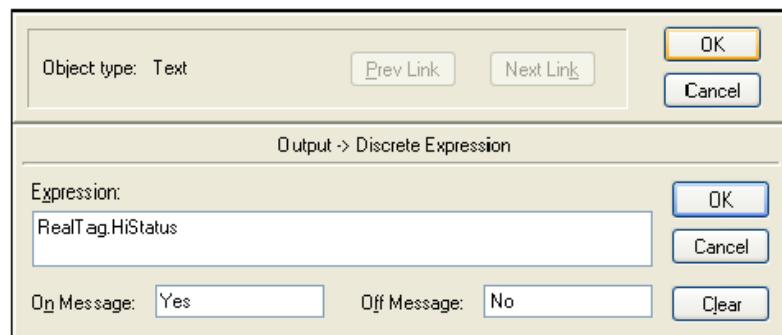
76. 최상위에 있는 OK 버튼을 클릭하면, 저장 된 후, 애니메이션 링크 창이 닫힙니다.

77. Touch Link / User Input – Analog 링크를 RealTag.HiLimit text Display 객체를 설정하기 위해서는 이전 단계를 참고하기 바랍니다.

78.



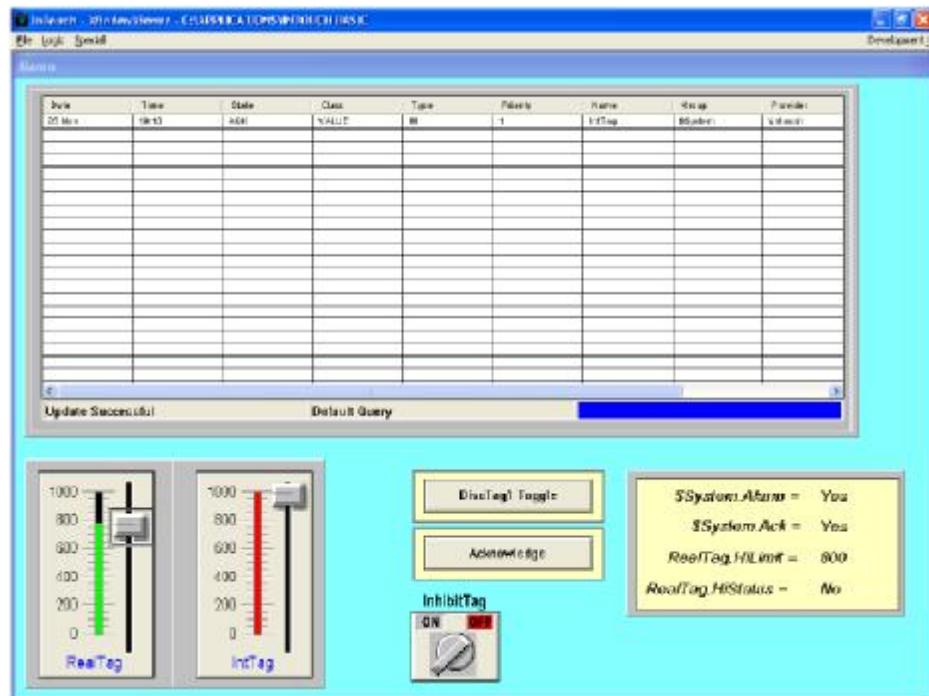
79. Valueview Display – Discrete 링크에서 RealTag.HiStatus 문자를 보기 위한 객체를 설정하기 위해선 이전 단계를 참고하기 바랍니다.



80. OK 버튼을 클릭합니다.

81. Dot-field 값의 변화를 확인하기 위해서, Runtime으로 전환합니다.

82. RealTag.HiLimit의 값을 변경시킵니다.

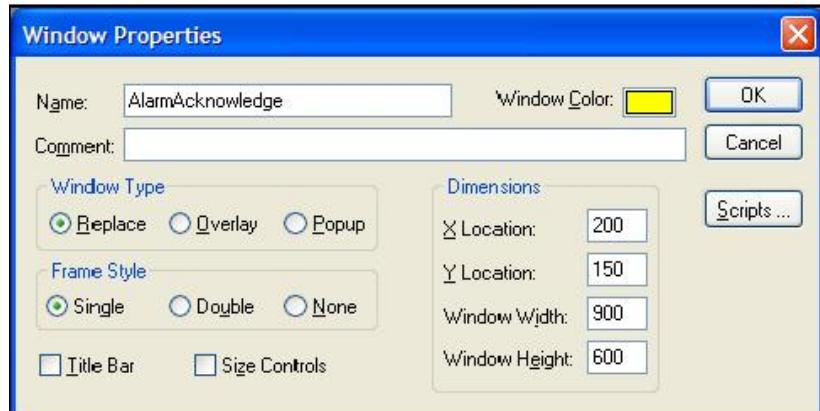


83. Development 모드로 전환합니다..

Lab 6d – 알람 발생시 경보 팝업창

82. New Window 버튼 을 클릭해서, 새로운 윈도우를 엽니다.

83. 다음 그림에서 보이는 것처럼 윈도우의 속성을 설정합니다.

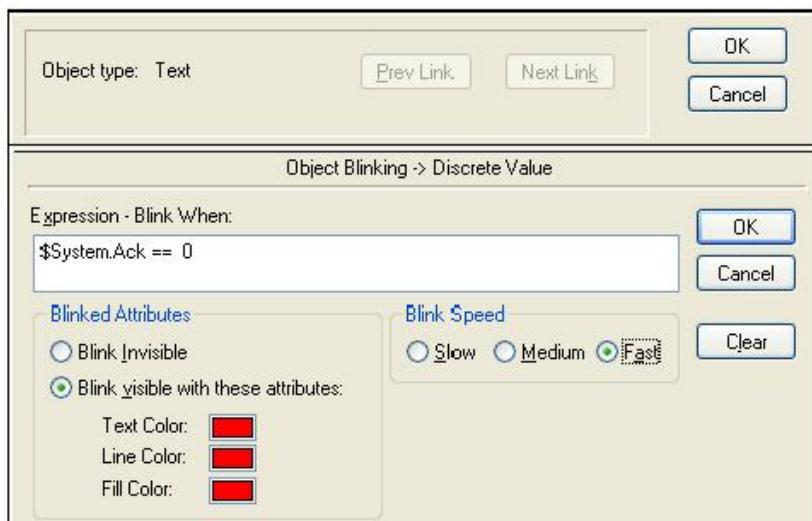


84. OK 버튼을 클릭합니다.

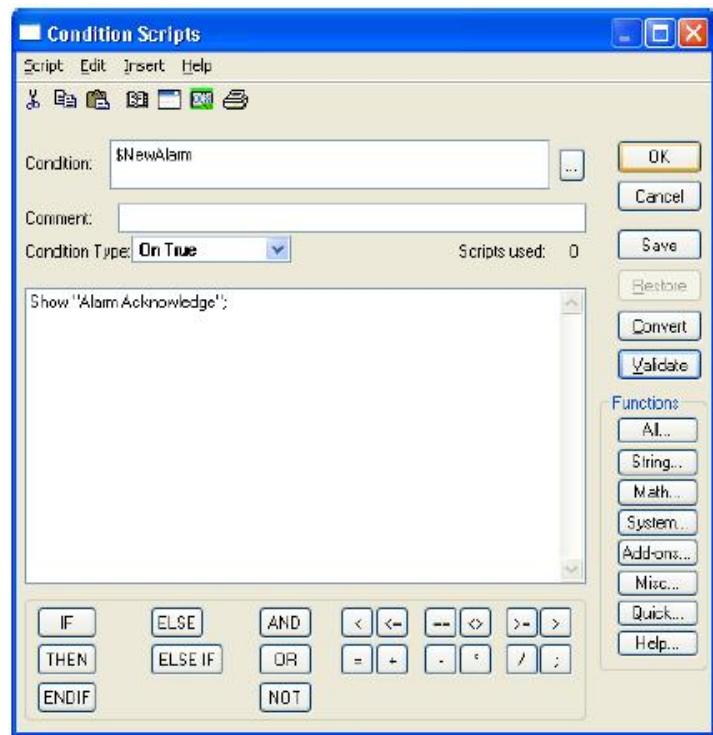
85. 문자, 버튼 객체를 추가하기 위해서 아래그림을 참고하기 바랍니다.



86. Attention 텍스트에 Blink 옵션을 사용하기 위해 아래 그림처럼 설정해 줍니다.

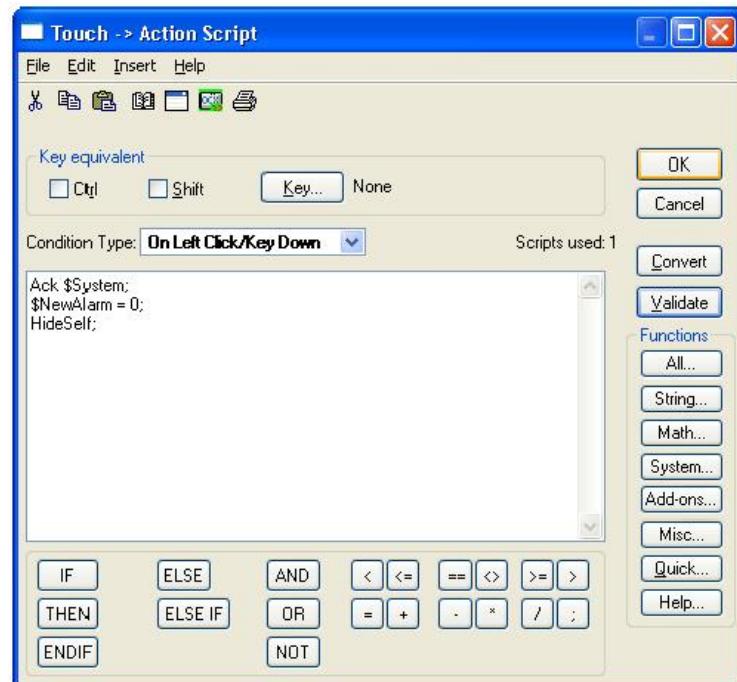


87. 새로운 Condition 스크립트를 생성한 뒤, On True Condition Type 의 아래 스크립트는 태그에 알람이 발생하게 되면, 윈도우 창이 열리게 합니다.

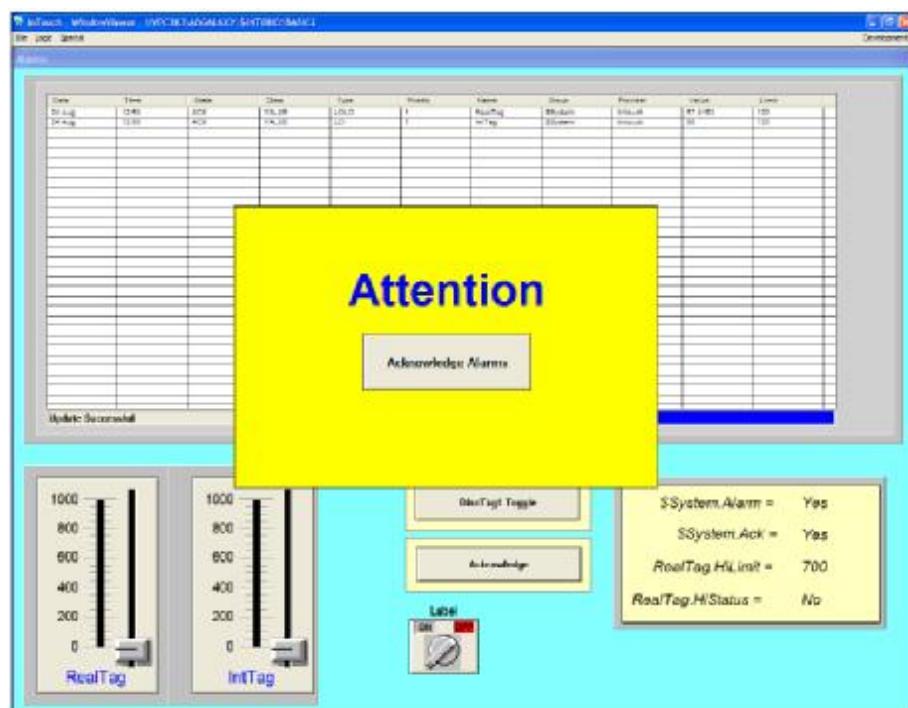


88. Validate 버튼으로 스크립트가 정확한지를 확인하고, OK 버튼을 클릭합니다.

89. Acknowledge Alarms 버튼에 아래 그림과 같이 Touch Pushbutton – Action 스크립트를 설정합니다.



90. Validate 로 스크립트를 확인한 뒤, OK 버튼을 클릭합니다.
91. OK 버튼을 클릭하고, 애니메이션 링크창을 닫습니다.
92. Alarm Acknowledge 윈도우를 저장하고 닫습니다. Alarm 윈도우가 열려있는지 확인합니다.
93. Runtime으로 전환합니다.
94. 슬라이더를 이동시킵니다. 태그에 알람이 발생하게 되면, Alarm Acknowledge 윈도우가 열리게 됩니다. Attention 문자 윈도우가 깜빡거리는 것을 볼 수 있습니다.
95. Acknowledge Alarms 버튼을 클릭합니다. 윈도우는 닫하고, 현재 발생한 알람이 ACK 상태로 변하는 것을 확인 할 수 있습니다.

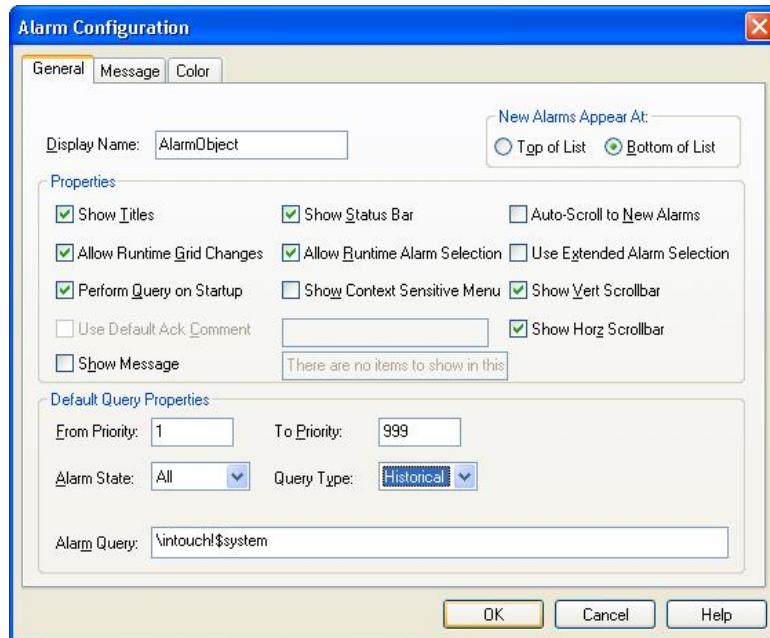


96. Development로 전환합니다.

Lab 6e – Event 모니터링

Create Events

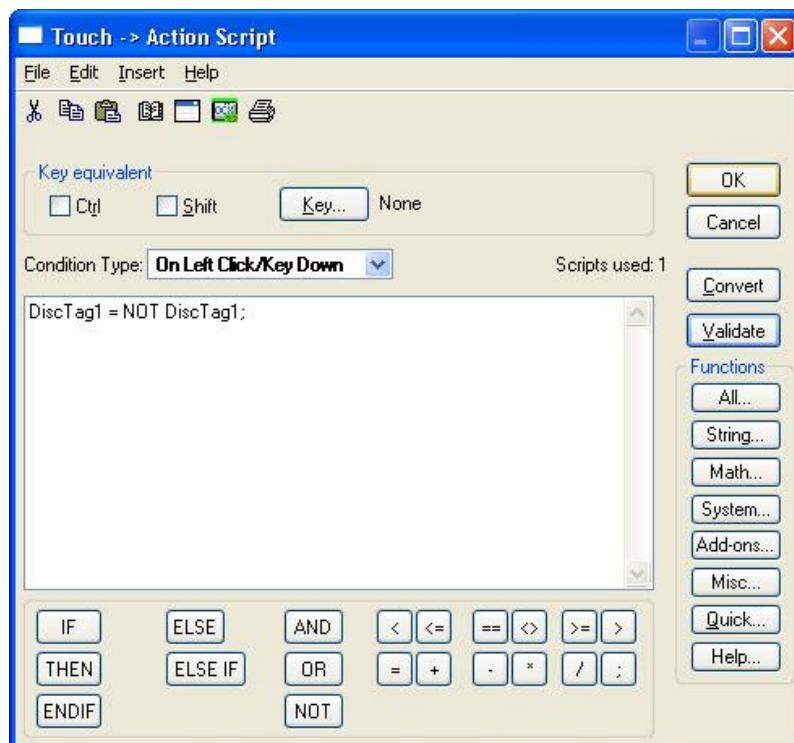
97. Alarm Display 객체를 더블-클릭합니다.
98. Default Query Properties 영역에서 Query Type 을 Historical로 변경합니다.



99. OK 버튼을 클릭합니다.

알람 Display 행은 Runtime 실행 시, 알람 발생함에 따라 행이 추가됩니다.

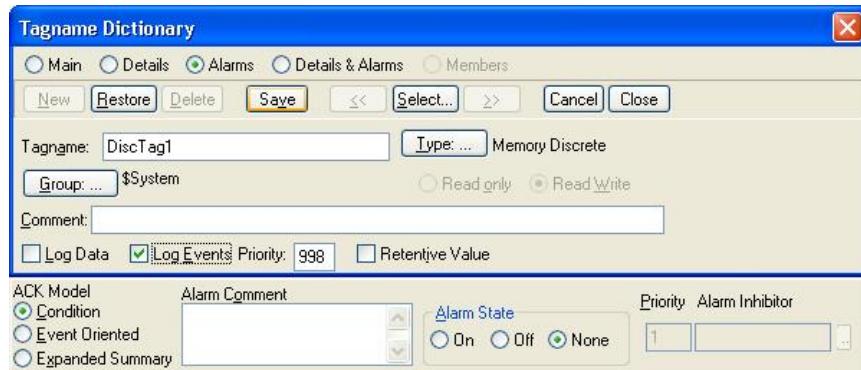
100. DiscTag1 버튼의 Touch Pushbuttons – Action 스크립트를 확인해 봅니다.



101.OK 버튼을 클릭한 뒤, 한번 더 OK 버튼을 클릭합니다.

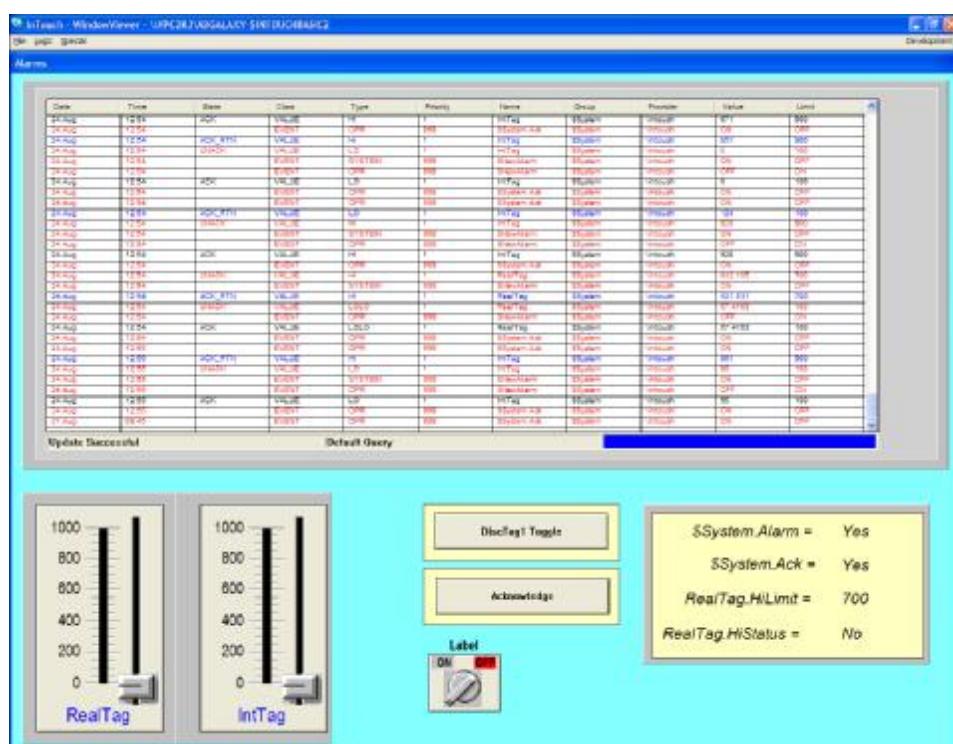
102. Tagname dictionary 을 엽니다.

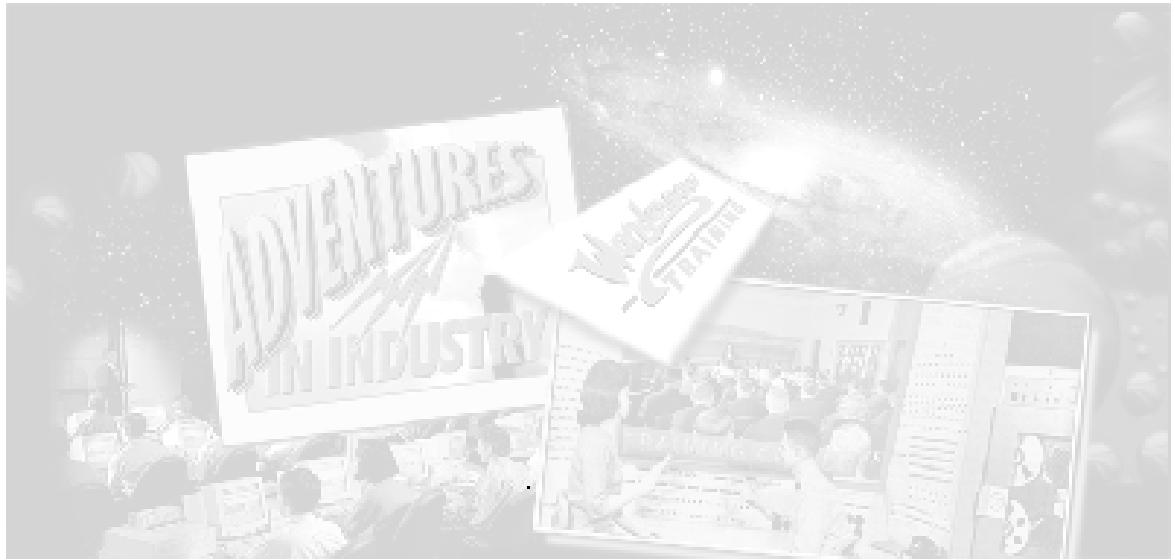
103. DiscTag1 에 우선순위 998 이 입력되었는지, Log Events 옵션은 체크가 되어 있는지를 확인합니다.



104. Runtime 으로 전환한 후, DiscTag1 Toggle 버튼을 클릭하고, 이벤트 상태가 보이는지를 확인합니다.

이 알람들은 Alarm Display 객체의 가장 아래에 나타날 것입니다.





Module 7

실시간 & 이력 트렌드

Section 1 – 실시간 트렌드

Lab 8 – 실시간 트랜드 설정

Section 2 – 이력 트렌드

Lab 9 – 이력 트랜드 설정

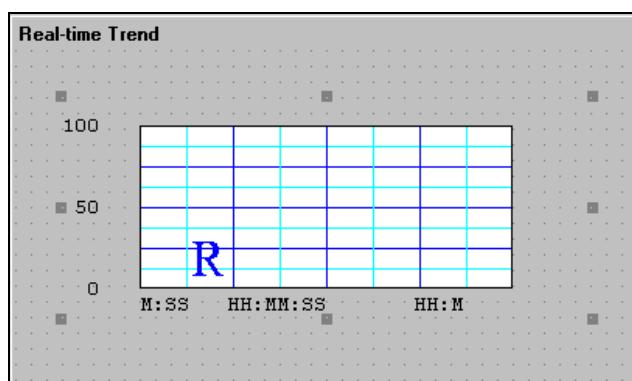
Section 1 - 실시간 트렌드

InTouch는 두 가지의 트렌드 디스플레이 방식을 제공 합니다. 실시간(Real-Time) 트렌드와 이력(Historical) 트렌드 입니다. 이 두 트렌드 객체 모두는 여러 개의 태그 데이터를 그래프를 이용하여 화면에 디스플레이 하도록 구성 해주는 기능 입니다.

실시간 트렌드는 4개의 펜(태그 데이터 값)까지 사용 가능하고, 이력 트렌드는 최대 8개의 펜을 사용 하실 수 있습니다. 이 두가지의 트렌드는 WindowMaker 내의 트렌드 tool 을 이용하여 만드실 수 있습니다. InTouch는 또한 트렌드 구성 콘트롤을 제공 합니다. 예를 들면, 시간의 범위를 설정 할 수 있고, 값의 범위, 시간 디스플레이 장소, 값 디스플레이 장소, 펜의 숫자 그리고 색의 속성 등 트렌드를 구성하는 콘트롤들 사용자 편의에 맞게 제공 하고 있습니다.

실시간 트렌드 작성하기

1. 실시간 트렌드 도구  를 WindowMaker 도구상자에서 선택합니다.
2. 윈도우에서 클릭하고, 마우스를 대각선으로 끌어서 원하는 트렌드의 사각 크기를 그립니다. 트렌드 차트는 선택한 어느 크기로도 그릴 수 있으며 화면에 놓을 수 있는 차트의 수에는 제한이 없습니다.
3. 마우스를 해제합니다. 실시간 트렌드가 윈도우에 나타납니다.

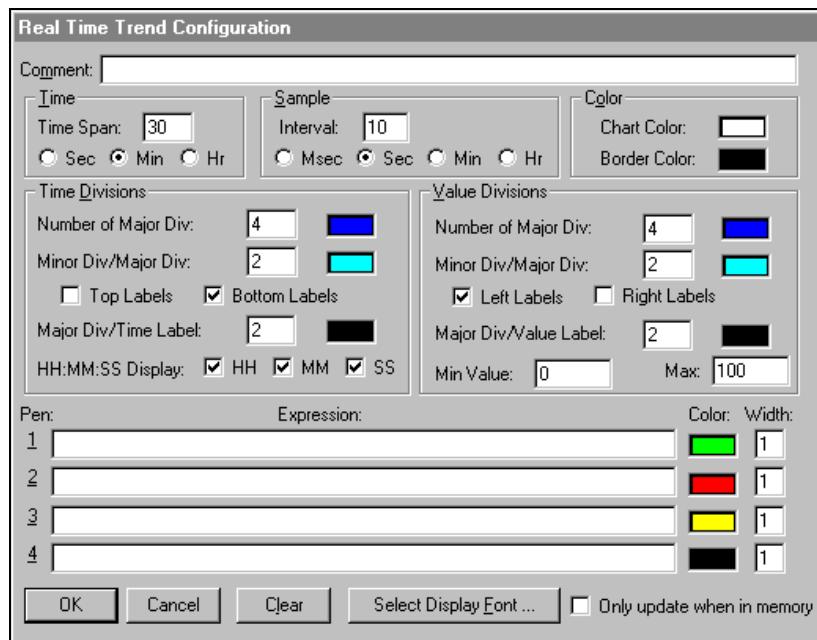


4. 트렌드를 두번 클릭하여 그것의 구성 대화상을 엽니다.

실시간 트렌드 구성하기

실시간 트렌드를 처음으로 그릴 때 시스템 기본 구성 설정은 다음과 같습니다.

- 트렌드에 두 번 클릭하여 **Real Time Trend Configuration** 대화상자를 엽니다.



Note: 실시간 트렌드 구성 대화상자에서 제작된 모든 항목은 트렌드 윈도우의 크기에 무관하고 실행 시간에 교정할 수 없습니다.

- Time Span** 필드에 트렌드에 대각선으로(x-축) 디스플레이 될 시간의 길이 값을 입력합니다. 라디오 버튼에 작성되어 있는 단위시간을 선택하시기 바랍니다. 예를 들어, **30** 이 입력되고 **Min** 이 선택되었다면, 차트의 수평 시간 범위는 30 분입니다.
- Sample Interval** 필드에 트렌드 차트가 업데이트 될 간격을 입력합니다. 번호가 관련될 시간 증분의 라디오 버튼을 선택합니다. 예를 들면, **10** 이 입력되고 **Sec** 가 선택되었으면, 식은 10 초 마다 계산될 것입니다.
- Color** 필드의 각 색 선택 상자를 클릭하여 트렌드의 배경과 패선 색을 선택합니다.
- Time Divisions** 그룹의 **Number of Major Div** 필드에 주 분산 트렌드의 수를 입력합니다. 선의 새로운 색을 선택하기 위해 색 상자에 클릭합니다. 분배의 수는 반드시 보조 분배/주 분배의 공배수여야 합니다.
- Minor Div/Major Div** 필드에 각 주 분배 내에서 보여야 할 보조 분배의 숫자를 입력합니다. 선의 새로운 색을 선택하기 위해 색 상자에 클릭합니다.

7. **Top Labels/ Bottom Labels** 체크 박스는 위/아래쪽의 레이블을 갖거나 혹은 레이블이 전혀 없을 수 있게 합니다. 트렌드를 위해 레이블을 구성한다면 **Major Div/Time Label** 상자에 주 분배 선 당 시간 레이블의 수를 입력합니다. 시간 레이블의 색을 선택하기 위해 색이 있는 사각형에 클릭합니다. 그리고 나서 원하는 시간 디스플레이 포맷을 선택합니다.
8. **Value Divisions** 그룹의 값 레이블은 **Time Divisions** 그룹과 같은 방법으로 구성합니다. 보조와 주 값 분배는 트렌드의 수직 값 (y-축) 범위를 설정합니다. 이 범위는 엔지니어링 단위를 사용하여 모든 트렌드된 태그명과 같습니다. 실행시에 보조 그리고 주 값 분배를 위한 소수점 자릿수를 디스플레이 하기 위해서 아래와 같은 포맷을 따라야 합니다. 예. 0.00에서 100.00
9. 트렌드 윈도우에 네 개 펜까지 사용 하실 수 있습니다. **Pen**은 어느 지역 데이터베이스 태그명 또는 한 개 이상의 태그명을 포함한 식을 디스플레이 하기 위해 사용합니다. **Expression**을 트렌드할 수 있는 능력은 태그명을 보여주기 위해 사용자 정의 디스플레이를 작성하기 위해 사용할 수 있습니다. 그 안에서 트렌드될 태그명 색은 동행하는 **Color** 사각형에 디스플레이 됩니다 - 새 색 선택을 위해 사각형에 클릭합니다. 펜의 넓이(픽셀)는 **Width** 항목 상자에서 구성될 수 있습니다. 펜의 넓이를 1 이상으로 하였을 경우, 컴퓨터의 성능에 따라 실행 속도의 저하를 보일 수도 있습니다.
10. **Select Display Font** 버튼을 클릭하여 차트를 인쇄할 때 사용하는 글꼴, 유형과 크기를 선택하기 위해 글꼴 대화상자를 사용합니다.
11. 현재 디스플레이 된 윈도우에 있을 때만 트렌드가 업데이트 되길 원하면 **Only update when in memory** 옵션을 선택합니다. 이 옵션을 선택하지 않으면, 차트는 윈도우가 열리지 않았어도 항상 업데이트 될 것입니다. 이것은 전체적인 시스템의 시스템 성능을 다소 낮추는 결과를 가져올 수 있습니다.

실시간 트렌드 성능 높이기

다음과 같이 실시간 트렌드 성능을 높일 수 있습니다.

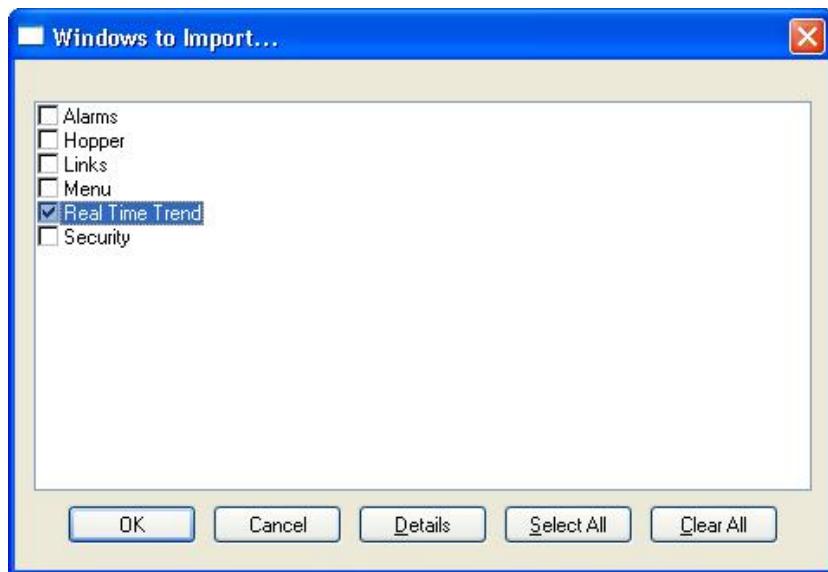
1. 펜 넓이를 '1'로 설정합니다.
2. 아무 다른 객체도 실시간 트렌드의 위쪽에 놓이지 않게 확인합니다.
3. 취해진 “표본”의 수를 줄입니다.

Lab 8 - 실시간 트랜드 설정

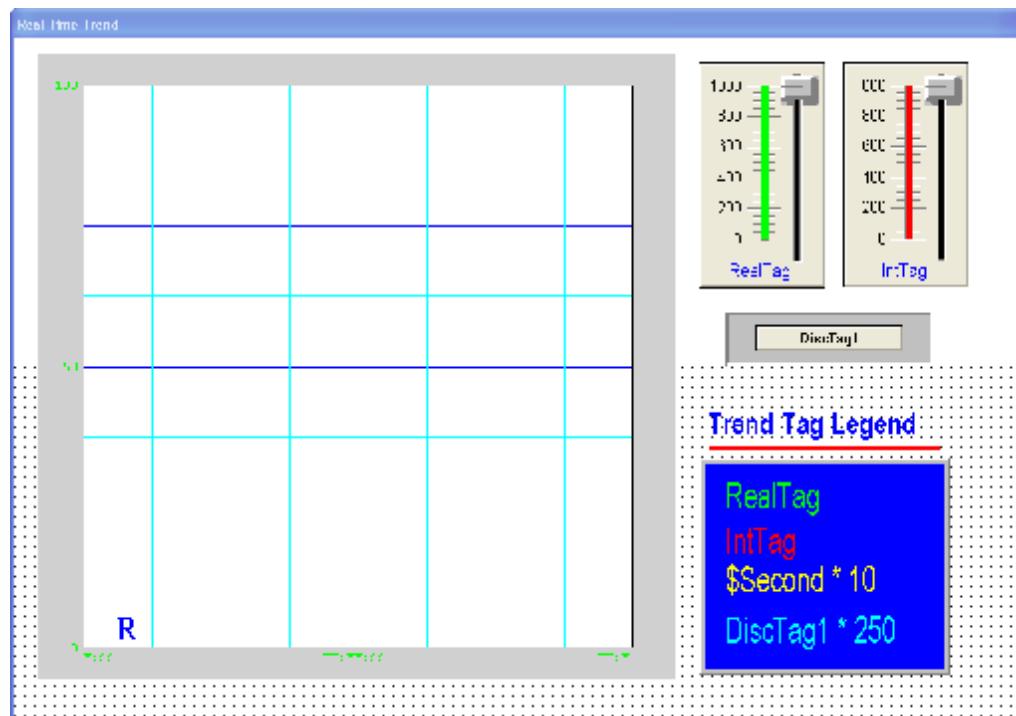
Lab Instructions

실시간 트렌드 윈도우 import 변환하기

1. WindowMaker에서 열려 있는 모든 윈도우를 닫고 Windowviewer를 종료합니다.
Wwimport 디렉토리로부터 Realtime Trend 윈도우를 import 합니다..



2. Real Time Trend 윈도우를 엽니다.
 3. 윈도우 위의 모든 객체를 선택하기 위해 F2를 누릅니다.
 4. 태그들을 Local 태그로 변환합니다.
5. Drawing toolbar안에 Real-Time Trend 툴 을 선택합니다.
6. 실시간 트렌드 윈도우의 회색펜에 Real-Time Trend 객체를 추가합니다.
 7. 패널에 맞게 Real-Time Trend의 크기를 조정합니다. . Real-Time Trend 윈도우는 지금 다음과 같은 그림으로 보여집니다.

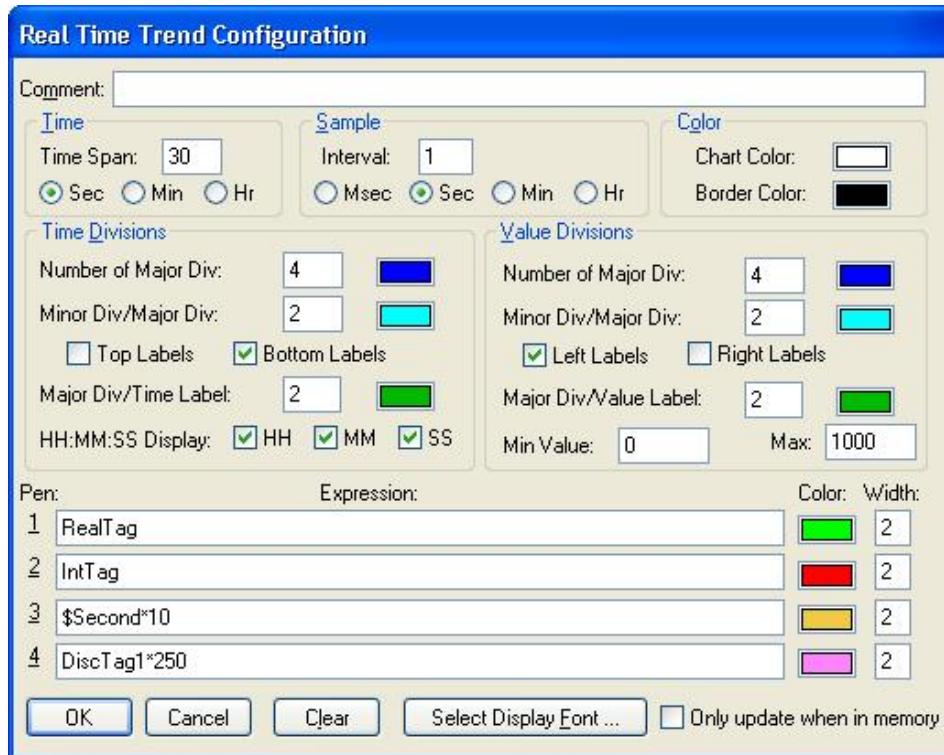


실시간 트렌드 설정

8. Real Time Trend를 더블클릭하여 Configuration 대화상자를 나타냅니다.
9. Time/Time Span를 30 초로 설정합니다.
10. Sample/Interval을 1초로 설정합니다.
11. Value Divisions/Max를 1000으로 설정합니다.
12. Pen 1에 Expression 필드를 더블클릭하고 RealTag를 선택합니다.
13. Pen 2에 Expression 필드를 더블클릭하고 IntTag를 선택합니다.

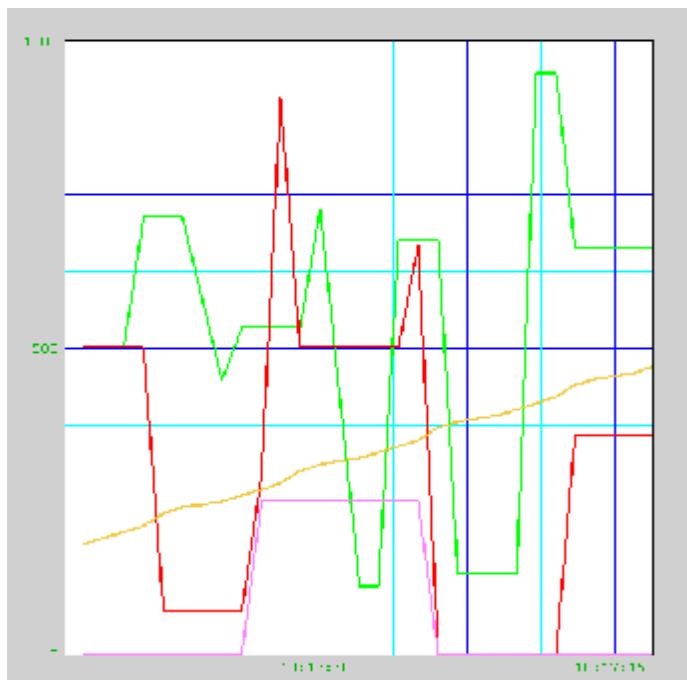
14. Pen3과 Pen4를 다음과 같이 입력합니다.
 Pen 3: $\$Second*10$
 Pen 4: $DiscTag1*250$

15. Pen Width를 2로 설정합니다.



16. OK를 클릭하고 Runtime 모드로 전환합니다.

17. 트랜드안의 값을 변화시키기 위해 슬라이드를 움직이고, DiscTag1 버튼을 클릭합니다.
슬라이드의 움직임이 반영된 트랜드선을 볼 수 있습니다.
(알람에 대한 지식이 필요 할지도 모릅니다.)



18. 개발모드로 전환합니다.

Section 2 – 이력 트렌드

이력 트렌드는 이전의 시간과 날짜로부터 데이터의 "Snapshot"을 제공합니다. 이력 트렌드는 동적으로 트렌드를 업데이트 하는 툴이 아닙니다. 실시간 트렌드와 달리 이력 트렌드는 스크립트의 실행이나 버튼 클릭하기와 같은 오퍼레이터의 명령을 통해서만 업데이트 됩니다.

디스플레이 될 트렌드의 수 제한 없이 한꺼번에 여덟 개의 펜 까지 트렌드 할 수 있습니다. 트렌드의 인터페이스를 디자인 하는데는 완전한 융통성이 있습니다. 스쿠터의 현재 위치에 따라 여러 데이터를 사용하기 위해 오퍼레이터가 "트렌드 위로 지나가는 스쿠터"를 작성할 수 있습니다. 예를 들면, 가시적인 데이터가 있는 트렌드의 한 영역에 스쿠터를 놓을 때, 모든 트렌드에 있는 데이터베이스 값 위치의 시간과 값을 돌려줍니다.

또한 스쿠터 사이나 최대에서 최소값 등의 데이터 사이를 확대 축소하는 버튼을 작성할 수 있습니다. 평균과 표준 편차가 전체 차트 또는 스쿠터 사이의 영역에 디스플레이 될 수 있습니다. 이력 또한 어느 시간의 양에 따라 화면 이동할 수 있습니다. 최소와 최대 앤지니어링 단위를 디스플레이 하기 위해 규모를 사용자가 정의할 수 있고 그것들을 .MinEU 와 .MaxEU 태그명 .field에 연결할 수 있습니다.

분산 이력 시스템은 이력 트렌드의 재생 능력을 확장하여 원격 기록 데이터베이스를 포함합니다. 이 시스템은 복수 이력 로깅 데이터베이스로부터의 정보를 단일 트렌드로 디스플레이 하도록 허용합니다.

Note : 태그명 데이터 사전에서 '**Log Data**' 옵션이 켜진 태그만을 이력 트렌드에서 사용합니다.

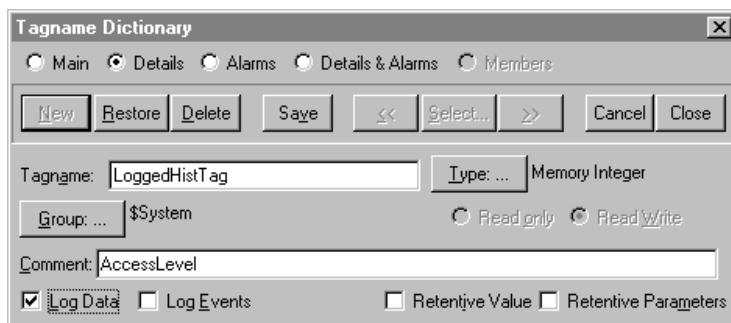
트렌드 기능과 함께, InTouch는 이력 저장 파일과 함께 수행 되도록 되어 있는 HistData Utility를 포함하고 있습니다. HistData Utility는 암호화 되어있는 이력 저장 파일(.LGH)을 MS Excel과 같은 스프레드 시트나 텍스트 편집 환경에서 볼 수 있는 콤마 분리 변수(.CSV)로 바꿔 줍니다.

이력 파일에 태그명 기록하기

값이 변할 때마다 그리고 기본으로 수정치와 상관없이 매 시간마다 값이 이력 파일에 쓰여집니다. 태그명 값이 이력 파일에 쓰이려면, **InTouch** 데이터베이스에 기록되도록 구성을 하셔야 합니다.

이력을 남기기 위한 태그사전 구성

1. Special / Tagname Dictionary 명령을 불러서 **Tagname Dictionary** 대화상자를 열고 원하는 태그명 정의를 디스플레이 합니다.

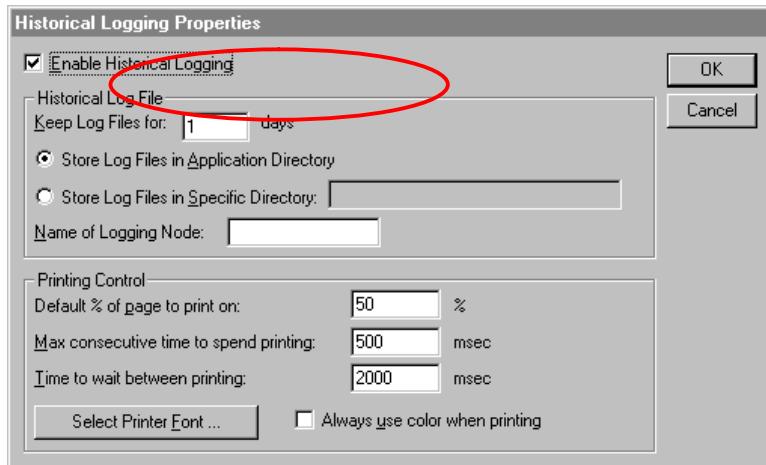


2. 일단 원하는 태그명 정의가 디스플레이 되면 **Log Data** 옵션을 사용합니다.

이력 기록 구성하기

Log Data 옵션을 가진 태그명이 이력 기록 파일에 쓰이게 하려면, 기록 기능이 사용 가능해야 합니다.

1. 기록을 가능하게 하려면 Special / Configure / Historical Logging 명령을 부릅니다. **Historical Logging Properties** 대화상자가 나타납니다.



2. **Enable Historical Logging** 옵션을 선택해서 이력 기록을 사용 가능케 합니다. **Keep Log Files for** 필드에 이력 파일이 디스크에 유지되어야 할 만큼의 날 수 (오늘 이전)를 입력합니다. 예를 들면, 0 이 입력되었으며 그 달의 12 일 이면, 2 일부터 12 일 까지 (10 일 플러스 오늘) 의 기록 파일이 보관되고 첫 파일은 자동적으로 삭제됩니다. 0 (영) 이 일에 사용되면, 기록 파일은 영원히 보관됩니다.

Note: InTouch는 매일 (24 시간) 2 개의 이력 저장 파일을 만들 것입니다. 그러므로, 이 값이 세팅 될 때 반드시 디스크의 공간이 고려되어야 합니다. 만약 이력 저장 파일을 저장하기 위해 충분한 공간이 있지 않을 경우, 저장은 멈춥니다. 따라서, 공간을 확보 한 후 다시 저장을 시작 해야 합니다. 이력 저장은 버튼이나 스크립트를 통해 내부 태그명 \$HistoricalLogging 에 연결 함으로서 또는 WindowViewer 안에 Restart Historical Logging 명령을 이용해서 실행 시에 시작하고 멈출 수 있습니다.

3. **Store Log Files in Application Directory** 옵션을 선택해서 이력 데이터가 어플리케이션 디렉토리에 기록되게 합니다. **Store Log Files in Specific Directory** 라디오 버튼을 선택해서 이력 데이터가 명시된 디렉토리에 기록되도록 하기 위해 상자 안에서 경로 이름을 입력합니다. 이 항목은 UNC (\Node\Share\Path) 또는 직접 경로 (C:\Path) 일 수 있습니다.

이 파일들은 선택한 디렉토리에 배치되고 자동적으로 다음과 같이 명명합니다.

YYMMDD00.LGH 와 YYMMDD00.IDX

해당 사항: YY 는 파일이 작성된 해

MM 은 파일이 작성된 달 (01-12)

DD 은 파일이 작성된 날 (01-31)

00 은 항상 화면에 0 을 나타냅니다

예를 들면, 파일들이 1995년 9월 30일에 작성되었다면, 다음과 같이 명명합니다.

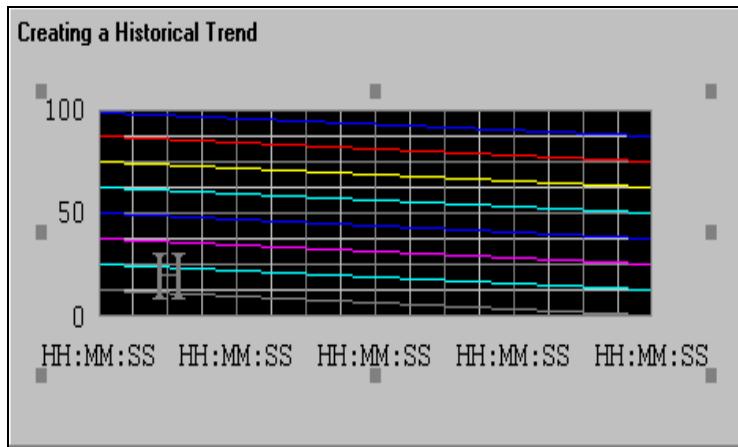
95093000.LGH 와 95093000.IDX

Note: InTouch의 이번 버전은 .LGH 와 .IDX 의 확장성을 가진 새로운 저장 파일을 지원합니다. 이전 버전의 InTouch는 저장 파일을 위해서 .LOG 파일을 사용했습니다.

4. **Name of Logging Node** 필드에 이력 파일에 기록될 노드의 이름을 입력합니다.
컴퓨터 노드 이름이 아닌, NetDDE 노드 이름을 입력합니다.
5. **Default % of page to print** 상자 안에, 트렌드를 위한 펜이지 크기 퍼센트를 넣습니다.
예를 들면
이력 트렌드를 프린트 할 때, 만약 50%를 사용 했다면, 페이지의 반 만을 채울 것입니다.
6. **Max consecutive time to spend printing** 필드에 이력 트렌드 인쇄 모듈이 인쇄하는데 사용할 프로세스 시간(밀리 초)을 입력합니다.
7. **Time to wait between printing** 필드에 이력 트렌드 인쇄 모듈이 다른 프로세서 시간을 취하기 전에 기다릴 시간(밀리 초)을 입력합니다.
8. 만약 컬러 프린터나 플로터를 사용 하고 있다면 **Always use colors when printing** 을 선택 합니다.
9. **Select Printer Font** 버튼을 클릭하여 디스플레이 된 데이터의 글꼴, 그것의 유형과 크기를 선택하기 위해 글꼴 대화상자를 열기위해 이 버튼에 클릭합니다.

이력 트렌드 작성하기

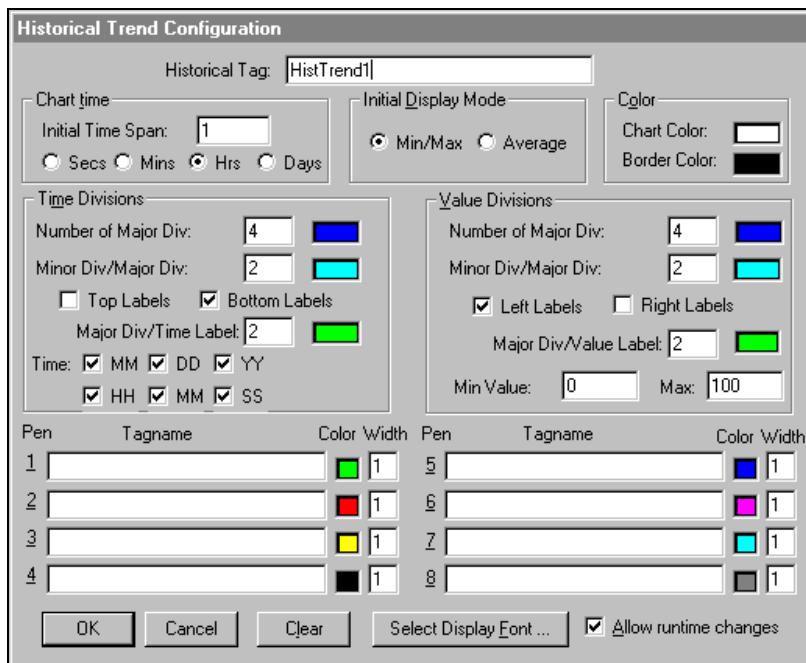
1. 이력 트렌드 도구  를 WindowMaker 도구상자 안에서 선택합니다.
2. 윈도우에서 클릭하고 마우스를 대각선으로 끌어서 원하는 트렌드 크기의 사각형을 그립니다. 어느 크기의 트렌드 차트를 그릴 수 있고 화면에 놓을 수 있는 차트의 수에는 제한이 없습니다.
3. 마우스를 해제합니다. 이력 트렌드가 윈도우에 나타납니다.



이력 트렌드 구성하기

이력 트렌드를 처음 그릴 때 시스템 기본 구성 설정이 보입니다. 그 후부터는 이전에 그려진 트렌드의 설정이 디스플레이 됩니다.

1. 트렌드에 두 번 클릭하여 **Historical Trend Configuration** 대화상자를 엽니다.



2. **Historical Tag** 필드에 트렌드 태그명을 입력합니다. 만약 태그명이 현재 태그명 사전에 정의 되어 있지 않다면, 새로 만들 것인지에 대한 질문을 받게 됩니다. 태그명은

태그 유형이 **Hist Trend**로 정의되어질 것이고 이는 필수사항입니다. **InTouch**는 자동적으로 기본으로 태그명에 **Hist Trend** 유형으로 정합니다. 각 이력 트렌드 차트에 다른 이름을 작성하는 것을 강력히 권합니다.

3. **Initial Time Span** 상자에, 트렌드 위에 수직 적으로 보여 줄 시간의 길이를 넣고, 시간 길이를 위한 시간의 증가 분을 선택 합니다.

4. **Initial Display Mode** 부분은 명명된 차트의 디스플레이의 초기 유형을 선택 합니다.

Min/Max - 차트에 있는 각 픽셀은 그 픽셀에 의해 표시된 시간에서 포인트가 포함한 최소에서 최대 범위를 디스플레이 합니다.

Average - 차트는 각 픽셀의 평균 값을 나타냅니다.

5. **Chart Color** 색 상자를 클릭하여 차트의 색을 선택합니다.

6. **Border Color** 색 상자를 클릭하여 차트의 패션 색을 선택합니다.

7. **Time Divisions** 그룹의 **Number of Major Div** 필드에 주 분산 트렌드의 수를 입력합니다. 선의 새로운 색을 선택하기 위해 색 상자에 클릭합니다.

8. **Minor Div/Major Div** 필드에 각 주 분배 내에서 보여야 할 보조 분배의 숫자를 입력합니다. 선의 새로운 색을 선택하기 위해 색 상자에 클릭합니다.

9. **Top Labels/ Bottom Labels** 체크 박스는 위/아래쪽의 레이블을 갖거나 혹은 레이블이 전혀 없을 수 있게 합니다. 트렌드를 위해 레이블을 구성한다면 **Major Div/Time Label** 상자에 주 분배 선 당 시간 레이블의 수를 입력합니다. 시간 레이블의 색을 선택하기 위해 색이 있는 사각형에 클릭합니다. 그리고 나서 원하는 시간 디스플레이 포맷을 선택합니다.

10. 만약 시간 레이블을 사용 한다면, **Time Divisions** 그룹의 **Major Div/Time Label** 상자 안에 major time division 라인 당 시간 레벨의 숫자를 넣습니다.

11. **Time Divisions** 그룹 안에, major time division 라인을 위한 색을 선택 합니다.

12. **Value Divisions** 그룹의 값 레이블은 **Time Divisions** 그룹과 같은 방법으로 구성합니다. 보조와 주 값 분배는 트렌드의 수직 값 (y-축) 범위를 설정합니다. 이 범위는 엔지니어링 단위를 사용하며 모든 트렌드된 태그명과 같습니다.

13. **Expression** 상자 안에, 트렌드 될 Pen에 지역 태그명 또는 식을 넣습니다. 8 개의 펜 가지 트렌드 화면에 보여 질 수 있습니다.

각 펜의 트렌드 색은 **Color** 가 있는 사각형에 디스플레이 됩니다; 사각형에 클릭하여 새로운 색을 선택합니다. 펜의 넓이 또한 **Width** 입력 상자에서 구성할 수 있습니다. 그러나, 펜 넓이가 1 보다 큰 것을 선택하면, 성능에 큰 영향을 줍니다.

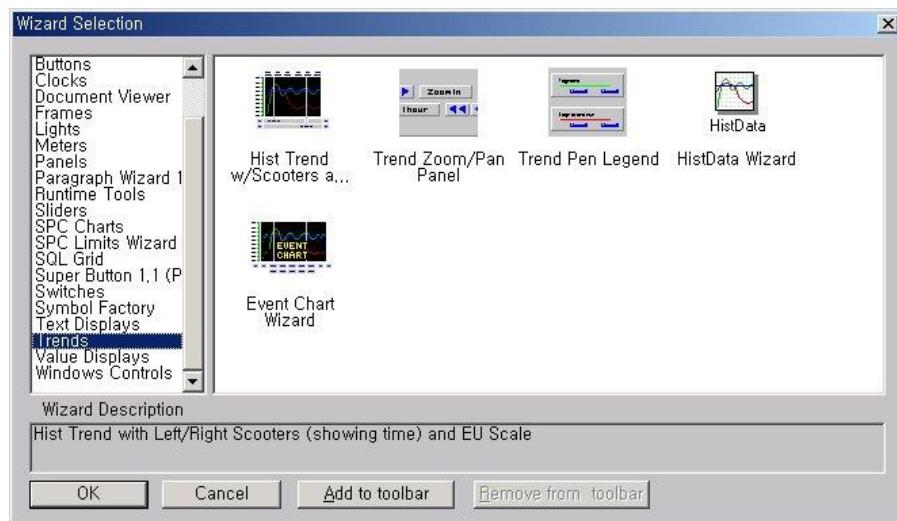
14. **Select Display Font** 버튼을 클릭하여 글꼴 대화상자를 열어 트렌드를 인쇄할 때 사용할 글꼴, 그것의 유형과 크기를 선택합니다.

15. 실행시간 동안 오퍼레이터가 펜 지정 변경, 시작 날짜와 시간 변경, 등의 변경을 트렌드 구성에 할 수 있게 허용하려면 **Allow runtime changes** 옵션을 선택합니다. 이것을 사용하면서 오퍼레이터가 실행시간에 트렌드에 두 번 클릭하면 **Runtime Setup** 대화상자가 나타나고 그렇게 해서 오퍼레이터는 그의 선택을 입력할 수 있게 됩니다.

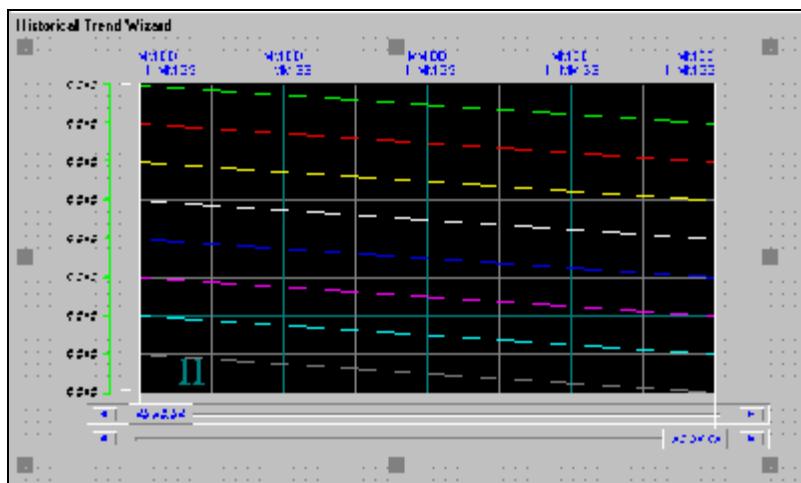
이력 트렌드 마법사 사용하기

InTouch는 트렌드 마법사라는 이력 트렌드를 작성하는 빠르고 쉬운 방법을 제공합니다. 트렌드 마법사는 스쿠터, 확대/축소, 등과 같은 모든 기능을 가진 이력 트렌드를 몇 번 마우스를 클릭함으로 구성할 수 있습니다.

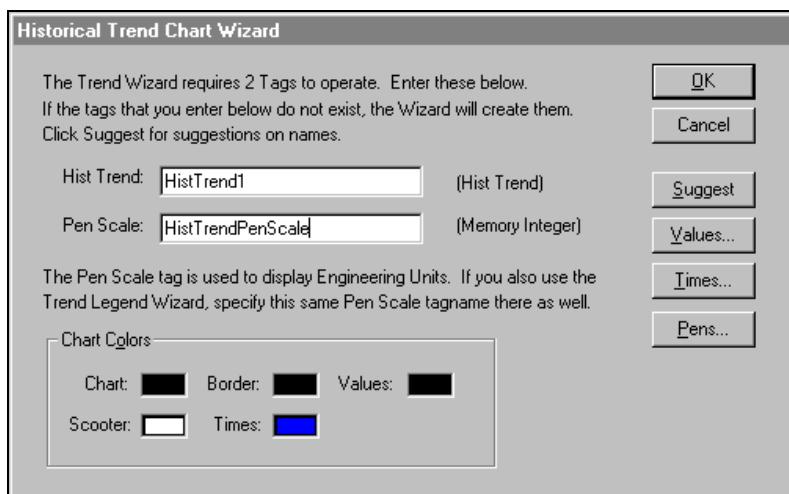
1. 도구상자에 있는 마법사 도구 를 클릭하여 **Wizard Selection** 대화상을 엽니다.



2. 마법사 목록에서 **Trends**를 선택합니다. 사용할 수 있는 이력 트렌드 마법사가 미리 보기 영역에서 목록의 오른쪽에 나타납니다.
3. **Hist Trend with Scooters** 마법사를 두 번 클릭하거나 또는 그것을 선택한 후 **OK**을 클릭합니다. 대화상자는 닫히고 InTouch 윈도우가 "붙여 넣기" 모드의 커서로 다시 나타납니다.
4. 윈도우에서 클릭하여 트렌드 마법사를 붙여 넣습니다.



5. 트렌드 마법사에 두 번 클릭하여 **Historical Trend Chart Wizard** 구성 대화상을 엽니다.

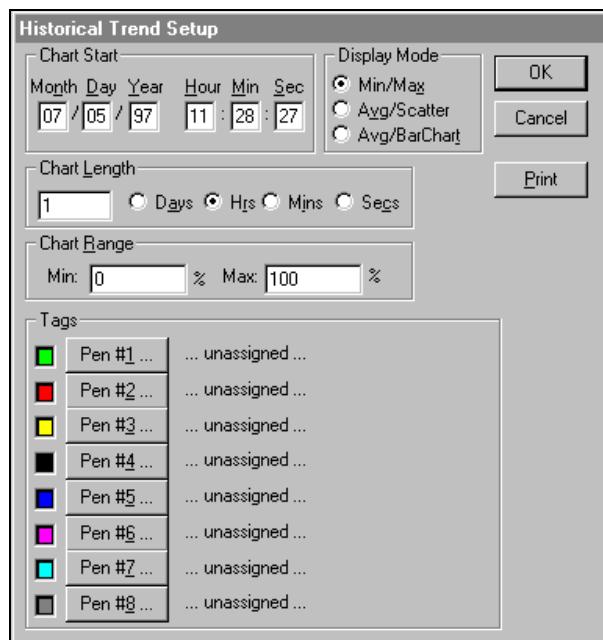


6. 트렌드를 구성하기 위해 필요한 정보를 입력한 후 **OK**에 클릭합니다. **Suggest**에 클릭하여 마법사로 하여금 이 정보를 채우게 합니다.
7. 확대/축소와 이동 기능 또는 트렌드의 팬 제어를 추가하려면, 각각 트렌드 **Zoom/Pan Panel** 및 **Trend Pen Legend** 마법사를 사용합니다. 이 구성요소들이 모두 함께 움직이려면, 모두 같은 이름이어야 합니다.

실행시간 중 이력 트렌드 구성하기

이력 트렌드를 구성할 때 **Allow runtime changes** 옵션을 사용 가능하게 하면 차트는 **WindowViewer**에서 접속에 민감하게 됩니다. 오퍼레이터는 펜 지정 변경, 시작 날짜와 시간 변경 등을 할 수 있습니다.

1. **WindowViewer**에서 트렌드에 클릭하면 **Historical Trend Setup** 대화상자가 나타납니다.



2. 차트의 시작 날짜와 시간을 **Chart Start** 필드에 입력합니다.

3. **Display Mode** 필드에서 차트의 화면보기 모드를 선택합니다. 이것들은 트렌드 디스플레이의 세 모드입니다.

Min/Max Historical Trend

이 모드는 트렌드나 변화를 엔지니어링 단위 배율의 퍼센트로 수정의 양 보다는 시간 흐름과 변경율을 강조하여 시간을 두고 수직 선으로 화면에 나타냅니다

Average/Scatter Historical Trend

이 예는 그 시간 간격 동안에 포인트의 평균 값을 보여 줍니다.

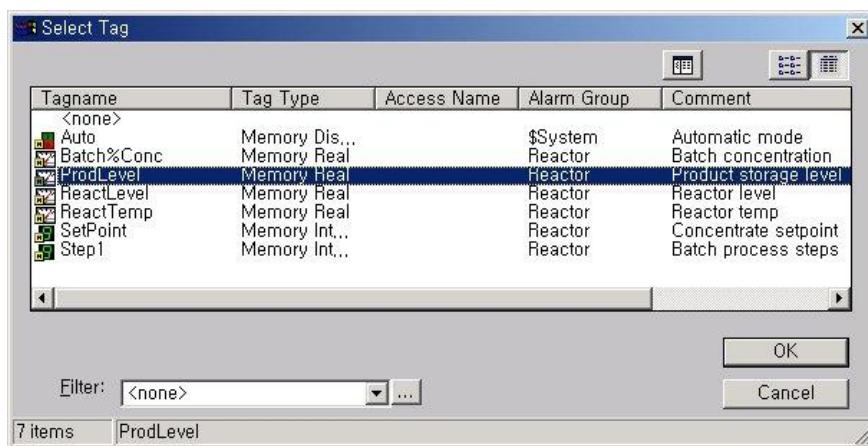
Average/Bar Chart Historical Trend

이 차트는 그 시간 간격 동안에 포인트의 평균 값을 막대 형식으로 보여 줍니다.

Chart Length 필드에 트렌드에 디스플레이 되어야 할 시간의 수평 (x-축) 길이를 입력합니다. 길이의 시간 증분을 선택합니다. 예를 들면, **1** 을 입력하고 **Hrs** 를 선택하면, 트렌드는 1 시간 동안입니다.

4. **Chart Range** 필드에 트렌드에 화면 표시되어야 할 시간의 수직 (y-축) 범위를 입력합니다. (범위의 단위는 EU 배율의 퍼센트입니다. 이들 값은 **0** 부터 **100** 까지입니다.

5. **Pen#** 버튼을 클릭하여 트렌드할 태그명을 선택합니다. **Select Tag** 대화상자가 나타납니다. 원하는 펜의 태그명에 클릭합니다.



단지 Log Data 옵션이 선택 되어져 정의된 태그명만 이 화면에 보여 집니다.

6. 태그명을 더블 클릭하거나 태그명을 선택 한 후 OK를 클릭 합니다.

실행시간 중 이력 트렌드 업데이트하기

이력 트렌드가 처음 디스플레이 되었을 때 **WindowViewer**에서 명시된 구성의 데이터를 디스플레이 할 것입니다. 실시간 트렌드와 달리 이력 트렌드는 계속해서 그들 자신을 업데이트하지 않습니다. 처음 데이터가 디스플레이 된 후에는 트렌드에 변경사항이 있어야만 업데이트가 됩니다. 트렌드를 업데이트 하기 위해서는 다음 중 어느 방법을 사용할 수 있습니다.

1. **Historical Trend Configuration** 대화상자(**WindowMaker**에서)에서 Allow runtime changes 옵션을 사용 가능케 하여 오퍼레이터가 수동적으로 트렌드 시간 그리고/또는 날짜를 수정하여 업데이트가 되게끔 합니다.
2. 다음을 스크립트 또는 푸쉬버튼에서 사용하여 오퍼레이터가 차트를 업데이트를 허용합니다.
TrendName.UpdateTrend = 1

3. 다음 각각을 스크립트 또는 푸쉬버튼에서 사용합니다.

```
HTUpdateToCurrentTime(Hist_Tag);  
HTScrollLeft(Hist_Tag,Percent);  
HTScrollRight(Hist_Tag,Percent);  
HTZoomIn(Hist_Tag,LockString);  
HTZoomOut(Hist_Tag,LockString);  
HTSetPenName(Hist_Tag,PenNum,Tagname);
```

4. 다음의 트렌드 태그 도트 필드를 수정합니다.

```
.ChartStart  
.ChartLength  
.MaxRange  
.MinRange  
.Pen1-.Pen8
```

이력 트렌드 .Fields

이력 트렌드 태그명에만 적용하는 많은 **.fields** 가 있습니다. 각 이력 트렌드 **.field** 는 아래에 간단히 설명되었습니다.

.Field	설명
.ChartLength	읽기/쓰기 정수 태그명 .field 로 이력 트렌드 그래프에 디스플레이 된 시간의 길이를 제어하기 위해 사용합니다. 차트의 길이를 초로 화면에 나타냅니다.
.ChartStart	읽기/쓰기 정수 태그명 .field 로 시작 시간을 제어 및/혹은 대응하는 이력 트렌드를 화면 이동하기 위해 사용합니다. 1/1/70, 오전 12:00 이후의 시간 경과를 초로 디스플레이 합니다.
.DisplayMode	읽기/쓰기 아날로그 태그명 .field 로 트렌드 값을 화면에 나타내는데 사용할 방법을 결정하는데 사용합니다.
.Max / MinRange	읽기/쓰기 실수 태그명 .fields 로 각 트렌드 된 태그를 위해 디스플레이 되어야 할 태그명의 엔지니어링 단위 범위 퍼센트를 대표하는데 사용합니다. .MaxRange 와 .MinRange 의 한계는 0에서 100이고 .MinRange 는 항상 .MaxRange 보다 적어야 합니다. 만약 0 보다 더 적거나 100 보다 더 큰 값이 둘 중 하나의 필드에 지정되면, 값은 0이나 100에 고정됩니다. .MinRange 가 .MaxRange 이상이면, 트렌드는 어느 데이터도 디스플레이 하지 않습니다.
.Pen1 - .Pen8	읽기/쓰기 TagID 유형 태그명 .fields 로 각 펜에 의해 이력 트렌드 된 태그명을 제어하기 위해 사용합니다. TagID 유형 태그명은 다른 TagID 태그명에 <u>만</u> 지정할 수 있습니다. .TagID 필드 확장명이 다른 태그명에 추가되지 않는 한 다른 태그명 유형과 섞여질 수 <u>없습니다</u> . 이것은 원격 이력 제공자 태그를 지정하는데 사용할 수 없습니다.
.ScooterLockLeft	읽기/쓰기 이산 필드입니다. 이 필드의 값이 참이면, 오른쪽 스쿠터는 왼쪽 스쿠터 위치의 왼쪽으로 이동할 수 없습니다. (0=거짓, 1=참).
.ScooterLockRight	읽기/쓰기 이산 필드입니다. 이 필드의 값이 참이면, 왼쪽 스쿠터는 오른쪽 스쿠터 위치의 오른쪽으로 이동할 수 없습니다. (0,1).
.ScooterPosLeft	읽기/쓰기 실제 필드로 왼쪽 스쿠터의 위치를 나타낸다 (범위 0.0에서 1.0).
.ScooterPosRight	읽기/쓰기 실제 필드로 오른쪽 스쿠터의 위치를 나타냅니다 (범위 0.0에서 1.0).

.TagID	읽기/쓰기 TagID 태그명 .field로 이력 트렌드 .Pen1 - .Pen8 TagID 태그명과 유합하여 태그명이 펜으로 트렌드되는 것을 감시 그리고/또는 제어하는데 사용합니다.
.UpdateCount	읽기전용 정수 필드로 트렌드의 읽기가 다 끝났을 때 증가합니다.
.UpdateInProgress	읽기전용 이산 필드로 이력 트렌드 데이터 읽기 상황 (0=읽기 진행 중 아님, 1=읽기 진행 중)을 보여 줍니다.
.UpdateTrend	읽기/쓰기 이산 태그명 .field로 모든 현재 값을 사용하여 이력 트렌드를 업데이트하도록 1로 설정될 수 있습니다.

이력 스크립트 함수

각 펜에 의해 트렌드 된 태그명은 명세, 스쿠터 위치의 값을 디스플레이, 퍼센트에 의해 트렌드를 이동하기 위해 사용할 수 있는 몇 가지 내부 기능이 있습니다.

함수	설명
HTGetLastError	명시된 펜의 최근의 읽기 동안 오류가 있었는지 결정합니다.
HTGetPenName	명시된 트렌드의 명시된 펜 #에 현재 사용하는 태그명을 돌려줍니다.
HTGetTimeAtScooter	ScootNum과 ScootLoc에 의해 명시된 스쿠터 위치에서 표본으로 1970년 1월 1일 GMT 시간으로 00:00:00 이후의 시간을 초로 돌려줍니다. UpdateCount, ScootNum, ScootLoc은 이중 어느 매개변수가 변할 때 식을 계산하게 합니다. 이것은 식이 새 읽기가 있은 후 또는 스쿠터가 이동한 후에 계산됨을 확실하게 합니다.
HTGetStringAtScooter	ScootNum과 ScootLoc에 의해 명시된 스쿠터 위치에서 표본 시간/날짜를 포함하는 문자열을 돌려줍니다. UpdateCount, ScootNum, ScootLoc은 이중 어느 매개변수가 변할 때 식을 계산합니다. 이것은 식이 새 읽기가 있은 후 또는 스쿠터가 이동한 후에 계산됨을 확실하게 합니다. 문자열의 서식이 돌려주는 값의 내용을 결정합니다.
HTGetValue	전체 트렌드의 명시된 펜의 요청된 유형의 값을 돌려줍니다.

HTGetValueAtScooter	명시된 스쿠터 위치, 트렌드, 펜 #의 표본으로 요청된 유형의 값을 돌려줍니다. UpdateCount 매개변수는 읽기가 끝난 후 식을 계산합니다.
HTGetValueAtZone	명시된 펜 트렌드의 오른쪽과 왼쪽 스쿠터 위치 사이에 포함된 데이터의 요청된 시간 값을 돌려줍니다.
HTScrollLeft	트렌드의 시작 시간을 현재 시작 시간보다 트렌드 넓이의 퍼센트 만큼의 이전 값으로 설정합니다. 결과는 차트의 날짜/시간을 주어진 퍼센트 만큼 왼쪽으로 이동하게 합니다.
HTScrollRight	트렌드의 시작 시간을 현재 시작 시간보다 트렌드 넓이의 퍼센트 만큼의 이후 값으로 설정합니다. 결과는 차트의 날짜/시간을 주어진 퍼센트 만큼 오른쪽으로 이동하게 합니다.
HTSetName	다른 태그명을 트렌드의 펜에 지정합니다.
HTUpdateToCurrentTime	종료 시간이 현재 시간과 같은 데이터를 읽고 화면에 나타냅니다. 시작 시간은 차트의 종료 시간 - 넓이와 같습니다.
HTZoomIn	새 차트 넓이와 시작 시간을 계산합니다. 트렌드의 .ScooterPosLeft는 0.0이고 .ScooterPosRight은 1.0 이면, 새 차트 넓이는 이전 차트 넓이를 둘로 나눈 것과 같습니다. 새 시작 시간은 LockString의 값에 따라 계산합니다.
HTZoomOut	새 차트 넓이와 시작 시간을 계산합니다. 새 시작 시간은 LockString의 값에 따라 계산합니다.

HistData 유ти리티 프로그램

HistData 유ти리티 프로그램은 **InTouch**에 의해 작성된 이력 데이터 파일(*.LGH)을 사용합니다. 선택한 이력 데이터를 일반 텍스트 에디터 혹은 엑셀에서 편집할 수 있는 파일(*.CSV)로 변환합니다. 이력 데이터에 액세스는 요청하는 프로그램에서 또는 **InTouch** 내에서 매크로 기능을 통해 얻을 수 있습니다. **HistData** 프로그램은 반드시 그것을 사용할 프로그램을 시작하기 앞에 시작해야 합니다.

Note: **HistData** 프로그램은 원격 태그 참조와 같이 이용할 수 없습니다.

HistData Database

HistData 프로그램이 이력데이터를 처리하기 위해서 필요한 매개변수 목록은 아래와 같습니다. 아래의 매개변수를 모두 조건에 맞게 넣지 않으면 이력데이터 변환이 정상적으로 이루어지지 않습니다.

항목	유형	설명
DATADIR	메시지	이력 데이터 파일을 포함하는 경로 이름. (예: C:\InTouch\App.)
DBDIR	메시지	InTouch 태그명 데이터베이스를 포함하는 디렉토리의 경로 이름.
STARTDATE	메시지	MM/DD/YY 포맷으로된 표본 시작 날짜
STARTTIME	메시지	24-시간 시계를 사용하여 HH:MM:SS 포맷으로된 표본 시작 시간
DURATION	메시지	데이터가 돌아오는 시간의 길이 DURATION 은 주, 일수, 시간, 분, 초로 표현할 수 있습니다. 다음은 유효한 문자입니다. w (주), d (일), h (시간), m (분), s (초). 값이 또한 허락됩니다, (예 : 0.5s 는 500 밀리초. 단일 표본을 요청하려면, DURATION 을 0 (영) 으로 설정합니다.
INTERVAL	메시지	표본 사이에서 시간의 길이 INTERVAL 은 주, 일수, 시간, 분, 초로 표현합니다. 예, 1w 는 1 주를 말함. 분수 값 또한 허용됩니다. (예, .25d 는 6 시간을 말함. 유효 문자는 DURATION 의 것과 동일)

Note: **DURATION** 과 **INTERVAL** 에 허용된 시간의 최대 길이는 6 주입니다. 이것은 모든 요청 유형들, 일수, 초, 등에도 적용합니다. 예를 들면, 일수를 사용하면, 42 가 최대 (7 일 x 6 주 = 42).

TAGS 메시지 이력 데이터를 돌려줄 태그명의 항목 **태그가 "TagA, TagB, TagZ"**의 형식으로 입력됩니다. 또한, 표본 날짜 시간이 내부 시스템 태그명 **\$Date** 와 **\$Time** 을 이용함으로 요청될 수 있습니다. 예를 들면: **"\$Date,TagA,TagB"** 또는,
"\$Time,TagA,TagB" 또는,
"\$Date,\$Time,TagA,TagB"

TAGS1, TAGS2.... 메시지 "태그" 문자열은 **WindowViewer**에서 131 문자가 가능하고 엑셀에서는 255입니다. **"Tags1," "Tags2,"** 등과 같은 태그 항목 이름을 추가 함으로 더 긴 요청을 문자열에 덧붙입니다. 태그명에 태그 텍스트가 덧붙여져야 하면, 더하기 (+)를 문자열의 끝에 입력합니다. 예를 들면:

```
TAGS="$Date,ProdLevel,ProdTemp,+ "
TAGS1=" ReactLevel,Temp,GasLevel,+"
TAGS2=" MotorStatus"
```

Note: 중복된 태그명은 허용되지 않고 각 태그 문자열의 최대 길이는 512 바이트입니다.

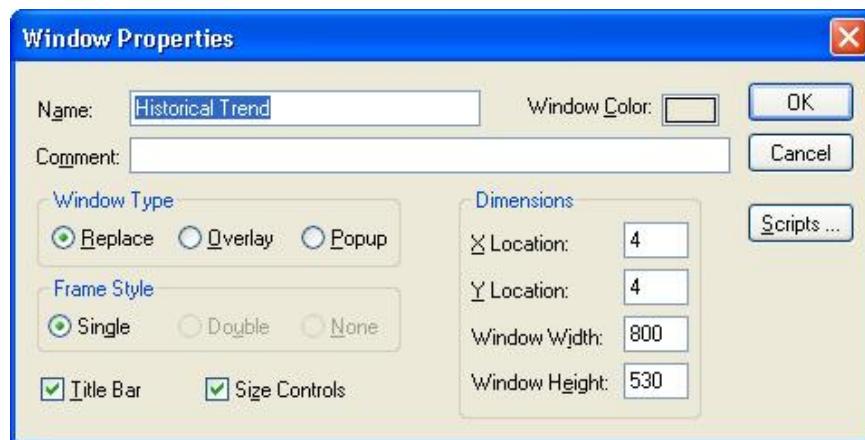
PRINTTAGNAMES	이산	이 항목은 기본값이 1 이고 HistData 가 연관된 열의 값 위 출력 파일의 첫째줄에 태그명을 인쇄하게 만듭니다. 태그명이 인쇄되어야 하면, 이 항목의 값은 반드시 0 (영) 으로 변해야 합니다.
DATA	메시지	이 항목은 HistData 프로그램에서 요청 된 데이터를 쉼표 구분 변수 포맷으로 갖는데 사용합니다. 이것은 조언 또는 DDE 를 통한 요청 데이터를 원하는 다른 응용프로그램에서 사용합니다.
SENDDATA	정수	1 로 설정되면, HistData 는 요청된 데이터로 DATA 항목을 업데이트 합니다. 업데이트가 끝나면, SENDDATA 는 자동적으로 0 (영) 으로 재설정됩니다.
FILENAME	메시지	요청된 데이터를 쓰기 위한 파일의 전체경로 이름.
WRITEFILE	정수	설정이 1 일 때, HistData 는 요청된 데이터를 FILENAME 항목 이름에 의해 명시된 파일에 씁니다. 파일의 업데이트가 끝나면, WRITEFILE 이 자동적으로 0 (영) 으로 재설정됩니다.
STATUS	이산	최근의 작동의 상황을 디스플레이 합니다. 1 은 성공을, 0 (영) 은 오류가 발생했음을 나타냅니다.
ERROR	메시지	최근 오류의 설명을 포함하는 문자열입니다. STATUS 가 1 이면 "없음" 이고, STATUS 가 0 (영) 일 때의 오류 메시지.

Lab 9 – 이력 트랜드 설정

Lab Instructions

이력 트랜드 윈도우 생성

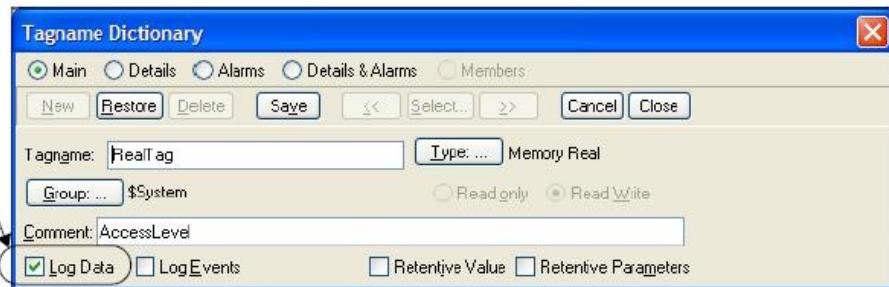
1. 아래 보여 지는 그림과 같이 새로운 윈도우를 정의하고 속성을 설정합니다..



2. OK를 클릭합니다.
3. Tagname Dictionary를 열기 위해 Ctrl+T 키를 누릅니다.
4. IntTag에 Log Data 옵션을 설정 합니다.



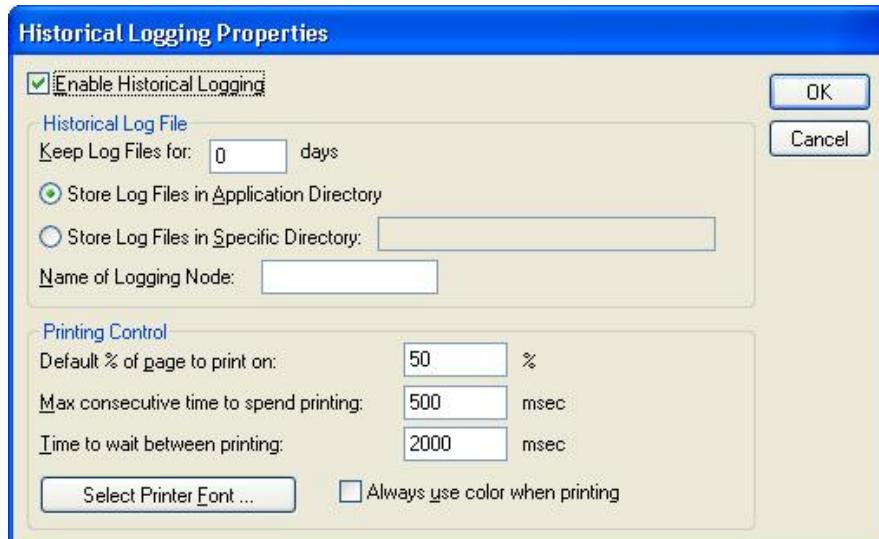
5. SAVE를 클릭합니다.
6. >> 버튼을 사용하여 RealTag를 선택합니다..



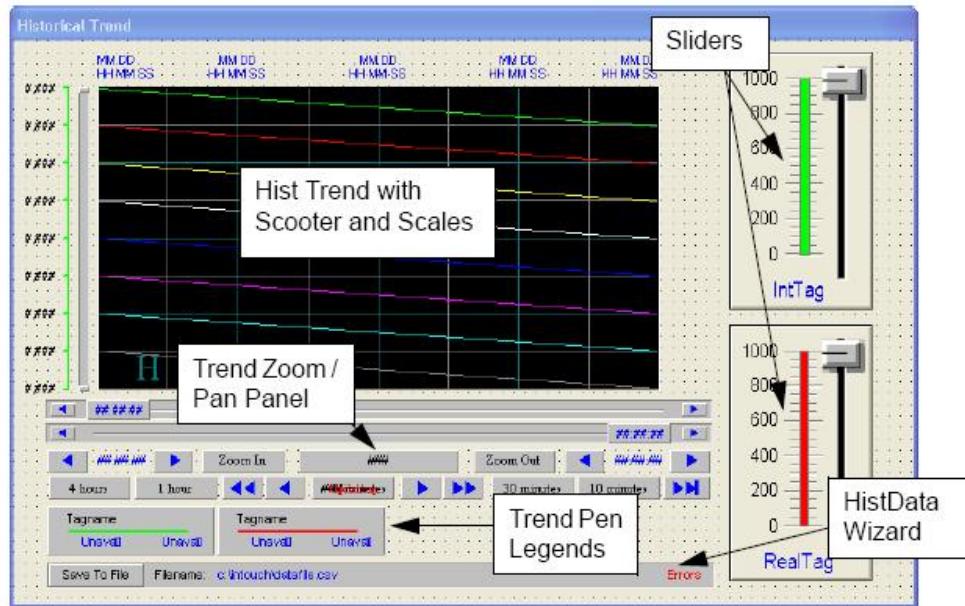
7. IntTag에 Log Data 옵션을 설정합니다.
8. SAVE를 클릭하고 닫습니다.

Note: 메시지 태그 타입은 데이터를 로깅할 수 없습니다..

9. Special / Configure / Historical Logging을 클릭합니다.
Historical Logging Properties 대화상자가 나타납니다.



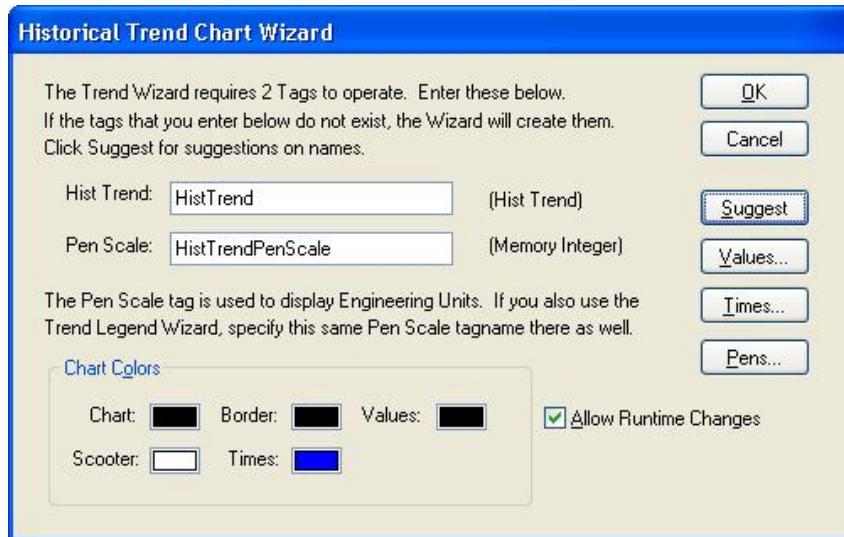
10. Enable Historical Logging 체크박스를 체크합니다.
11. OK를 클릭합니다.
12. WindowViewer가 실행중이라면 종료합니다. WindowViewer를 재시작하지 않습니다.
13. 아래의 그림처럼 Trend Wizard에서.. HistTrend W/Scooter and sales, Trend Zoom/ Pan Panel, Trend Pen Legend, HistDataWizards를 윈도우에 그려 넣습니다. 또한 두개의 슬라이더를 삽입합니다. Real-Time Trend에서 만들었던 슬라이더를 복사해서 붙여넣습니다..



Trend Wizard 설정

14. 트랜드 위에서 더블 클릭합니다.

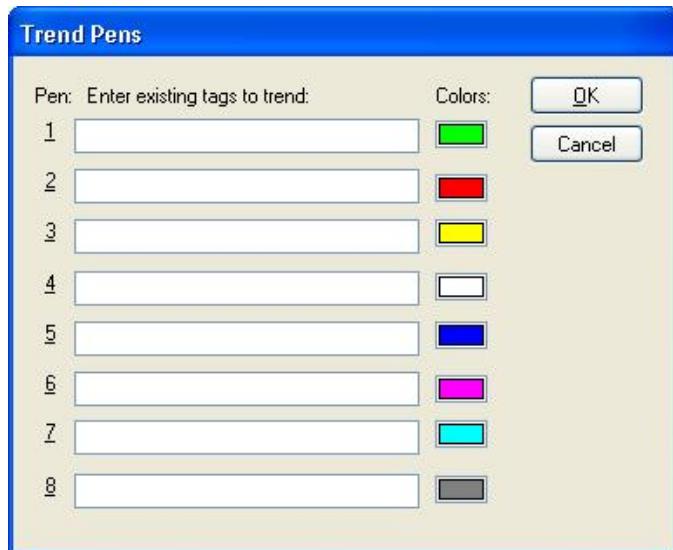
Historical Trend Chart Wizard 대화상자가 나타납니다.



15. 태그명들을 입력하기위해 Suggest를 클릭합니다.

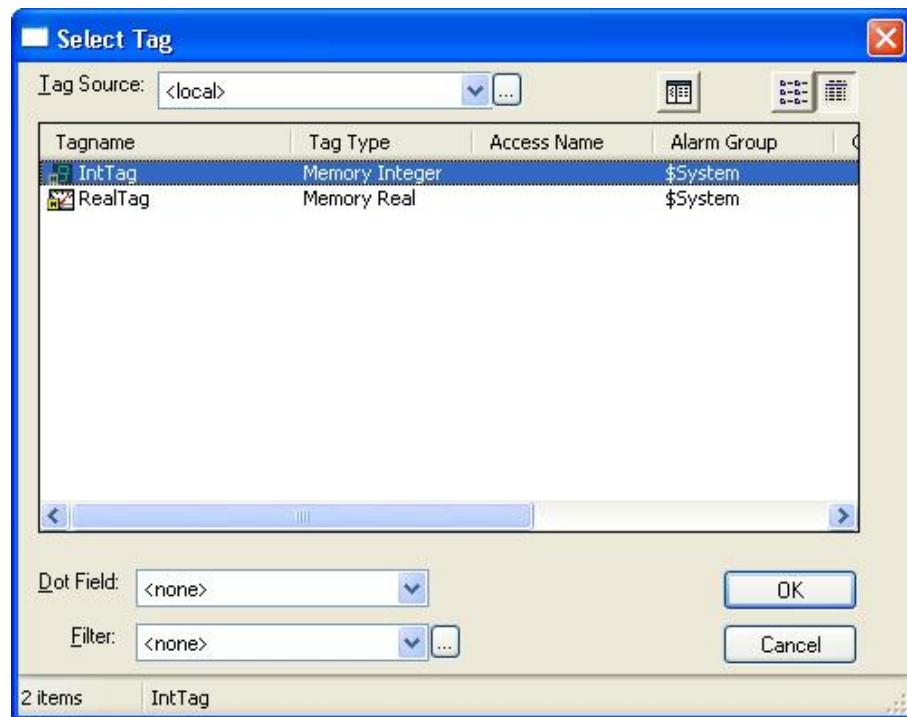
제시된 이름들이 자동적으로 나타납니다. 지금 윈도우와 관련된 HistTrend 태그에서 Trends 마법사의 모든것을 유지합니다.

16. Pens 버튼을 클릭합니다. Trend Pens 대화상자가 나타납니다.



17. Pen 1 필드를 더블클릭합니다.

SelectTag 창이 나타납니다.



18. IntTag를 선택하고 OK를 클릭합니다.
19. Pen 2 필드에 이전 두 단계를 반복하고 RealTag를 선택합니다.
20. OK를 클릭합니다.
21. Historical Trend Chart Wizard 대화상자를 종료하기 위해 OK를 다시 클릭합니다.

Trend Zoom/Pan Panel 정의

22. Trend Zoom/Pan Panel을 더블클릭합니다. Trend Panel Wizard 대화상자가 나타납니다.

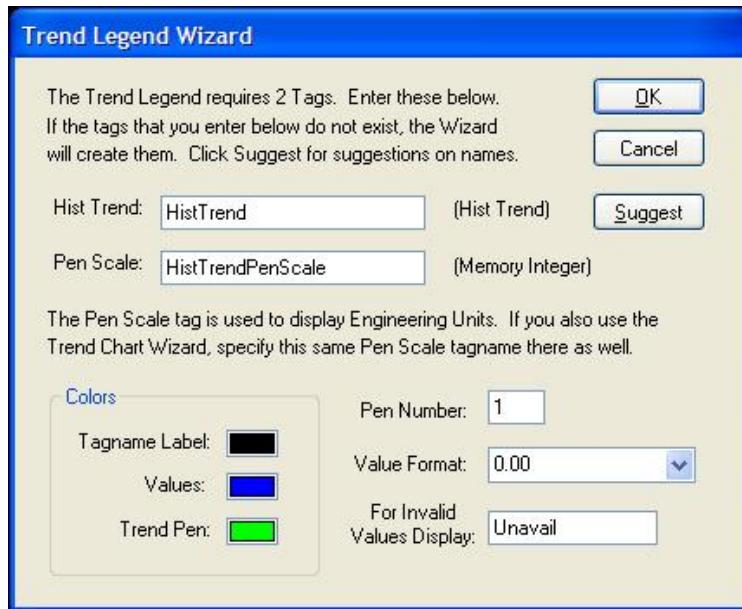


23. 앞에서 사용된 HistTrend 태그명인지 확인합니다..
24. OK를 클릭합니다.

Trend Pen Legend 정의

25. 첫번째Trend Pen Legend 객체를 더블 클릭합니다. 이 객체는 IntTag 태그를 보여지게 할 것입니다. Trend Legend Wizard가 아래와 같이 나타납니다.

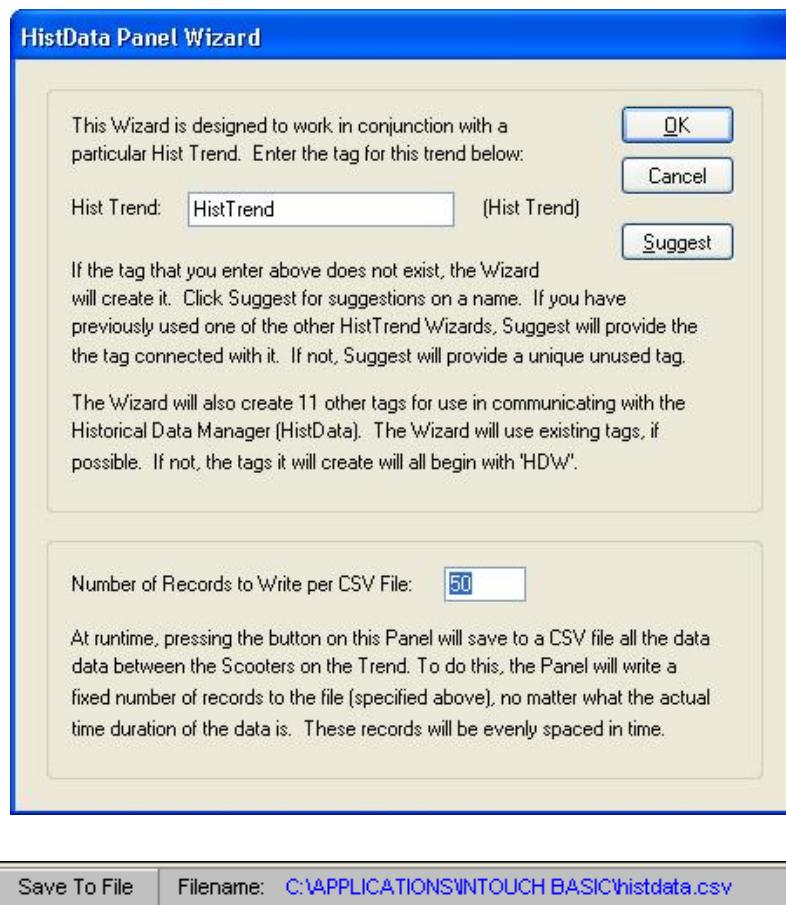




26. 정확한 태그명을 입력하기 위해 **Suggest**를 클릭하고 사용된 **HistTrend** 태그명을 확인합니다.
27. Color 그룹에서 **Trend Pen**에 녹색을 사용합니다.
28. **Pen Number** 상자에 1이 입력된 것을 확인합니다.
29. 다른 항목은 기본 설정으로 남겨둡니다..
30. **OK**를 클릭합니다.
31. 두번째 Trend Pen Legend(RealTag 태그에)를 더블 클릭합니다.
32. **Suggest**를 클릭합니다.
33. Trend Pen에 빨간색을 입력하고 **Pen Number**에 2를 입력합니다.
34. **OK**를 클릭합니다.

HistData Wizard 정의

35. HistData Panel Wizard를 보여주기 위해 HistData 객체를 더블클릭합니다.
36. HistTrend 태그를 입력하기 위해 **Suggest**를 클릭합니다.
37. Number of Records to Write per CSV File에 50를 입력합니다.
38. **OK**를 클릭합니다.



슬라이드 정의

Real Time Trend 윈도우로 부터 두개의 슬라이드를 복사하지 않는다면, 그들 사용을 다음단계로 설정합니다.

39. Upper 슬라이드를 더블 클릭합니다. IntTag 태그에 이 슬라이드가 사용될 것입니다.



40. Slider Wizard 대화상자를 아래와 같은 속성으로 설정합니다.

Tagname / IntTag

Slide Face / Label IntTag

Fill color / Green

Slide Range / Maximum / 1000

Decimal Places / 0

다른 기본 설정들은 남겨둡니다.

41. OK를 클릭합니다.

42. 두번째 슬라이드 사용을 이 과정으로 반복합니다.

Tagname / RealTag

Slide Face / Label RealTag

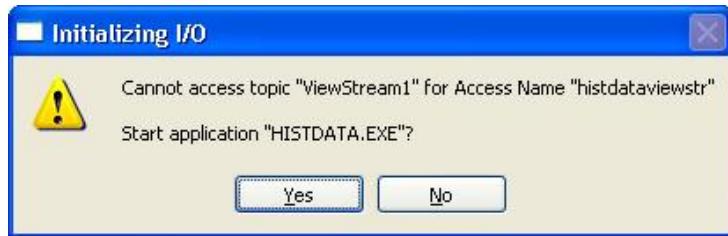
Fill color / Red

Slide Range / Maximum / 1000

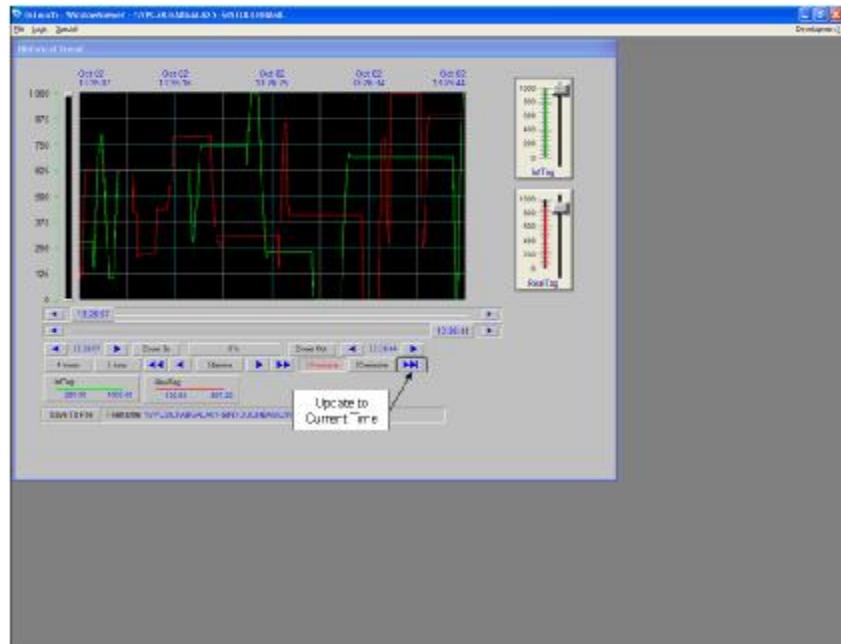
Decimal Places / 0

43. Runtime으로 전환합니다.

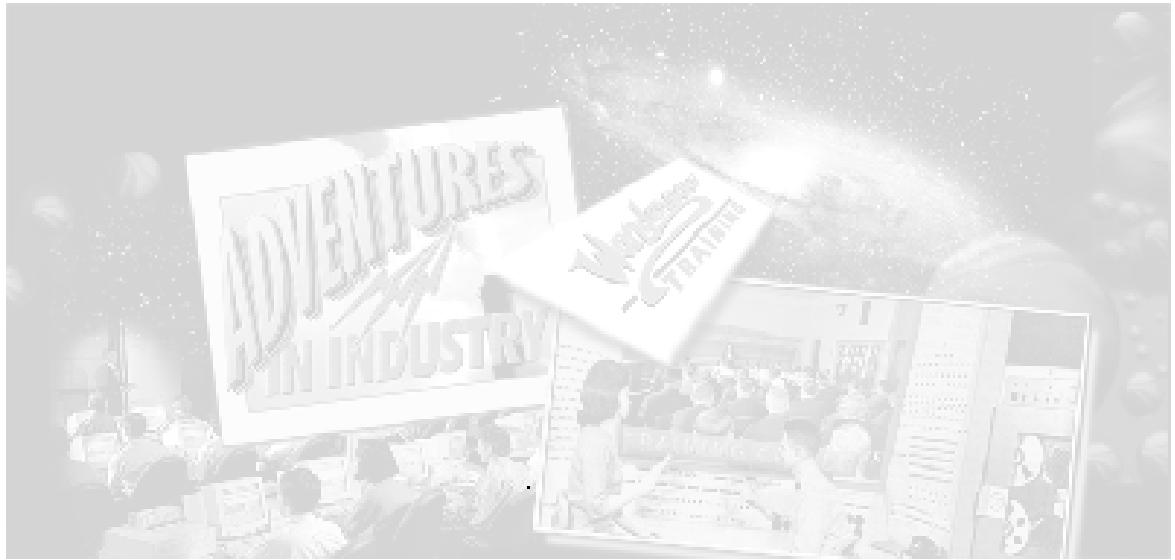
HistData 프로그램을 시작하기 위해 다음의 메세지 박스가 나타날 것입니다.



44. YES를 클릭합니다.



45. 슬라이더를 아래위로 움직여 값을 변경하면 Historical Log file에 데이터가 저장됩니다.
46. 가장 최근의 데이터로 Trend를 update 하기 위해 Update to Current Time를 클릭합니다.
47. Scooters를 움직이고 zoom in 과 out을 하기 위해 Zoom 버튼을 클릭합니다.
48. Historical Data Wizard에서 file name을 클릭합니다. 이력데이터를 변환하여 만들 .csv 파일의 파일명(path 포함)을 입력 또는 확인합니다.
49. Save to file 버튼을 클릭합니다. HistData Wizard는 “filename”에 의해 지정된 csv에서 추출된 정보들을 spreadsheet 형식으로 보여 줄 것입니다.



Module 8

I/O 통신

Section 1 – I/O Communication Protocols

Section 2 – Wonderware I/O Servers

Lab 10 – Configuring I/O Communication

Section 3 – Troubleshooting I/O

Section 1 – I/O Communication Protocols

이 Section은 I/O 통신 프로토콜과 InTouch에서 access 명을 어떻게 설정하는지를 설명하고 있습니다.

Dynamic Data Exchange(DDE)

DDE는 원도우즈 환경에서 Application 간에 서로 데이터를 주고받을 수 있도록 마이크로소프트에 의해 개발된 통신 프로토콜입니다. 이는 동시에 실행되는 두 개의 프로그램간에 ‘클라이언트 – 서버’ 관계를 형성합니다.

Server Application은 데이터를 원하는 다른 Application으로부터 데이터 요청을 받고, 데이터를 제공합니다. 요청하는 Application을 클라이언트라 부릅니다. 몇몇 Application은 (InTouch나 마이크로소프트 엑셀 등) 클라이언트인 동시에 서버가 될 수 있습니다.

Note : 예전에는 Wonderware 소스와 비 Wonderware 소스간에 통신은 네트워크상에서만 통신 가능한 프로토콜이었던 NetDDE가 Windows XP와 Windows 2000에서 지원 되었지만 windows 2003 server에서는 Wonderware 소스와 비 Wonderware 소스간에 통신에 사용되었던 NetDDE(네트워크상에서만 통신 가능한 프로토콜) 프로토콜을 지원하지 않습니다.

Wonderware SuitLink

Wonderware SuitLink는 TCP/IP가 기본인 프로토콜입니다. SuitLink는 산업용으로 쓰일 수 있도록 필수적인 요소들을(예를 들어, 데이터 보존성, 높은 처리량, 좀 더 쉬운 오류 진단 등) 구체적으로 만족시키기 위해 설계되었습니다. 이 프로토콜은 Windows 2000, 2003 Server, XP에서 사용할 수 있습니다.

SuitLink는 DDE를 대체하기 위한 프로토콜은 아닙니다. Wonderware에서는, DDE는 내부적으로 클라이언트끼리 통신을 할 경우, SuitLink는 네트워크상에서 통신을 이용할 경우에 사용할 것을 추천합니다.

각각의 Server-Client 통신은 현장 상황에 따라 달라집니다.

SuitLink는 다음과 같은 장점을 가지고 있습니다:

- | 높은 데이터 전송의 일관성을 가진 단일 노드나, 많은 노드를 가진 분산된 환경에서도 Application 사이의 전송량이 유지 됩니다.
- | Value Time Quality(VTQ)는 VTQ를 사용할 수 있는 클라이언트로 들어오는 모든 데이터 값에 대해서, Time stamp 및 통신 품질의 등급을 사용할 수 있습니다.
- | 데이터 처리량, 서버 로딩, 컴퓨터 자원 소비량, 네트워크 전송량에 대한 확장된 오류 진단은 Microsoft Windows NT 버전 운영체제의 성능 모니터링을 통해서 접근 할

수가 있으며, 이 기능은 시스템 스키마(scheme)와 분산된 산업용 네트워크를 유지, 보수하는데 있어서 중요한 역할을 합니다.

- | 네트워크 전송 프로토콜은 Microsoft 사의 표준 WinSock : 소켓 소스)인터페이스를 사용하는 TCP/IP 를 사용합니다.
SuitLink 통신 프로토콜을 사용하기 위해서는 다음과 같은 조건을 만족해야 합니다:
 - | 적절하게 설치되어 사용되고 있는 Microsoft 의 TCP/IP 드라이버를 가지고 있어야 합니다.
 - | 15 개의 문자를 넘기지 않는 컴퓨터 이름(노드 명)을 가지고 있어야 합니다.
Microsoft 사의 TCP/IP 를 설정하고, 설치하기 위해서 더 많은 정보가 필요하다면, Microsoft Windows 운영체제의 가이드를 참고하기 바랍니다.
 - | Wonderware SuitLink 가 운영체제 서비스로 실행되고 있어야 합니다. 만일, 어떤 예기치 않은 상황으로 SuitLink 서비스가 멈춘다면, 다시 재시작을 시켜 줘야 합니다. (SuitLink 는 InTouch 를 설치 할 때, Common Component 로써 자동적으로 설치가 되며, Windows 서비스로써 자동 실행되도록 설정 됩니다.)

InTouch I/O 주소 지정

InTouch 는 ‘application name’, ‘topic name’, ‘item name’ 이렇게 세가지 부분에 대해서 각각의 이름을 만들게 됨으로써, I/O 서버 프로그램의 데이터 구성 요소를 동일화 할 수 있습니다.

다른 Application 으로부터 데이터를 받기 위해서, Client 프로그램(InTouch)은 이러한 세가지 항목들을 명시함으로써 Server 프로그램으로의 채널을 열게 됩니다.

또한, InTouch 가 다른 Application 으로부터 데이터 값을 얻기 위해서는, 데이터 값을 제공하는 application name, 데이터 값을 가지고 있는 Application 의 topic name, topic 에 정의된 item 의 name 을 알아야 합니다. 뿐만 아니라 데이터의 타입도 알아야 합니다: Discrete, Integer, Real(floating point), Message(string)

이러한 정보는 InTouch 의 데이터베이스에 정의 될 때, 태그의 I/O 타입을 결정하게 됩니다. WindowViewer 가 실행 중일 경우, 이러한 항목의 데이터 값을 유지하고, 값을 가져오기 위해 요구되는 모든 사항을 자동적으로 수행하게 됩니다.

예를 들어, Excel 의 경우, application name 은 “Excel”, topic name 은 데이터가 입력되어 있는 spreadsheet 의 이름이며, item name 은 데이터가 쓰이거나 읽혀지는 spreadsheet 의 개별적인 cell 들의 위치입니다.

다른 윈도우 Application 이 InTouch 로부터 데이터 값을 요청할 경우, 마찬가지로 위에서 언급한 세가지 I/O 주소 항목들을 알아야만 합니다.

다음은 I/O 주소 지정을 InTouch 에서 사용하는 방법입니다.

- I **VIEW (application name)** : 데이터 요소를 가지고 있는 실행 중인 프로그램
- I **TAGNAME (topic name)** : InTouch 의 데이터베이스에 태그명을 읽거나 / 쓸 경우 항상 사용되는 단어입니다.
- I **ActualTagname (item name)** : InTouch 의 tag dictionary 에 item 으로 정의 되는 실제 태그명입니다.

예를 들면, 엑셀에서 InTouch 로 데이터 값을 로드시키기 위해서는(같은 노드에서 실행 중인 경우), DDE Remote Reference 공식을 데이터 값이 쓰여지는 cell 에 다음과 같은 방법으로 입력합니다.

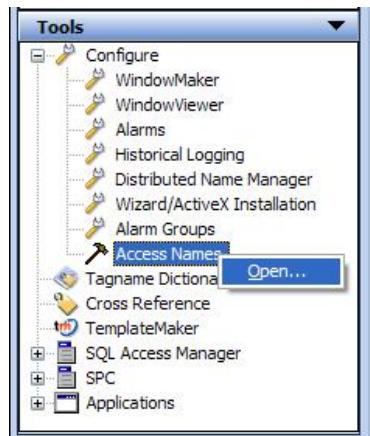
=VIEW/TAGNAME!'ActualTagname'

InTouch Access Names

I/O 타입의 태그나 remote 태그를 생성할 경우, 반드시 Access Name 과 연결되어 있어야 합니다. Access Name 은 node name, application name, topic name 을 갖는 다른 I/O 데이터 소스와 통신하기 위해 사용되는 정보들을 가지고 있습니다.

Access Name 생성

- a. Application Explorer 의 Tools 영역으로부터 Configure 를 클릭합니다.



- b. Access Names 를 오른쪽 마우스로 클릭하고, Open 을 선택합니다.

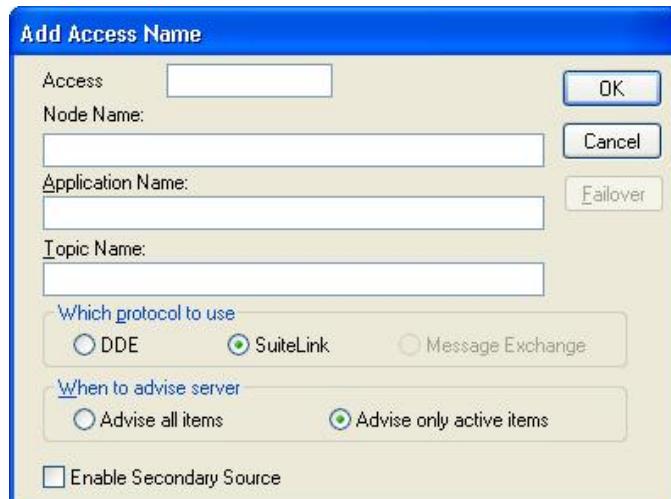
Note : Access Names 는 tagname dictionary 에서 I/O 타입의 태그를 정의하는 동안 생성될 수도 있습니다.

Access Names 창이 보일 것입니다:



- Add 버튼을 클릭합니다.

Add Access Name 창이 열린 것이 보일 것입니다:



- Access Name : 새로운 Access Name 을 입력합니다.

(간단하게, topic name 로 사용할 이름과 같은 이름으로 사용)

InTouch 는 실시간 I/O 데이터를 조회하기 위해서 Access Names 를 사용합니다. 각각의 Access Name 은 Node, Application, Topic 을 포함하는 I/O 주소와 동일시 됩니다. 분산된 Application 환경에서의 I/O 조회는 네트워크상의 I/O Server 에는 전역 주소를, 로컬 환경에서의 I/O Server 에는 로컬주소를 설정합니다.

- e. **Node Name** : 데이터가 네트워크상에서 원격 I/O Server에 있을 경우 사용됩니다.
- f. **Application Name** : 데이터 값을 가져오는 I/O Server 프로그램의 실제 프로그램 이름을 입력합니다. 만일, 데이터 값이 Wonderware Modbus I/O Server로부터 오고 있다면, **MODBUS**를 사용합니다. 프로그램이름의 확장자인 .exe 부분은 입력하지 않아야 합니다.

- g. **Topic Name** : 접속하기 위한 topic name을 입력합니다. **Topic Name**은 데이터 요소의 특정한 Application의 하위 그룹입니다.

Wonderware의 I/O Server 프로그램으로부터 데이터가 들어오는 경우, topic name은 I/O Server에 정의되어 있는 똑 같은 topic name을 사용해야 합니다.

- h. **Which Protocol to use** : DDE(내부적으로) 또는 SuiteLink(네트워크상에서)

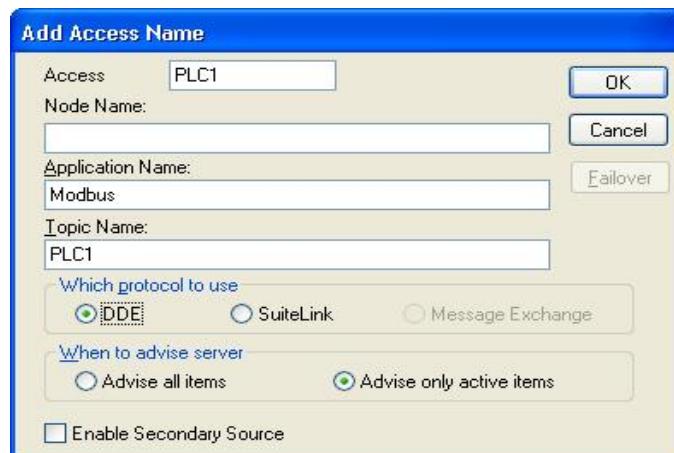
- i. **When to advise server** :

Advise all items : 만약 서버 프로그램이 윈도우나 알람, 로그, 트랜드, 스크립트의 사용 여부에 관계없이 모든 데이터를 보내는 경우 선택합니다. 이 옵션을 선택하는 것은 성능에 영향을 미치기 때문에 권장하지 않습니다.

Advise only active items : 만약 서버 프로그램이 알람, 로그, 트랜드, 스크립트가 사용되는 포인트나 윈도우가 있는 경우만 데이터를 보내는 경우 선택합니다. Touch Pushbuttons – action 스크립트는 해당 윈도우가 열려 있지 않다면, 데이터를 보내지 않습니다.

다음 그림은 access name을 완성하기 위한 예입니다. (만일, I/O Server가 로컬 노드에 위치해 있다면, Node Name은 다음 그림과 같이 공란으로 둡니다.)

새로운 Access Name을 만들기 위해 OK 버튼을 클릭합니다.



- j. Close 버튼을 클릭해서 창을 닫습니다.



Access Name 설정 또는 삭제

- a. Application Explorer 의 Tools 영역으로부터 Configure 를 클릭합니다.
- b. Access Names 를 더블-클릭합니다.

Access Names 창이 보일 것입니다:



- c. 수정하려는 access name 를 선택합니다.
- d. Modify 버튼을 클릭한다. Modify Access Name 창이 보일 것입니다.
- e. 수정을 끝마친 후, OK 버튼을 클릭합니다
- f. 다른 Access Names 를 수정하고 싶다면, 이러한 과정을 반복합니다.

Access Name 을 삭제하기 위해서는, Delete 버튼을 클릭합니다.

삭제를 확인하기 위한 메시지 창이 뜨면, Yes 버튼을 클릭합니다.

- g. Close 버튼을 클릭하거나 다른 Access Names 를 삭제하고 싶다면, 이러한 과정을 반복합니다.

Note : 태그에 의해 사용된 Access Names은 삭제할 수 없습니다.

I/O 탑입 태그명 정의

모든 I/O 탑입의 태그들은 엑셀이나 서버 같은 다른 원도우 Application 프로그램들로부터 값을 받게 됩니다. 이러한 값은 'raw' 값으로 참조됩니다. Tagname dictionary에서 태그를 정의 할 경우, Min – Max Raw 값으로 입력할 수 있으며, 데이터베이스에 저장이 되어 I/O Device로부터 들어오는 실제 데이터값을 고정시키는데 사용됩니다. 예를 들어, 만약 I/O Server로부터 들어오는 값이 0이고, Min Raw 값이 50으로 설정되었다면, 데이터베이스는 Raw 값을 50으로 인식하게 됩니다.

InTouch는 raw 값을 보여주지 않습니다. 대신에, engineering unit(EU)을 보여주게 됩니다. Tagname dictionary에서 I/O 탑입의 태그를 정의할 때, 구체적인 Min –Max EU 값을 설정해야 합니다. 이러한 값들은 raw 값과 보여지는 값을 비교하는데 사용됩니다. 만약 비교를 원하지 않는다면, Min/Max EU 값과 Min/Max Raw 값을 같게 맞춰주도록 합니다.

예를 들어, 송전기에 100%의 전류가 흐른다면 값은 9999로, 0%의 전류가 흐른다면 값은 0이라고 가정할 경우, 다음과 같이 입력합니다:

Min EU = 0 Max EU = 100

Min Raw = 0 Max Raw = 9999

raw 값이 5000이라면 50으로 보여지게 됩니다.

송전기에 전류가 흐르지 않는다면, 6400의 값이, 300 GPM 일 경우, 32000 이 값으로 설정된다면,

Min EU = 0 Max EU = 300

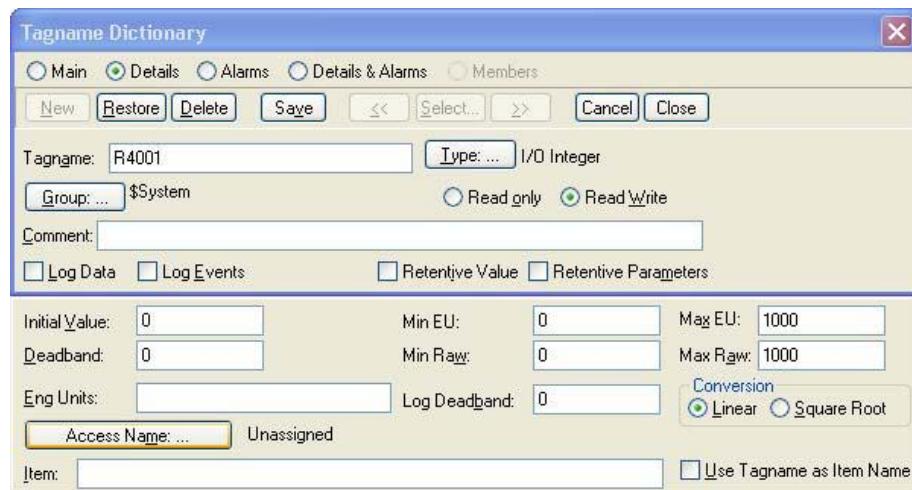
Min Raw = 6400 Max Raw = 32000

이와 같은 경우에는, raw 값이 12800이라면, 150으로 출력이 되며, raw 값이 6400일 경우에는 0으로 출력됩니다. raw 값이 0일 경우 역시 0으로 출력이 됩니다. (Min Raw 와 Max Raw의 값에 의해 설정된 경계값을 초과하거나 미만인 raw 값을 제한됩니다)

InTouch tagname dictionary에서 다른 원도우 ApplicationWrite 작업이 이루어지면, 위와는 반대의 작업이 이루어지게 될 것입니다.

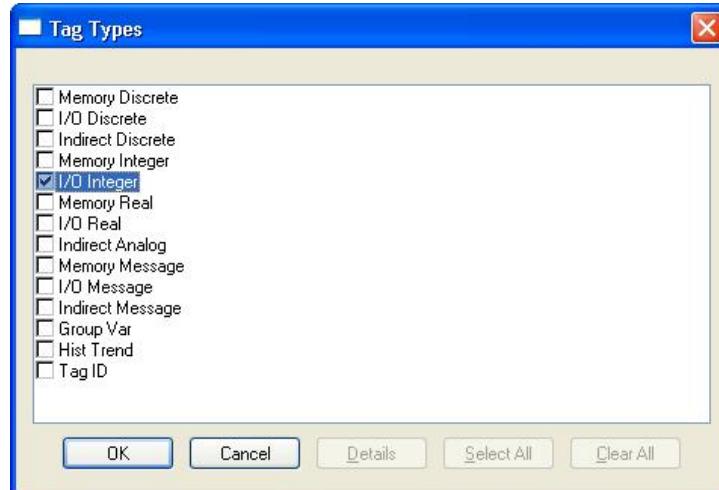
a. Ctrl + T 키를 누릅니다.

Tagname dictionary 창이 열릴 것입니다.



- b. Details 버튼을 클릭합니다.
- c. 새로운 태그를 정의하기 위해서, New 버튼을 클릭합니다.
- d. Tagname 란에 새로운 태그명을 입력합니다.
- e. Type 버튼을 클릭합니다.

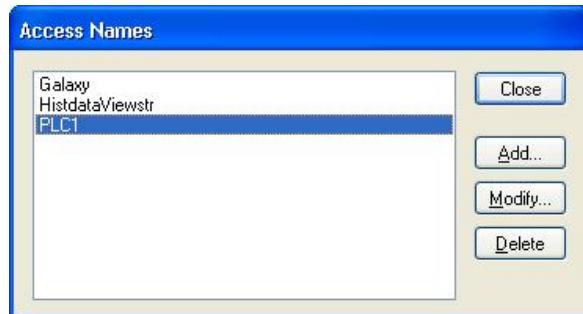
Tag Type 창이 열릴 것입니다:



- f. I/O Integer 를 체크합니다.
- g. OK 버튼을 클릭합니다. 각각의 자세한 항목들이 tagname dictionary 창에서 보일 것입니다.
- h. item 항목을 정의하기 위해서 필요한 모든 항목들을 명시해야만 합니다.

- i. Access Name 버튼을 클릭합니다.

Access Name 창이 열릴 것입니다:

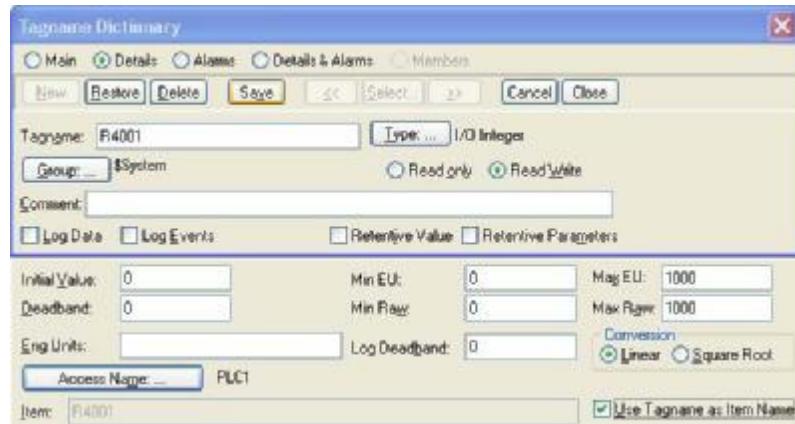


- j. 사용하려는 Access Name를 선택하고 Close를 클릭합니다.

위의 그림에서, PLC1 access name은 현재 입력된 태그에서 사용됩니다. 이 선택된 Access Name은 다음에 보여지는 tagname dictionary 창의 Details 영역에서 Access Name 버튼의 바로 옆 부분에서 보여질 것입니다.

- k. Item 영역에서, item 이름이나 IO 서버 Application이나 연관된 Application의 데이터 값의 숫자를 입력합니다. 만일, Item 이름이 태그명과 같다면, Item의 공란에 자동적으로 태그명을 입력되게 하기 위해서 Use Tagname as Item Name의 체크박스에 체크해 줍니다.

Note : “Tagname”은 데이터 값(태그)을 참조하기 위해서 InTouch에서 사용되는 이름입니다. “Item”은 같은 데이터 값을 참조하기 위해서 원격 윈도우 Application에서 사용되는 이름입니다. 이러한 이름들은 같아서는 안됩니다. 기본적으로 이러한 이름들은 같이 사용하지는 않지만, 어떤 케이스에서는 같은 이름을 사용하는 것이 추천됩니다.



- I. Close 버튼을 클릭합니다.

Note : 애니메이션 링크나 스크립트에서 사용이 되기 전까지는, item 을 추가한 후에는 Type 의 변경이 가능합니다.

Section 2 – Wonderware I/O Servers

이번 장에서는 Wonderware I/O Server(Modbus)의 설정방법에 대해서 설명할 것입니다.

소개

Wonderware I/O Server는 실제 현장(PLC, RTU 등)의 데이터에 DDE를 인식할 수 있는 Windows Application의 접속을 가능하게 하는 Microsoft Windows Application 프로그램이라고 할 수 있습니다.

Wonderware server들은 주로 Wonderware의 InTouch 프로그램과의 통신을 하기 위해서 사용되지만, DDE client로써 작동하는 Microsoft Windows 프로그램과의 통신을 위해서 사용되기도 합니다.

이번 장에서는, Wonderware I/O Server의 사용법, 설정 그리고 통신 개통을 어떻게 할 것인가에 대해서 알아 볼 것입니다. Wonderware의 I/O server는 Windows Application으로, 각각의 Server들은 비슷한 로직과 성능을 가지고 있습니다. server의 필수적인 요구사항과 추가적인 하드웨어(network.. 등등)가 필요할지도 모른다는 점과 설정 창에서 별도의 추가적인 정보들이 요구될지도 모른다는 사실을 명심해야 합니다.

다음의 내용들은 Wonderware 기본적인 교육 환경에서 제공이 되는 PLC의 RS-232 포트를 사용한 point-to-point의 Modbus I/O server의 참고용 정보 문서내용입니다.

Configuring I/O Servers

일단 I/O Server가 설치되면, 어떤 설정이 이루어져야 합니다. Server 설정을 하게 되면, MODBUSDV.CFG로 명명된 설정 파일이 자동적으로 생성이 됩니다. 이 파일은 통신 포트와 정의된 topic의 모든 내용들에 대한 정보들이 저장되어 있습니다.

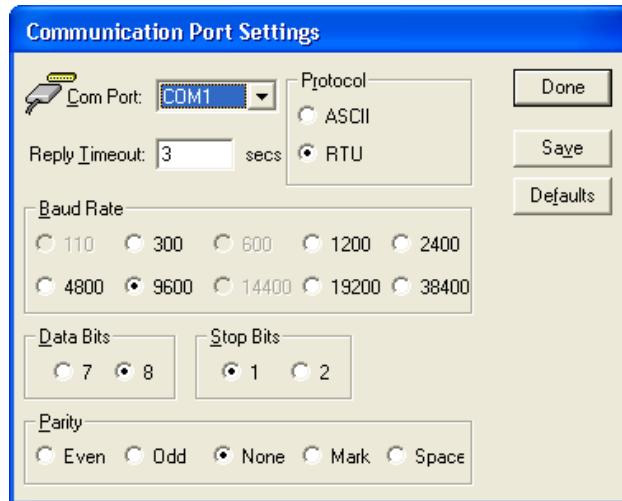
만일, Configure / I/O Server Settings 명령을 통해서 설정 파일이 저장될 디렉터리를 지정하지 않았다면, 이 설정 파일은 I/O server가 설치된 디렉터리에 자동 저장 됩니다.

- 시작 / 프로그램 / WonderwareFactorySuite / IOServers / Modicon MODBUS를 선택합니다.



- b. Configure / Com Port Settings 를 클릭합니다.

Communication Port Settings 창이 보일 것입니다. 이 창은 PLC 와 통신하기 위해서 사용되는 통신 포트의 설정을 위해서 사용됩니다.



Com Port : PLC 와 연결될 통신 포트를 선택합니다.

Reply Timeout : 위의 serial 통신 포트로 연결된 PLC 들이 I/O Server 로부터의 데이터에 재응답하는 시간을 입력합니다.

Note : 이 timeout 은 PLC 가 응답이 없을 경우에만 사용됩니다. PLC 가 보통상태로 응답하고 있다면, 사용되지 않습니다. 기본 설정 값은 3초이며, 대부분의 설정 값은 이 값으로도 충분합니다.

Protocol : 위에서 설정된 통신 포트에 연결되는 장비의 protocol 을 선택합니다.

Baud Rate : 위에서 설정된 통신 포트에 연결되는 장비에 맞는 데이터 처리율(serial bit)을 선택합니다.

Data Bits : 위에서 설정된 통신 포트에 연결되는 장비의 설정 값에 상응하는 data bits 의 숫자 옵션을 선택합니다.

만일, Protocol 에서 ASCII 를 선택 하게 된다면, 7.

만일, Protocol 에서 RTU 를 선택 하게 된다면, 8 을 사용합니다.

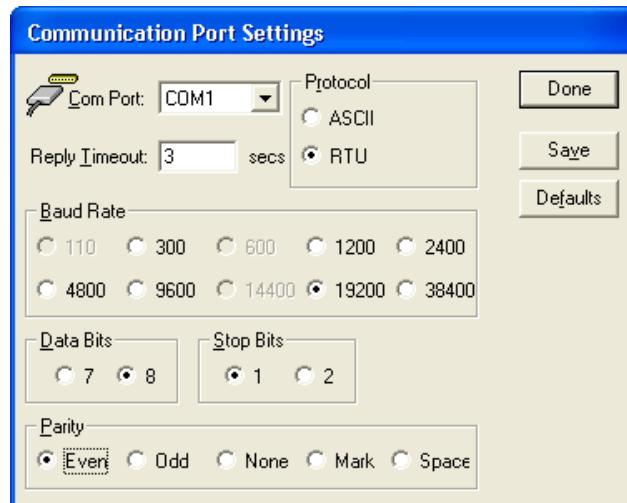
Stop Bits : 위에서 설정된 통신 포트를 위한 stop bits 의 대략적인 수를 선택합니다.

만일, baud rate(데이터 처리율)이 300 이상이라면, 이 stop bits 는 1 로 설정합니다.

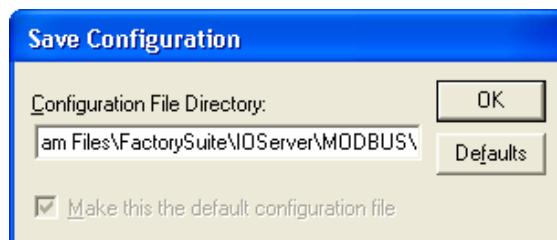
Parity : 위에서 설정된 통신 포트에 연결되는 장비의 설정 값에 상응하는 설정 값을 선택합니다.

Note : 모든 단일 통신 포트의 장비들은 같은 Protocol, Parity, Stop Bits, Data Bits, Baud Rate로 설정이 되어야 합니다.

- 변경된 내용을 **Save** 버튼을 클릭해서 저장한 뒤, **Done** 버튼을 클릭해서 종료합니다.



Note : 통신 포트를 위해 현재 입력된 설정 값을 저장하기 위해서, **Save** 버튼을 클릭한 후, **Communication Port Settings** 창에서 다른 통신 포트를 설정 할 수 있습니다.



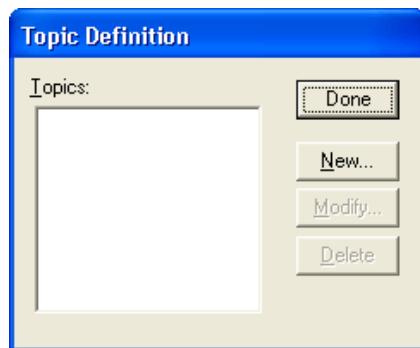
Creating Topic Definitions

Configure / Topic Definition 을 선택하면, topic 을 삭제하거나, 수정, 생성 할 수가 있습니다. 한 개 또는 그 이상의 topic 은 각각의 PLC(I/O Server 와 통신하려는 PLC)에 생성되어 있어야 합니다.

각각의 topic 들은 PLC 들에 별개로 연결이 되어 있으므로, 유일한 이름으로 설정 되어야 합니다. 이 유일한 이름은 모든 DDE 통신에서 DDE 주소로써 *topic name*으로 사용이 됩니다.

- a. Configure/ Topic Definition 을 클릭합니다.

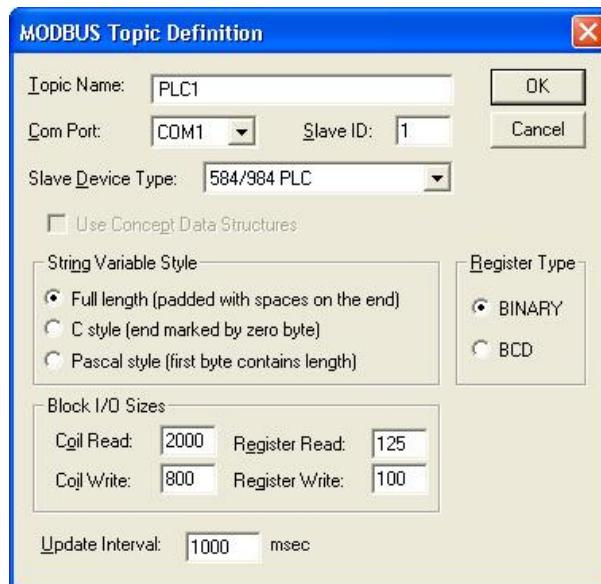
Topic Definition 창이 열릴 것입니다:



Note : 일단 topic 이 정의되고 나면, 위의 그림에서 Topics 창안에 목록화되어 보일 것입니다.

- b. topic 을 새로 정의 하기 위해서 New 버튼을 클릭합니다,

MODBUS Topic / Definition 창이 열릴 것입니다:

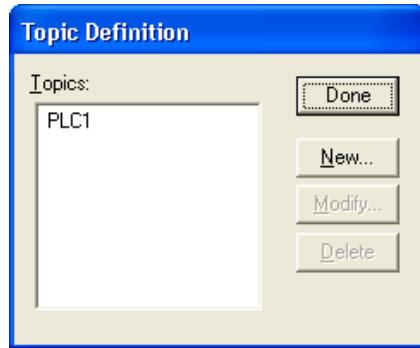


- c. **Topic Name :** 이 공란에는 PLC 와 연결될 유일한 이름을 입력합니다.

(32 글자까지 지원)

Note : InTouch 와 통신을 할 경우, 바로 이 이름이 Access Name definition 에서 topic name 으로 사용됩니다.

- d. **ComPort** : 0| topic 과 연관된 통신 포트를 선택합니다.
 - e. **Slave ID** : PLC 의 Slave ID 를 입력합니다.
 - f. **Slave Device Type** : 기본 설정값이 아닌 slave 장비의 type 을 Drop-down 목록으로 보여줍니다.
 - g. **String Variable Type** : PLC 는 레지스터에 아스키코드의 문자를 저장하기 위해 여기서 설정한 style 을 사용 할 것입니다.
 - h. **Register Type** : 하드웨어에서 지원하는 레지스터 Type 0| BINARY 인지 BCD 인지를 선택하는 옵션입니다.
 - i. **Block I/O Sizes :**
 - Coil Read** : 한 번에 읽기가 가능한 연속된 coil 의 최대 숫자를 입력해야 합니다. 예를 들어, 위의 그림에서는 읽기가 가능한 coil 의 수는 8 에서 2000 사이이며, 8 의 배수인 값들만 허용이 됩니다.
 - Coil Write** : 한 번에 쓰기가 가능한 연속된 coil 의 최대 숫자를 입력해야 합니다. 예를 들어, 위의 그림에서는 읽기가 가능한 coil 의 수는 8 에서 800 사이이며, 8 의 배수인 값들만 허용이 됩니다.
 - j. **Register Read** : 한 번에 읽기가 가능한 연속된 레지스터의 최대 숫자를 입력해야 합니다. 예를 들어, 위의 그림에서는 읽기가 가능한 coil 의 수는 1 에서 125 입니다.
 - k. **Register Write** : 한 번에 쓰기가 가능한 연속된 coil 의 최대 숫자를 입력해야 합니다. 예를 들어, 위의 그림에서는 읽기가 가능한 coil 의 수는 1 에서 100 입니다.
 - l. **Update Interval** : topic 과 연관된 item / point 들을 I/O Server 가 읽거나 던질(poll) 경우, 그 주기를(milliseconds 단위) 입력해야 합니다. 각각의 다른 item / point 들은 각각의 topic 에 서로 다른 업데이트 주기를 설정하거나, 같은 PLC 에 다수의 topic 을 정의 함으로써 각기 다른 비율로써 poll 을 던져 줄 수가 있습니다.
 - m. 창을 닫고, 설정한 Topic 을 적용시키려면, OK 버튼을 클릭합니다.
- Topic Definition 창은 새롭게 정의된 목록과 함께 열릴 것입니다:



- n. 이 창을 닫기 위해서 Done 버튼을 클릭한 후, server 의 프로그램 윈도우로 돌아갑니다.
- o. 기존에 생성되었던 topic 을 수정하려면, Modify 버튼을 클릭합니다.
- p. 기존에 생성되었던 topic 을 삭제하려면, Delete 버튼을 클릭합니다.

Lab 10 – I/O 통신 설정

Modbus Server 설정

- a. Modbus server 를 시작합니다.
- b. COM 포트 설정을 다음과 같이 합니다:

COM Port:	COM1
Protocol:	RTU
Reply Timeout:	3
Baud Rate:	19,200
Data Bits:	8
Stop Bits:	1
Parity:	Even

Define the MODBUS Topic

- c. Topic Definition 을 다음과 같이 생성하고 설정합니다:
 - d. Modbus server 창을 최소화 하고, WindowMaker 창을 최대화 합니다.
- | | |
|------------------|------|
| Topic Name: | PLC1 |
| Com Port: | COM1 |
| Register Type: | BCD |
| Update Interval: | 100 |

Access Name 생성

- e. Access Name 을 추가하고 다음과 같이 설정합니다:
- | | |
|--------------|------------------|
| AccessName: | PLC1 |
| Node: | Blank (local PC) |
| Application: | Modbus |
| Topic: | PLC1 |

- f. 창을 닫기 전에, 새로운 Access Name (PLC1)^{0|} Access Names 창에 정의 되었는지를 확인합니다.

Factory Floor Window 생성

- g. Main 으로 설정할 window 를 새로 하나 생성합니다.
 h. PLC 와 통신하기 위해서는 다음과 같은 태그들을 생성해야 합니다.

Tagname	Tag Type	Access Name	Item
Start	I/O Discrete	PLC1	2049
Gate_Valve	I/O Discrete	PLC1	2050
Valve_A	I/O Discrete	PLC1	2051
Valve_B	I/O Discrete	PLC1	2052
Agitator	I/O Discrete	PLC1	2053
Drain_Valve	I/O Discrete	PLC1	2054

Tagname	Tag Type	Access Name	Item
Tank_Level	I/O Integer	PLC1	40001 (Min/Max EU 0-1000) (Min/Max RAW 0-1000)

- i. Main 윈도우에서, 각각의 태그들을 text 로 추가 합니다: (Start, Gate_Valve, Valve_A, Valve_B, Agitator, Drain Valve, and Tank_Level = ###), 그리고 각 text 객체에는 다음과 같이 적당한 태그들을 등록합니다:

Tagname	Animation Link
Start	Touch Pushbutton – Discrete Value - Toggle, Discrete Text Color - Green/Red
Gate_Valve	Discrete Text Color – Green/Red
Valve_A	Discrete Text Color – Green/Red
Valve_B	Discrete Text Color – Green/Red
Agitator	Discrete Text Color – Green/Red
Drain_Valve	Discrete Text Color – Green/Red
Tank_Level	Value Display – Analog

Runtime 실행을 시키면, 다음과 같은 결과가 나타날 것입니다:

- | Start text 객체에는 클릭을 할 때마다, 녹색 또는 빨간색으로 교대로 변경될 것입니다.
- | 모든 문자 객체는 장치와 연결된 태그의 값에 변경사항이 발생하면, 녹색에서 빨간색으로 변경됩니다.
- | Tank_Level 문자 객체는 태그의 데이터 값을 보여줍니다.

Important Note : 데이터 처리 과정 설명:

'Start'가 참일 경우, 'Gate_Valve'는 열리게 되며, 'Tank_Level'은 데이터 값이 300에 도달할 때까지 계속 상승하게 될 것입니다. 300에 도달하게 되면 'Gate_Valve'는 닫히게 될 것입니다.

'Valve_A'가 열리게 되면, 'Agitator'는 원료를 섞기 시작합니다. 'Tank_Level'이 600에 도달하게 되면, 'Valve_A'는 닫히며, 동시에 'Valve_B'가 열리게 됩니다.

'Tank_Level'이 1000에 도달하게 되면, 'Valve_B'는 다시 닫히며 'Agitator'는 작동을 멈추게 됩니다. 그리고 'Drain_Valve'는 열림으로써 'Tank_Level'을 0 까지 떨어뜨리게 됩니다.

이런 과정들은 'Start' 포인트가 여전히 참의 값을 유지할 경우 계속 반복적으로 발생하게 됩니다. 하지만 'Start' 포인트가 거짓으로 설정 된다면, 'Tank_Level'의 값이 0으로 떨어졌을 경우, 반복작업을 멈추게 됩니다.

- j. 위의 내용을 토대로 가능한 독창적으로, 그리고 많은 animation link 들을 사용하면서 (trend 를 사용해 보거나, wizard, bitmap 등을 사용해보기도 하면서) Application 을 만들어 볼 것을 권해드립니다.

Modbus Server 설정 및 시작

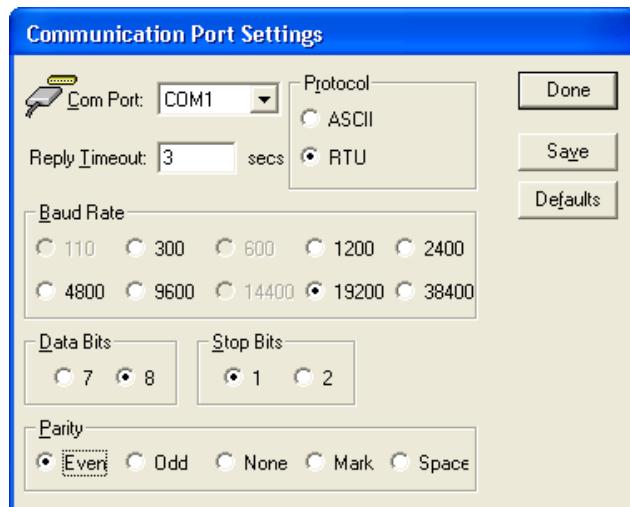
1. Modbus server 을 시작합니다.

Start / Programs / Wonderware FactorySuite / IO Server / Modicon MODBUS.



2. Server 가 실행이 되면, Configure / Com Port Setting 를 클릭합니다.

다음 그림과 같이 설정환경이 이루어졌는지를 확인합니다.

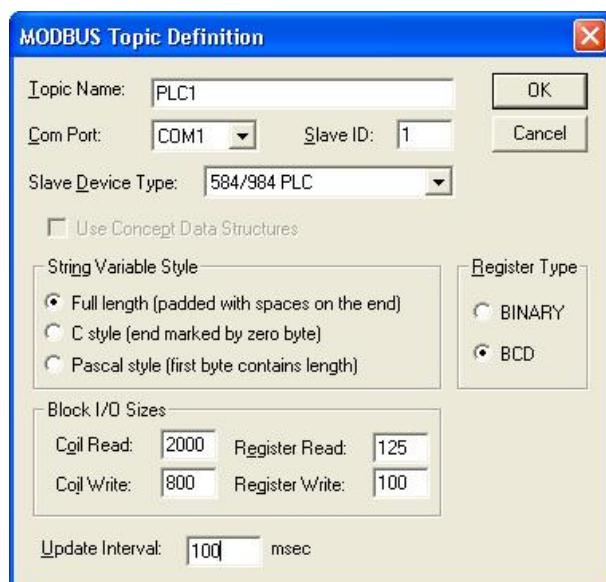


3. Done 버튼을 클릭합니다.

MODBUS Topic 정의

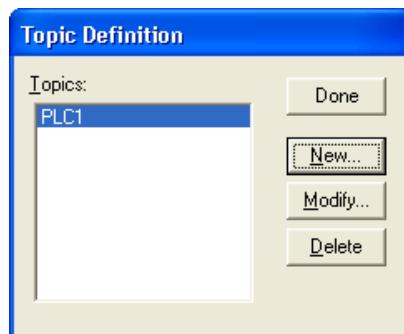
4. Configure / Topic Definition 을 클릭합니다.
5. New 버튼을 클릭합니다.

MODBUS Topic Definition 윈도우가 나타날 것입니다:



6. Topic Name 의 공란에 PLC1 을 입력합니다.
7. Com Port 공란에 COM1 이 선택되었는지를 확인합니다.
8. Register Type 으로 BCD 를 선택합니다.
- Update Interval 에는 100 이라는 값을 입력합니다.
9. OK 버튼을 클릭합니다.

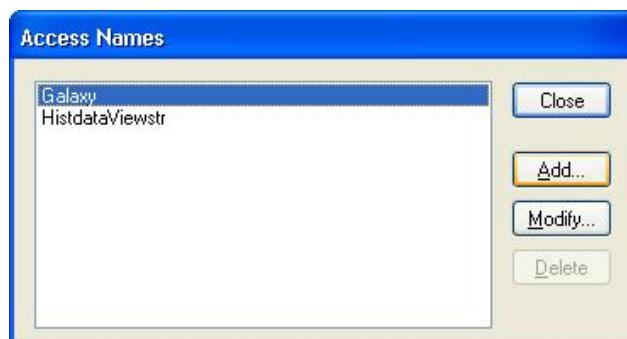
Topic Definition 창의 Topics 영역에 다음 그림과 같이 보일 것입니다:



10. Done 버튼을 클릭합니다.
11. Modbus server 를 최소화 시킵니다.
12. WindowMake 를 최대화로 창을 키웁니다.

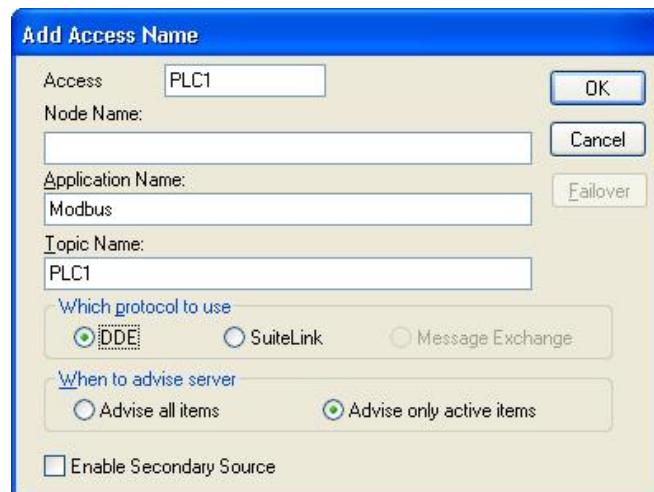
New Access Name

13. Application Explorer(Classic View)의 Tools 영역에서 Configure 옵션을 클릭합니다.
14. Access Names 을 더블-클릭합니다.
15. Add 버튼을 클릭합니다.



Add Access Names 창이 열리게 될 것입니다.

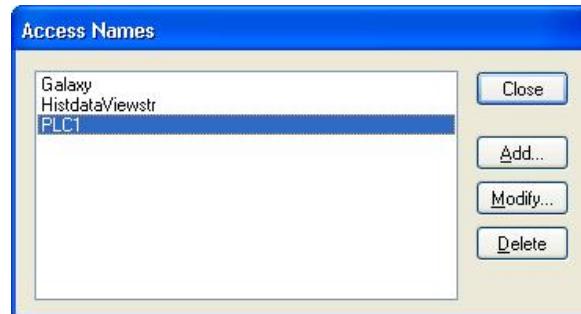
16. 다음 그림과 같이 각각의 정보들을 입력합니다.



17. 새로 생성된 Access Name을 저장하기 위해서 OK 버튼을 클릭합니다.

생성된 Access Name은 현재 Access Names 창에서 보일 것입니다.

18. Close 버튼을 클릭합니다.



Section 3 - Troubleshooting I/O

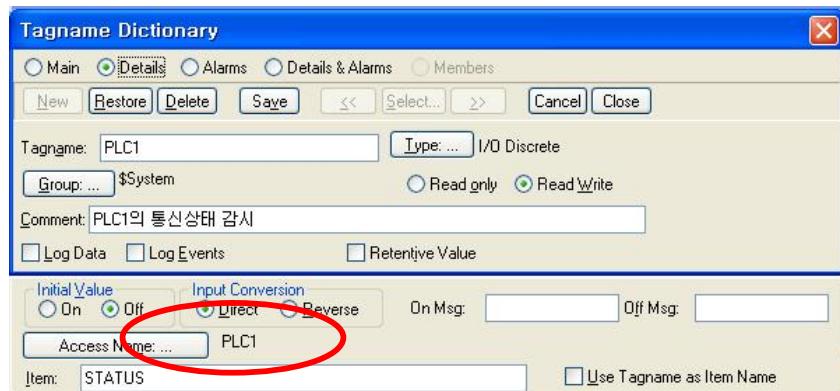
DDE 서버 통신 상태 감시하기

각 사용되는 *Topic* 이름마다 DDE 서버와의 통신 상태를 감시하기 위해 사용할 수 있는 내장된 이산 *Item*이 있습니다. 이산 *Item*, **Status**는 PLC 와의 통신이 실패할 때 0에 설정되며 (케이블이 빠졌거나 PLC 가 깨쳤거나 등등) 통신이 성공적이면 1로 설정됩니다.

Note : Status 항목을 사용하여 Topic의 상태를 감시할 때, 감시하는 Topic에 적어도 하나의 I/O 포인트가 활성 중이어야 합니다.

InTouch에서, 태그명을 정의하고 'Status'라는 단어를 *Item* 이름으로 사용하여 PLC를 위해 구성된 Topic과 연관시킴으로써 서버 통신의 상태를 읽을 수 있습니다. 'Status'가 반드시 아이템 명으로 사용되어야 합니다.

예를 들어, 'PLC1'이라는 AccessName의 DDE 통신 상태를 감시하기 위해 아래와 같이 Tag를 등록하여 사용합니다.



아래와 같은 DDE 참고 공식을 셀에 입력함으로써 엑셀에서 PLC 통신의 상태를 읽을 수 있습니다.

=MODBUS|PLC1!STATUS

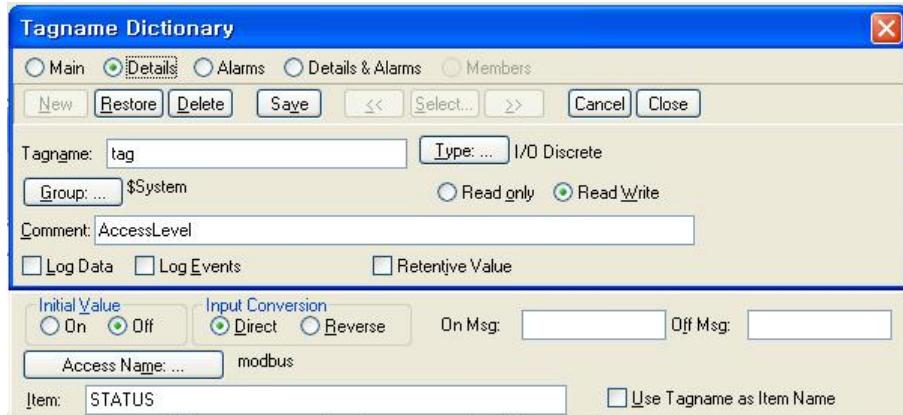
DDE 통신 상태 감시하기

WindowViewer 는 특정한 DDE 통신의 상태를 감시하는데 사용할 수 있는 내장된 **IOStatus**(INTOUCH 7.0 이전 버전에서는 **DDEStatus**)라 하는 *Topic* 이름을 지원합니다.

InTouch에서 DDEStatus 사용하기

예를 들면, **WindowViewer(VIEW)**가 WonderWare Modbus I/O Server로 *Topic* 이름으로 **PLC1**이라 정의된 PLC와 통신한다 가정해 봅니다.

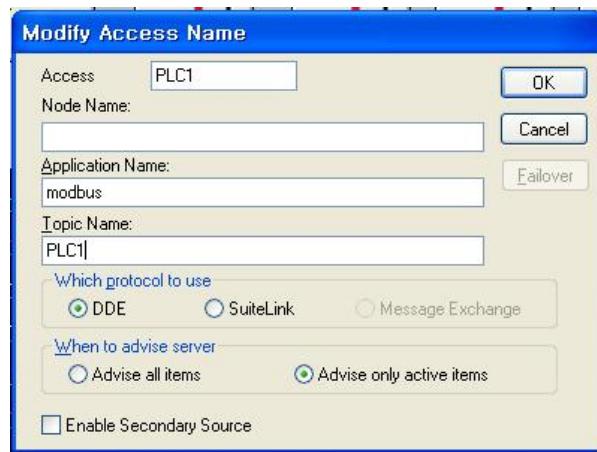
1. **Special/Tagname Dictionary** 를 클릭하면 **Tagname Dictionary** 대화 상자가 나타납니다.
2. 먼저, DDE 이산 태그명(간단하게 하기 위해 감시하고자 하는 *Topic* 이름과 동일한 이름)을 사용하기로 합니다.



3. 다음, 이 DDE 태그명을 *Topic* 이름을 위해 **IOStatus** 를 사용하는 access name 정의와 연관 지어야 합니다. 이렇게 하려면, **Access Name** 버튼을 클릭하여 초기의 **Access Name** 대화 상자를 엽니다.



PLC1(감시하고자 하는)이라는 DDE access name 정의가 이미 존재하는 것을 주시합니다. 이것이 바른 DDE access name (그의 Topic 이름이 **PLC1**)인 것을 확인하려면 **Modify** 버튼을 클릭하면 다음 그림을 볼 수 있습니다.



4. 태그명과 토픽 명을 같게 주었기 때문에, 이 예에서 정확한 토픽명을 가지고 있는 access name 을 찾는 것이 쉽습니다.
5. **Cancel** 버튼을 클릭하여 대화상자를 닫고 처음의 **Access Name** 대화상자로 돌아갑니다. **Add** 버튼을 클릭합니다. **Access Name** 대화 상자가 나타납니다.

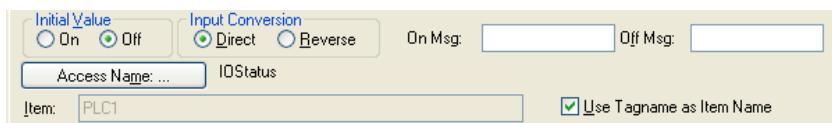


6. **Access Name** 상자에 **IOStatus** 를 넣습니다.
7. WindowViewer 의 상태를 모니터링 할 것이기 때문에, **Application Name** 상자에 "View."를 넣습니다.

8. Topic Name 상자 안에, InTouch 내부 토픽(IOStatus)를 넣습니다.
9. 대화상자를 닫기 위해 OK 를 클릭 합니다. 초기 **Access Name Definition** 상자가 나타나고, 새로운 **Access Name ‘IOStatus’**가 나타납니다.



10. 대화 상자를 닫기 위해 Close 를 클릭 합니다, 그리고 **I/O Discrete** 태그명에 새로운 **Access Name** 을 연결 시킵니다.



11. **Item** 상자에 모니터 하기 위한 **Access Name** 을 넣습니다. 이 예에서는 **PLC1**. 태그명과 access name 같기 때문에, **Use Tagname as Item Name** 을 사용 할 수 있습니다.

Note: IO 통신을 감시하기 위해 내장된 Topic **IOStatus**(InTouch 7.0 이전 버전은 **DDEStatus**)를 사용하면 DDE 액세스 이름이 항상 *Item* 이름으로 사용됩니다.

엑셀에서 IOStatus 사용하기

같은 정보를 스프레드시트 셀의 공식에 입력함으로써 같은 유형의 감시를 위해 엑셀이 사용될 수도 있습니다. 예를 들면, 위와 같이 같은 Topic을 감시하려면 아래처럼 입력합니다.

=view|IOStatus!PLC1



Module 9

Advanced Tag Functionality

Section 1 – Advanced I/O Functions

Section 2 – Indirect Tags

Lab 11 – Creating and Using Indirect Tags

Section 3 – DBDump and DBLoad

Section 4 – Optimizing the Tagname Dictionary

Lab 12 – Deleting Tags

Section 1 – Advanced I/O Functions

이 장에서는 IOSetAccessName 와 I/OSetItem Functions, Dynamic Reference Addressing 과 같은 고급 I/O기능을 설명합니다.

I/OSetAccessName Function

IOSetAccessName은 Runtime 중에 Accessname의 node name, application name 또는 Topic name을 변경할 수 있습니다. 이 기능은 InTouch의 이중화를 구현할 수 있도록 합니다.

Syntax IOSetAccessName (“AccessName” , “NodeName” , “AppName” , “TopicName”) ;

Arguments	Description
AccessName	Application 에 정의 된 AccessName으로 써, 새로운 AppName 또는 Topic Name 값을 실제 문자나 메시지 태그로 변경할 수 있습니다.
NodeName	실제 문자나 메시지 태그로 새로운 NodeName 을 지정합니다.
AppName	실제 문자나 메시지 태그로 새로운 Application Name을 지정합니다
TopicName	실제 문자나 메시지 태그로 새로운 Topic Name을 지정합니다.

Access Name, Node Name, Application Name 그리고 Topic Name의 값은 문자값으로 지정되거나, 다른 InTouch 태그 또는 Function으로 제공되는 문자값입니다. 만약에 빈 문자값 (“”) 이 지정되면 그것은 현재 값을 유지한다는 것입니다.

예를 들어, AccessName MyAccess1 을 Script를 사용해서,. Application은 Excel, Topic은 Book1, Node Name은 현재의 값을 유지하고 변환 시킬 수 있습니다.

IOSetAccessName (“MyAccess1” , “” , “Excel” , “Book1”)

Topic에 빈 문자값 (“”) 이 지정되면, AccessName의 ApplicationName 값은 업데이트 되고

TopicName 은 현재의 상태를 유지합니다.

IOSetAccessName (“MyAccess2” , “” , “Excel” , “”) ;

마찬가지로, Application Name에 빈 문자가 지정되면 AccessName의 Topic 값은 업데이트되고

Application Name 은 현재 상태를 유지합니다.

IOSetAccessName (“MyAccess3” , “” , “” , “Book2”);

예를 들어 MayAccess3 이라는 AccessName의 Topic 을 바꾸기 위해서는

IOSetAccessName (“MyAccess3” , “” , “” , “PLC2”);

이 예제는 PCL 이중화에 사용될 수 있습니다.

Note: IOSetAccessName 이 진행될 때, 새로운 설정으로 변환되기 위해서는 약간의 시간이 필요로 됩니다. 이 시간 동안 새로운 Topic에 대한 Poke, Write 신호를 놓칠 수도 있습니다.

IOSetItem Function 사용하기

IOSetItem Function 은 I/O 형태 태그의 .Reference 필드를 설정하는데 사용됩니다.

Syntax IOSetItem (“Tagname” , “Accessname” , “Item”);

Parameters	Description
TagName	InTouch I/O TagName이며, “” 안에 입력합니다.
AccessName	Tag의 변경될 AccessName
Item	Tag의 변경될 Item Tagname, AccessName, Item 값은 문자로 지정되거나 메시지 태그를 사용하여 지정합니다.

예를 들어, MyTag1 이라는 태그의 .Reference 필드는 다음 스크립트에 의해 Excel 이라는 AccessName 과 R1C1 이라는 Item으로 변경될 수 있습니다.

IOSetItem(“MyTag1” , “excel” , “R1C1”);

또는,

```
Number =1;
TagNameString = “MyTag” + Text(Number, “#”);
IOSetItem ( TagNameString , “ excel” , “R1C1” );
```

만약에 AccessName이나 Item이 빈 문자로 지정되면, Tag는 아무 변화가 없을 것입니다.

다음의 예에서 MyTag2는 아무 변화도 일어나지 않습니다.

IOSetItem (“MyTag2” , “” , “”);

Item에만 빈 문자가 지정된다면, 현재의 Item 값은 그대로 유지되고 AccessName 값만 업데이트 됩니다. 다음은 Item 변경없이 MyTag3이라는 Tagname의 AccessName을 Excel2로 변경하는 예입니다.

IOSetItem (“MyTag3” , “excel2” , “ ”);

Note : Double – quotes 사이에 빈 공간이(space) 존재한다면, 인수에 빈칸으로 설정되고 동작하지 않습니다.

마찬가지로 AccessName이 빈 문자로 지정되면, AccessName의 값은 유지되고 Item 값만 바뀌게 됩니다. 다음은 MyTag3의 Item 을 R1C2로 변경시키는 예제입니다.

```
IOSetItem ("MyTag3", "", "R1C2");
```

Dynamic Reference Addressing (DRA)

Dynamic Reference Addressing 기능은 하나의 태그명에 여러 개의 Data Source를 연결하도록 해줍니다. I/O 타입의 Tag 의 Reference 에 값을 참조하여, 동적으로 변화될 수 있는 Data source를 연결합니다.

I/O 태그는 Data source 의 address 와 관련되어 있습니다. Reference 필드의 정확한 문법은 다음과 같습니다.

Syntax	Description
Tagname.Reference="accessname.item"	Access Name 과 Item 을 변경합니다.
Tagname.Reference=".item"	같은 AccessName에 Item 을 변경합니다.
Tagname.Reference="accessname"	Access Name 을 변경합니다.
Tagname.Reference=""	Tagname 이 동작하지 않습니다. 만약 Access Name 이나 Item 이 지정되지 않는다면, 필드의 현재값은 아무 변화가 없습니다.
.ReferenceComplete	1 = 유효한 I/O Tag 0 = AccessName , Item 이 유효하지 않음

Note : 동적 참조는 진단 Application 과 같은 임시적으로 필요로 되는 Data 를 보기 위해 사용됩니다. 이것은 이상적으로 문제해결을 할 수 있는 Tool 을 만들 수 있습니다. 태그명의 data source 가 변경될 수 있기 때문에, 동적 참조는 알람 상태를 위해 지속적으로 변경 되거나 변경 하지 않으며, 저장될 필요가 있는 Data 에 사용되어서도 안됩니다.

Section 2 – Indirect Tags

이 장에서는 간접태그 (Indircet Tag)의 장점과 단점을 논의해보고, 개념과 예제를 살펴봅니다.

Indirect Tags

간접태그는 하나의 윈도우에 다중소스를 재할당 할 수 있도록 합니다. 예를 들어, 윈도우에 변화되는 값을 기반으로 하는 모든 태그를 변경하도록 Data Change Script를 생성할 수 있습니다.

태그를 간접태그에 연결할 때, 태그와 간접태그는 .fields, script 와 같은 속성들을 서로 정확하게 복제 하게 됩니다.

태그의 값이 변경된다면 간접태그에도 반영이 됩니다. 만약 간접태그의 값이 변경 된다면 태그의 값 또한 변경되게 됩니다.

간접태그에도 retentive 설정을 할 수 있어, 마지막 설정 값을 Application이 재 시작할 때 가져와서 재설정 할 수 있습니다.

간접태그는 Name 필드에 의해서 설정 됩니다. 예를 들어 Setpoint라 하는 간접아날로그태그를 생성한다고 가정할 때, Script 에는 아래와 같이 표현됩니다. 이로써 Setpoint1은 Setpoint 값의 Source 가 되며, 반대로 Setpoint 는 Setpoint1 값의 Source 가 됩니다.

```
Setpoint.Name = "Setpoint1";  
Setpoint.Name = Setpoint1.Name;
```

또한 간접태그에 태그명을 연결시킬 수도 있습니다. 예를 들어 매시간 “Number” 라는 태그값의 변화에 따라 DataChange Script를 생성한다고 할 때, 간접태그 “Setpoint” 는 Number의 값에 따라 변경되게 됩니다.

```
Number=1;  
Setpoint_Name = "Setpoint" + Text(Number,"#");
```

스크립트가 실행 될 때, 아날로그 태그인 Number의 값이 Text로 바뀌고, Setpoint 에 더해져서 Setpoint.Name 과 Setpoint1 은 동일하게 만들어집니다.

아날로그 형태의 간접태그는 Integer 와 Real 태그에 사용되며, 간접태그는 같은 형태의 태그와 연결될 수 있습니다.

간접태그 사용의 이점

- | 원도우 개발의 저장
- | 성능향상
- | 알람과 이력 트랜드의 지원
- | .field 지원 (태그의 속성)
- | 원격태그참조 (Remote Tagname Referencing)
- | Dynamic Reference Addressing 작업

간접태그의 단점

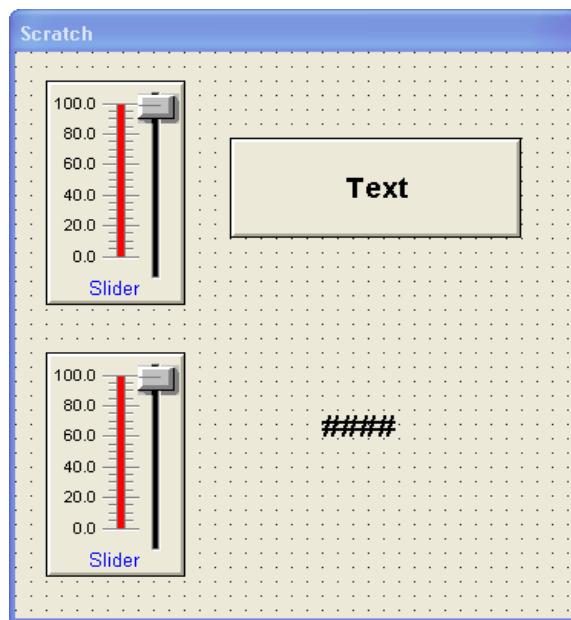
- | 태그 수의 증가
- | Application 개발 계획 필요 (태그명, 스크립트)
- | 어떠한 경우 간접태그가 지정된 특정Tag가 지워졌는지 아닌지를 확인해야만 합니다.

Note : 간접태그에 연결된 태그가 존재하지 않는다고 해서, 그 태그가 생성되지는 않습니다. 마지막으로 연결된 태그가 유지되게 됩니다.

Lab 11 – 간접태그의 생성과 사용

간접태그 생성

- a. Scratch 윈도우에 두 개의 슬라이더, 버튼 한 개, Text (#####)를 삽입합니다.



- b. 두개의 Memory Integer 태그를 정의 합니다 (Tag_A , Tag_B)
태그의 Min Value 는 0, Max Value 는 100으로 지정합니다.
c. 슬라이더에 Tag_A를 연결하고, 슬라이더의 라벨을 Tag_A라고 지정합니다.



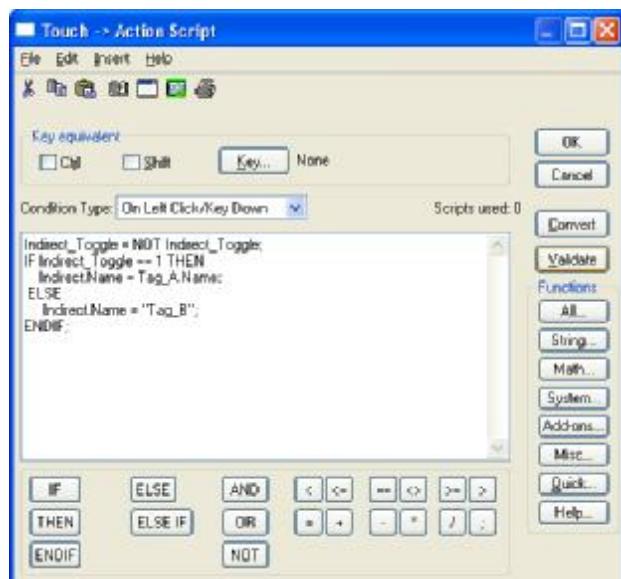
- d. 다른 슬라이더에 Tag_B를 연결하고, 라벨에 Tag_B라고 지정합니다.

버튼 객체 구성하기

- e. 버튼의 Touch Pushbutton Action – On Key Down 스크립트에 다음을 입력합니다.

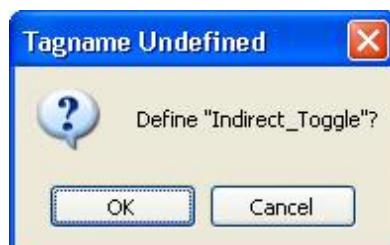
```

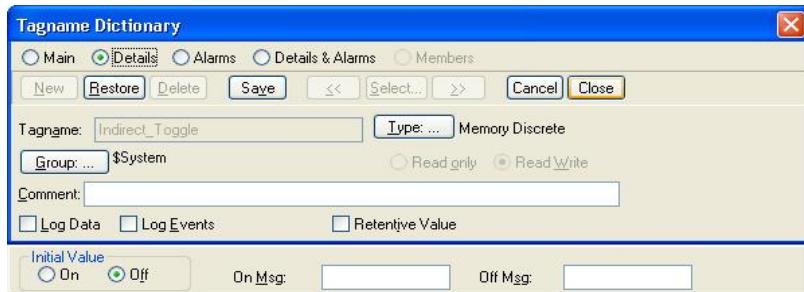
| Condition Type : On Left Click / Key Down
| Script : Indirect_Toggle = NOT Indirect_Toggle;
    IF Indirect_Toggle == 1 THEN
        Indirect_Name = Tag_A.Name
    ELSE
        Indirect.Name = "Tag_B";
    ENDIF;
```



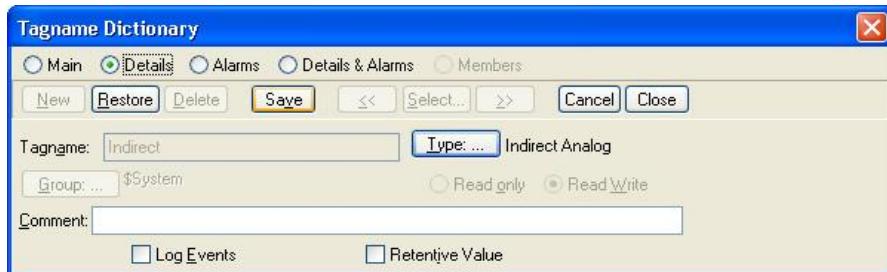
- f. Validate 버튼을 클릭하여 Script 문장을 점검합니다.

- g. Memory Discrete 타입으로 Indirect_Toggle 태그를 정의하고 Close 합니다.





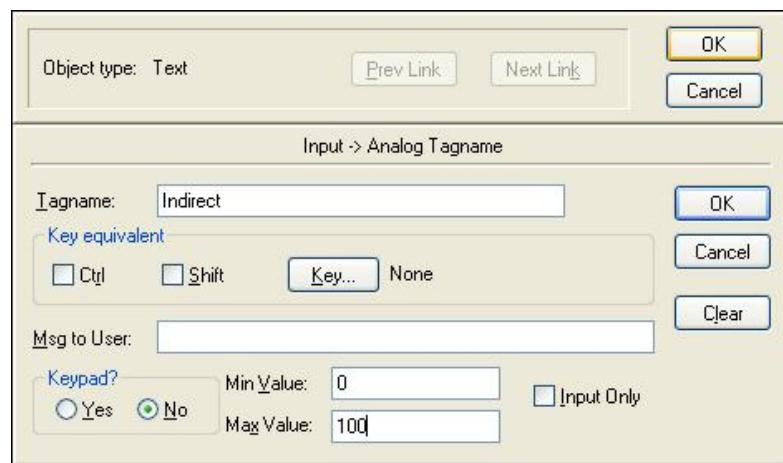
- h. Indirect Analog 타입으로 Indirect 라는 이름의 태그를 정의하고 Close 합니다.



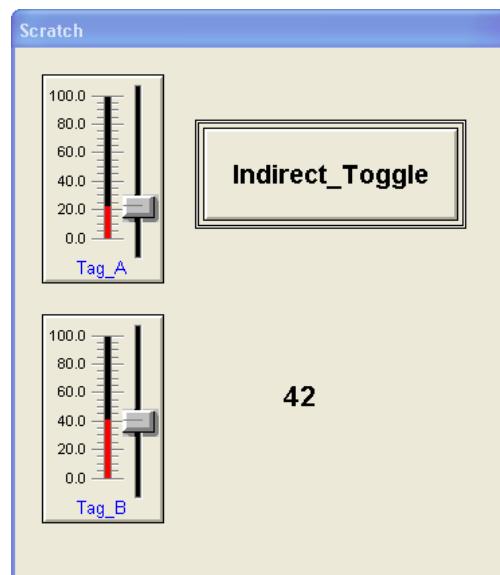
- i. 버튼의 이름을 Indirect_Toggle 로 입력합니다.

Text 객체 구성하기

- j. Text 객체에 애니메이션 링크의 UserInput-Analog를 구성하고 다음과 같이 설정합니다.
- | Tag : Indirect
 - | Min Value: 0
 - | Max Value: 100



- k. Runtime모드에서 슬라이더를 움직이며 Test 하고, Indirect_Toggle 버튼을 클릭하며 결과를 관찰 합니다.



Section 3 – DBDump and DBLoad

이 장에서는 DBDump 와 DBLoad 유ти리티를 설명합니다.

소개

태그명 유ти리티 프로그램에는 두 가지가 있습니다 : DBDump , DBLoad

DBDump : InTouch Application의 Tagname dictionary를 다른 프로그램(ex :Excel)에 편집하고 열어볼 수 있는 Text 파일로 Export 합니다.

DBLoad : Tagname Dictionary 형식의 파일 (Excel에서 만들어지거나 DBDump된 파일) 을 InTouch Application으로 import 합니다.

이 두 가지 유ти리티는 Tagname dictionary가 Application과 분리되어 복제, 수정, 개발되고 다시 합쳐질 수 있도록 지원합니다.

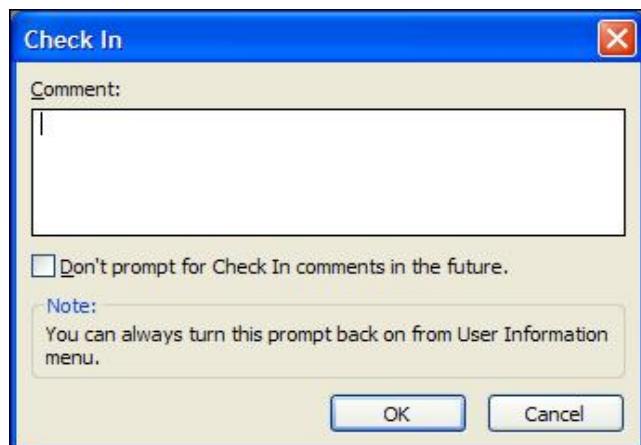
DBLoad 유ти리티는 SuperTag instance를 생성하기 위한 InTouch TemplateMaker로도 사용 될 수 있습니다.

Note: DBDump 와 DBLoad 유ти리티는 Application 형태에 따라 InTouch Application Manager 나 Archestra IDE에서 실행시킬 수 있습니다. 또한 유ти리티를 사용하기 위해서는 이전버전에서 개발된 Application 은 반드시 Converting 을 선행해야 합니다.

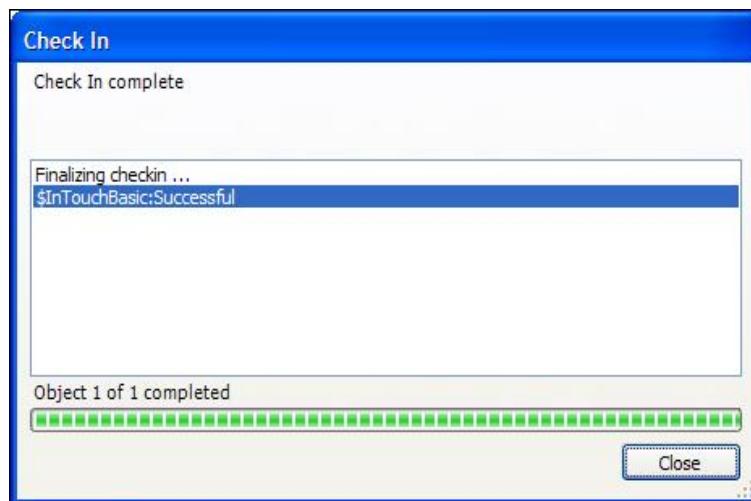
DBDump 유ти리티

다음의 과정은 InTouch Application의 Tagname dictionary를 어떻게 추출하는지를 보여줍니다.

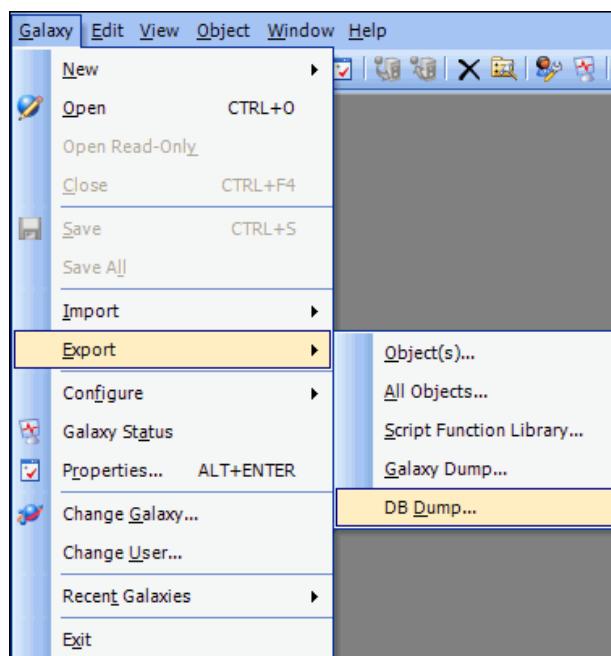
- 만약에 WindowViewer 가 실행되어 있다면 종료시켜야 합니다.



- b. WindowMaker를 종료하고, Check In 대화상자의 OK를 클릭합니다.



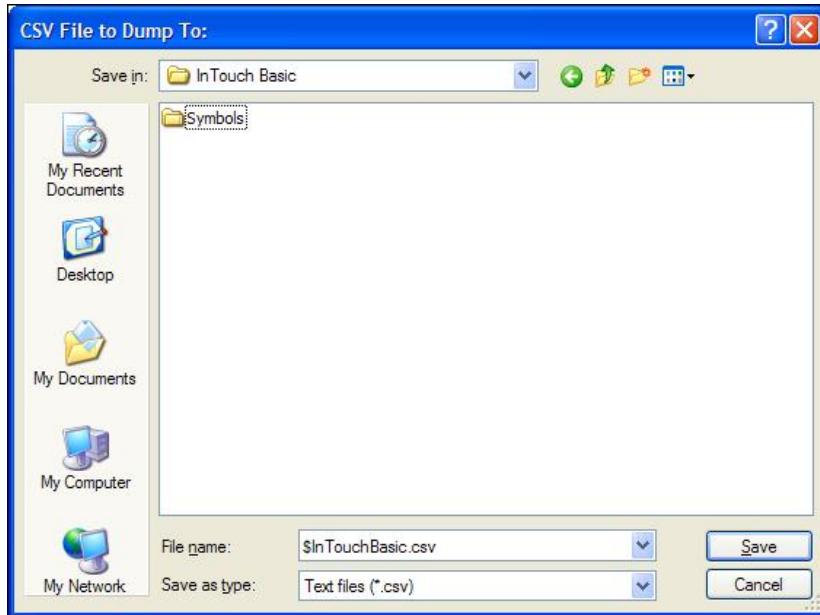
- c. Check In Complete 메시지가 표시되면 Close를 클릭합니다.



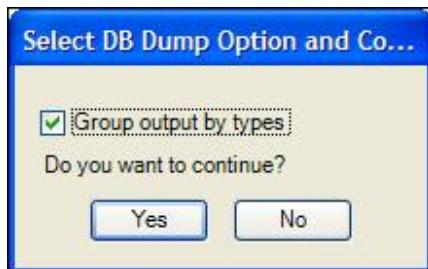
- d. Archestra IDE 에 Galaxy/Export/DBDump를 클릭합니다.

CSV File to Dump To 대화상자가 나타납니다. 이 파일은 현재 Application의 이름으로, 확장자는 .csv (콤마분리형변수)로 정의 됩니다. 이 경우에는 \$InTouchBasic.csv

Note: 기본적으로 파일은 export 되는 Application의 이름으로, 확장자는 .csv로 정의됩니다. 이 이름은 재설정 될 수 있지만, 이름의 앞에는 \$ 문자로 시작해야 합니다.



- e. Save를 클릭합니다.



Select DB Dump Option and Continue 대화상자가 나타납니다. Grouping 옵션을 선택하면 태그명을 태그형태 또는 알파벳순으로 추출할 수 있습니다. 이 옵션은 기본적으로 선택되어져 있습니다.

- f. 기본설정에서 Yes를 클릭하면 태그타입 순서로 추출됩니다.

잠시 후, 선택된 Application 디렉토리로부터 데이터베이스 정보가 지정된 파일로 다운로드 될 것입니다. Database Dump 메시지 박스에서 Dump successful 메시지를 확인합니다.



g. OK를 클릭합니다.

Database 입력 파일 생성하기

DBDump 와 DBLoad 데이터베이스 유ти리티는 Tagname dictionary에서 batch Type의 작업을 수행하기 위해 사용되는 도구입니다. 데이터베이스 입력파일은 콤마분리형변수파일형식을 지원하는 모든 프로그램에서 생성할 수 있습니다. (.CSV)

데이터베이스 입력파일은 .CSV형태의 파일로 저장되어야만 합니다. 입력파일이 생성되면, DBLoad 프로그램은 현재의 Application의 데이터베이스 파일 안에 포함되어있는 데이터에 파일을 로딩하거나 합치는데 사용됩니다.

데이터베이스 입력파일 Format

데이터베이스 입력 파일의 첫번째 라인은 DBLoad로 Application에 통합될 때의 작업 :MODE를 지정합니다.

모든 데이터 레코드는 태그명의 :type으로 시작하며, 다음은 각각의 데이터 레코드의 명칭을 나열합니다. (콤마분리형)

```
:Mode=test  
:IOMsg,Group,Comment,Logged,EventLogged,Event Logging Priority,...
```

각각의 태그명과 데이터레코드에 대한 명칭을 나타냅니다.

Excel에서 .csv 파일이 열릴 때, 자동으로 구분자인 콤마를 컬럼으로 인식해 데이터 레코드를 분리하게 됩니다.

.CSV 파일을 Notepad에서 열면, 데이터 레코드는 각각 콤마로 분리 됩니다.

DBLoad 기능을 사용하게 되면

- | Logging 속성을 변경할 수 있습니다.
 - | Tagname dictionary의 내용을 쉽게 한꺼번에 변경할 수 있습니다.
 - | 태그의 속성을 변경할 수 있습니다.

만약에 태그의 이름을 재설정하고 싶다면, **DBLoad**는 원래의 태그를 지우는 것이 아니라 새로운 태그를 생성하게 됩니다.

DBLoad를 사용해서 태그를 지울 수가 없습니다. 다만 새로운 태그를 추가 할 수는 있습니다.

입력 파일의 작업모드

다음의 목록은 입력파일의 작업모드의 종류를 나타내며, 중복되는 태그가 로딩될 때 각각의 모드가 어떻게 실행되는지를 설명합니다.

:MODE=ASK – 입력 또는 태그명 목록에 존재하는 태그명을 바꾸고 나서 새로이 정의된 태그를 Tagname dictionary에 넣습니다.

:MODE=REPLACE – 존재하는 목록을 삭제 후 새로운 목록으로 대체합니다.

:MODE=UPDATE – 입력파일에 정확하게 정의된 필드만 존재하는 Tagname dictionary에 덮어씌웁니다.

:MODE=TEST – 이 모드에서는 , DBLoad 가 replace mode처럼 수행됩니다. 그러나 Tagname dictionary (Database) 은 변경되지 않습니다.

TEST Mode 는 파일을 로딩하기 전에 입력파일을 검사하여, 에러사항과 Line number를 나타내 줍니다. TEST Mode를 사용해서 에러 사항을 검사한 후에 DB를 Update 나 Replace mode로 변경하여 로딩 할 수 있습니다.

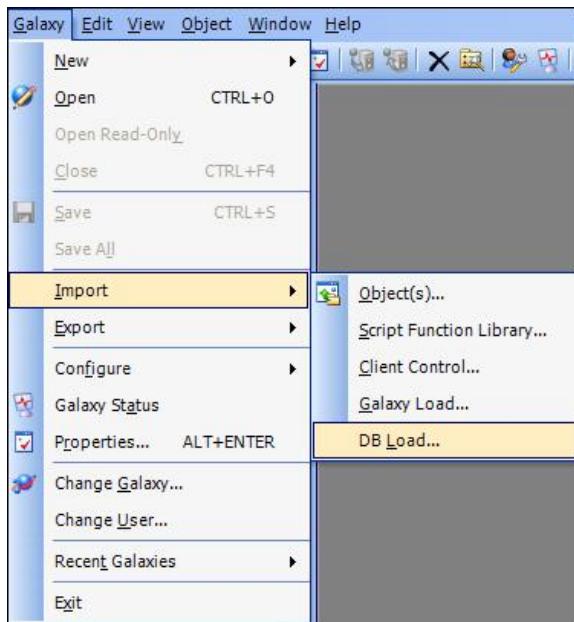
DBLoad 유틸리티

DBLoad 유틸리티를 사용하여 다른 Application으로부터 태그를 import 할 수 있으며, 그리고 나서 원도우를 가져와서 사용 할 수 있습니다.

Note: 시스템에 문제가 생길 경우 , export 된 백업파일을 사용하여, DBLoad 를 실행합니다.

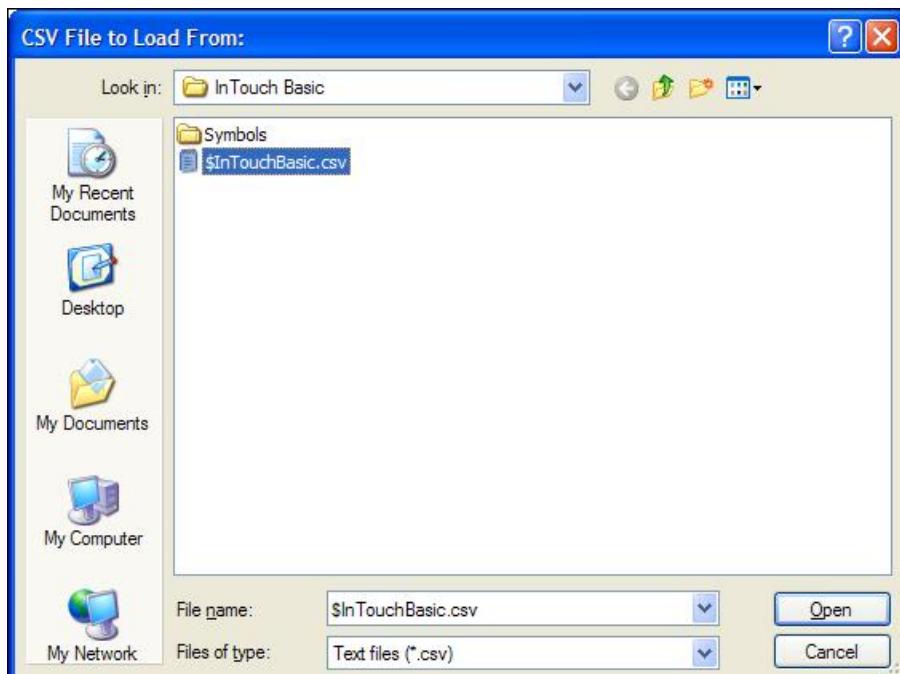
다음은 InTouch Application의 데이터베이스에 입력파일을 로딩시키는 과정을 보여줍니다.

- a. WindowMaker 와 WindowViewer를 종료합니다.
- b. ArchestrA IDE를 엽니다.



- c. 새로운 InTouch application 에 입력파일 데이터베이스를 로딩하기 위해서, Galaxy/import/DBLoad를 클릭합니다.

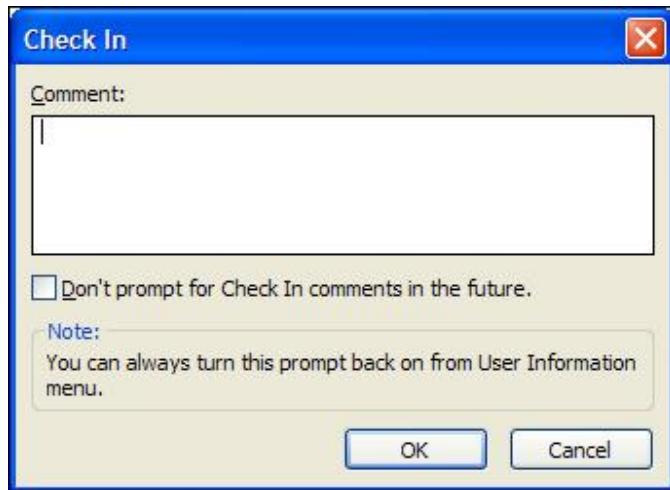
CSV File to Load From 대화상자가 나타납니다.



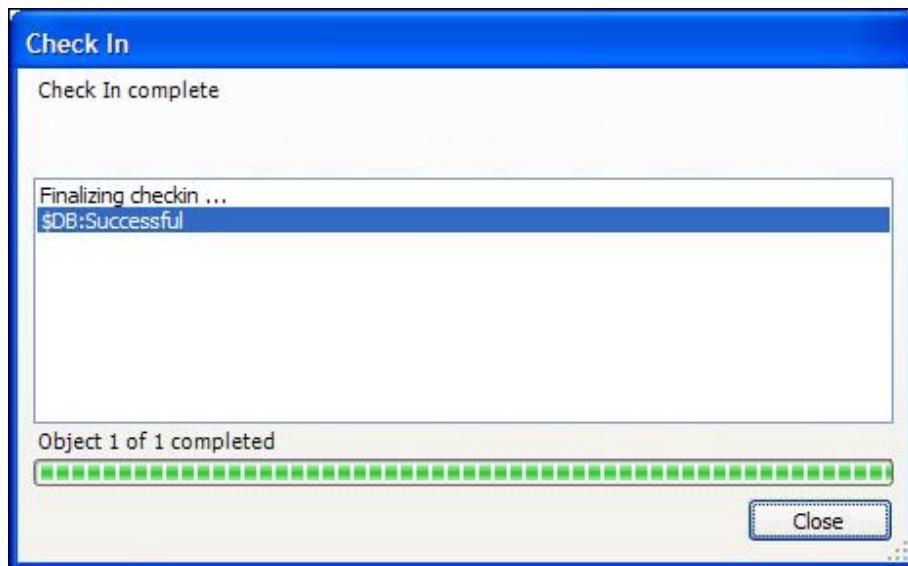
선택한 파일에 포함된 데이터베이스 정보는 선택된 Application의 Tagname dictionary에 업로딩 될 것 입니다. 잠시 후, Database Load 메시지 박스가 로딩과정을

성공했다는 메시지를 보여줄 것입니다.

- d. 메시지를 확인 한 후 , OK를 클릭합니다.
- e. ArchestrA IDE 는 InTouch Application 을 Check In 할지 확인합니다.



- f. OK를 클릭합니다.
- g. ArchestrA IDE 는 Check in Complete 메시지 박스를 보여줍니다.



- h. Close를 클릭합니다.

Section 4 - Tagname dictionary의 활용

이 장에서는 설정되어 있는 링크 및 태그를 편집하기 위한 몇 가지 방법을 살펴봅니다.

태그의 치환, 변환, 삭제 및 Cross Reference 유ти리티 사용에 대해 배워봅니다.

태그 삭제

InTouch에서 태그의 삭제는 어렵게 작업하도록 되어있습니다.

- | WindowViewer 가 실행되어 있다면, 태그 삭제를 할 수 없습니다.
- | 태그가 Script 나 애니메이션 링크에 사용되었다면, 그 태그는 삭제될 수 없습니다.
태그가 어디에 사용되었는지 찾기 위해, Cross Reference 유ти리티를 사용합니다.
- | 다시 말해 Tagname dictionary으로부터 사용되지 않은 태그는 지우기 위해 다음의 과정을 진행해야 합니다.
 - | WindowViewer를 종료합니다.
 - | Cross Reference 유ти리티를 사용하여 태그가 사용된 것을 모두 지웁니다.
(애니메이션링크, Script)
 - | Special 메뉴에서 Update Use Count를 실행합니다.
 - | 삭제를 확인합니다.

WindowViewer 가 실행되고 있거나, 태그가 Application 안에서 사용되고 있다면 Delete 버튼은 활성화 되지 않습니다.

InTouch Cross Reference 유ти리티

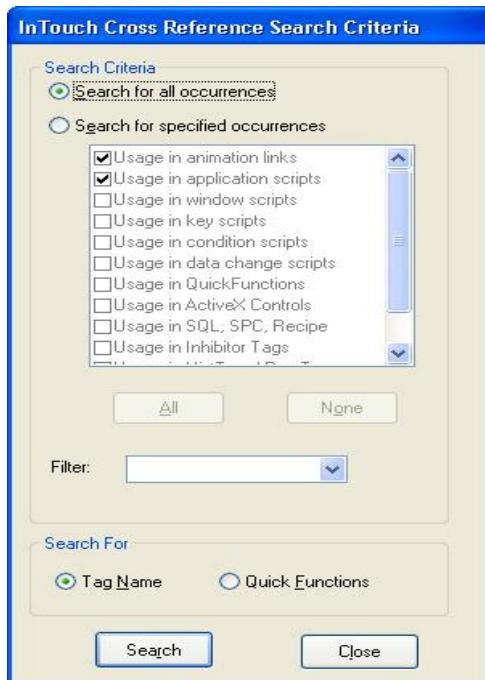
InTouch Cross Reference 유ти리티를 통하여 태그나 SuperTag가 애니메이션링크, Wizards, Script, QuickFunction, ActiveX Controls 또는 SPC Pro, SQL Access Manager, Recipe Manager에서 사용되었는지 확인할 수 있습니다.

Wizards, ActiveX controls, 애니메이션 링크와 같은 모든 객체에 대해, Cross Reference는 태그명이 링크되어 있는 모든 객체의 원도우 이름과 좌표를 표시 합니다. 또한 어떤 Script 나 QuickFunction에서 사용되었는지 찾아서 보여 줍니다.

Cross Reference 유ти리티는 WindowMaker에서 다른 작업과 동시에 사용할 수 있습니다.

InTouch Cross Reference 유ти리티 사용하기

- a. Special/Cross Reference를 클릭합니다.
InTouch Cross Reference Search Criteria 대화상자가 나타납니다.

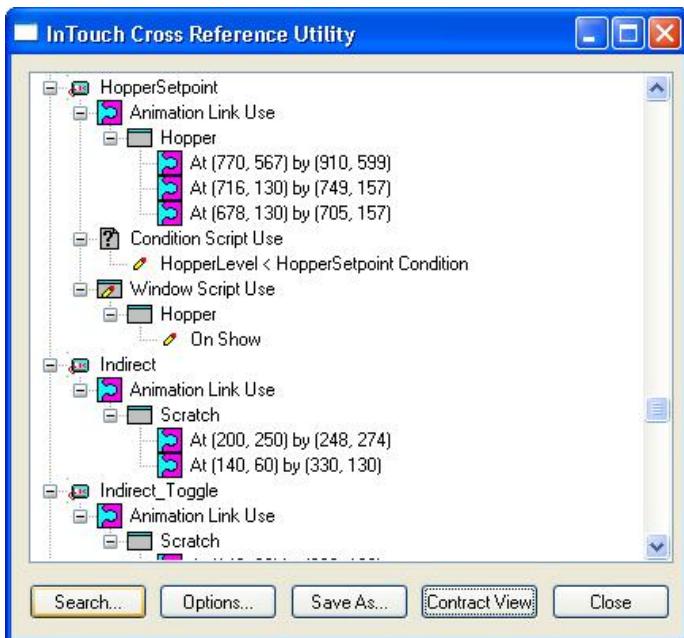


Search Criteria : 검색 범위를 제한 할 수 있습니다. 검색 범위는 다음의 옵션선택에 따라 결정됩니다.

- | **Search For all occurrences :** 애니메이션 링크, Script, SPC, SQL Access Manager, Recipe Manager 등 태그나 Super 태그의 모든 사용을 검색합니다.
- | **Search for specific occurrences :** 지정된 옵션 안에서 태그명이나 Super Tag를 검색합니다.
- | 예: **Usage in window scripts**를 선택하면, Window Scripts의 사용에 대해서만 검색합니다.
- | **Filter :** 검색 하길 원하는 태그명이나 Quick Function 을 입력합니다.
- | **Search :** 검색을 시작합니다.

Cross Reference 검색 결과

Cross reference 검색이 실행되면, InTouch Cross Reference Utility 대화상자가 나타나며, 검색을 지정했던 태그나 Function 이 사용된 목록을 보여 줍니다.



Cross Reference Utility Interface Icons

Icon	Description
	태그명이나 SuperTag가 InTouch 객체나 Script,Wizard,add-on 프로그램등에 사용된 것을 보여 줍니다. 클릭하여 펼칩니다.
	클릭하면 확장된 내용을 닫습니다.
	Application 의 Tagname dictionary에 정의된 태그와 SuperTag를 표시합니다. 그러나 객체에는 사용되지 않았음을 보여줍니다.
	애니메이션링크나 QuickScript 에 사용된 태그명과 SuperTag를 표시합니다.
	애니메이션 링크에 사용된 태그를 보여줍니다.
	Application 스크립트에 사용된 태그를 보여줍니다.
	Application의 On Startup, While Running, On Shutdown 스크립트; Window On Show, While Showing, On Hide 스크립트, Key On Key Down, While Down, On Key Up 스크립트 에 사용된 것을 표시합니다. 아이콘을 더블클릭 하면 내용을 확인할 수 있습니다.
	Window Script 에서 사용된 태그와 SuperTag를 나타냅니다.
	Data Change Script 에 사용된 태그와 SuperTag를 나타냅니다.
	Condition Script 에 사용된 태그와 SuperTag를 나타냅니다. 예를 들어 \$Hour==12 On True. 더블 클릭하면 내용을 확인할 수 있습니다.

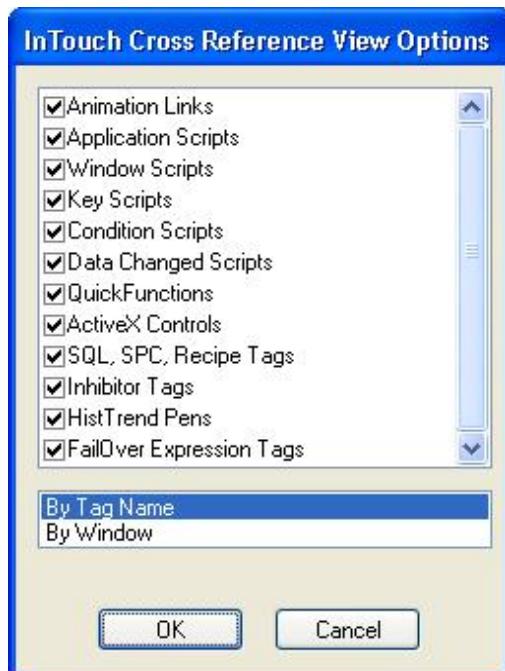
	Key Script 에 사용된 태그와 SuperTag를 나타냅니다. 예를 들어 F2 On Key Down. 더블 클릭하면 내용을 확인할 수 있습니다.
	QuickFunction 에 사용된 태그와 SuperTag를 나타냅니다.
	ActiveX Event Script 에 사용된 태그와 SuperTag를 나타냅니다.
	윈도우의 이름을 나타냅니다.
	SPC Pro Application 에 사용된 태그와 SuperTag를 나타냅니다.
	SQL Application 에 사용된 태그와 SuperTag 를 나타냅니다.
	Recipe Manager Application 에 사용된 태그와 SuperTag 를 나타냅니다.
	Alarm inhibitor 에 사용된 태그를 나타냅니다. 더블 클릭하면 inhibitor Tag 가 설정된 태그이름을 볼 수 있습니다.
	태그가 사용된 윈도우 이름을 나타냅니다.
	이력 트랜드에 사용된 태그를 나타냅니다.

Cross Reference 검색 조건 변경하기

처음 Cross Reference 검색을 실행한 후에 조건을 변경하여 다시 검색할 수 있습니다.

- InTouch Cross Reference 유ти리티 대화상자에서 Option 을 클릭합니다. (초기 검색 후에)

InTouch Cross Reference View Option 대화상자가 나타납니다.



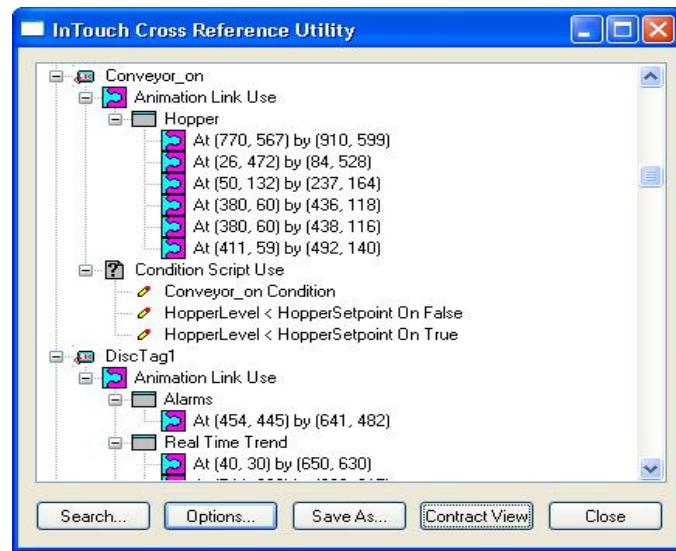
- b. 검색조건선택을 변경하기 위해, Cancel 을 클릭합니다.

InTouch Cross Reference 유ти리티 대화상자가 다시 나타납니다.

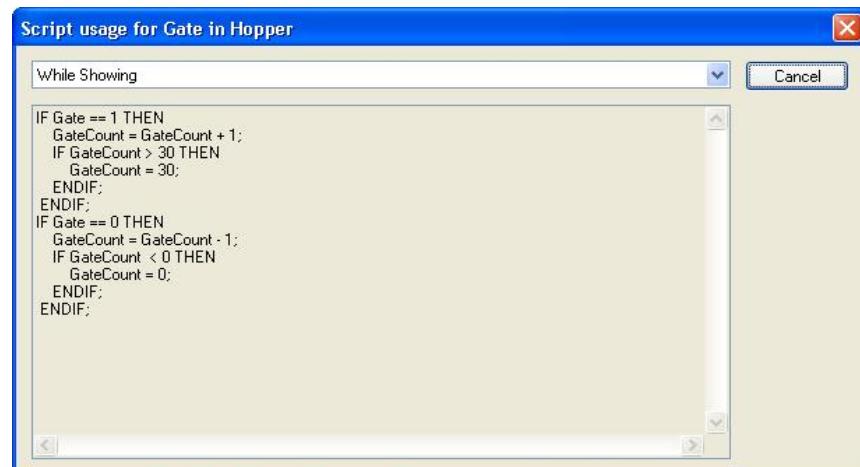
- c. 새로운 검색조건을 선택하고, Search를 클릭합니다.

- d. 대화상자의 버튼 목록에 TagName 또는 By Window를 선택합니다.

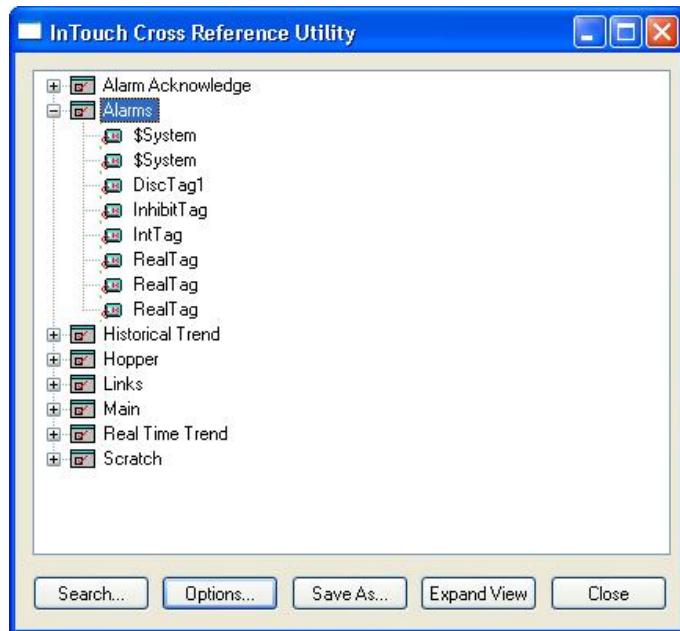
By Tagname : 검색된 내용을 찾을 때 Tagname에 의한 알파벳 순으로 검색합니다.



태그명에 대한 Script 사용은 더블 클릭하여 바로 볼 수 있습니다.



By Window Name : 검색 내용을 윈도우 이름순으로 정렬하여 보여 줍니다.



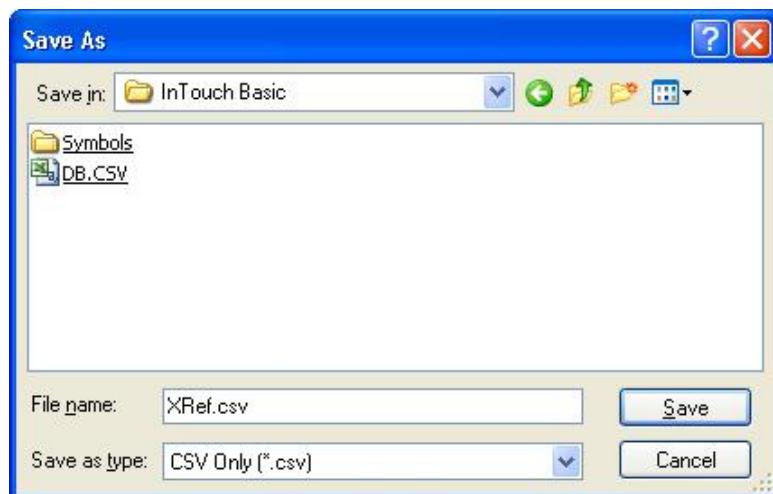
Cross Reference 파일 저장하기

Cross Reference 파일은 저장될 수 있으며, 콤마분리형 (.CSV) 파일 형식으로 제공되어 Text 편집 프로그램에서 열어 볼 수 있습니다.

Cross Reference 파일은 저장되는 정보는 유ти리티 창에 표시되고 있는 내용과 일치합니다.

- Cross Reference 유ти리티 대화상자에서 Save As를 클릭합니다.

Save As 대화상자가 나타납니다.

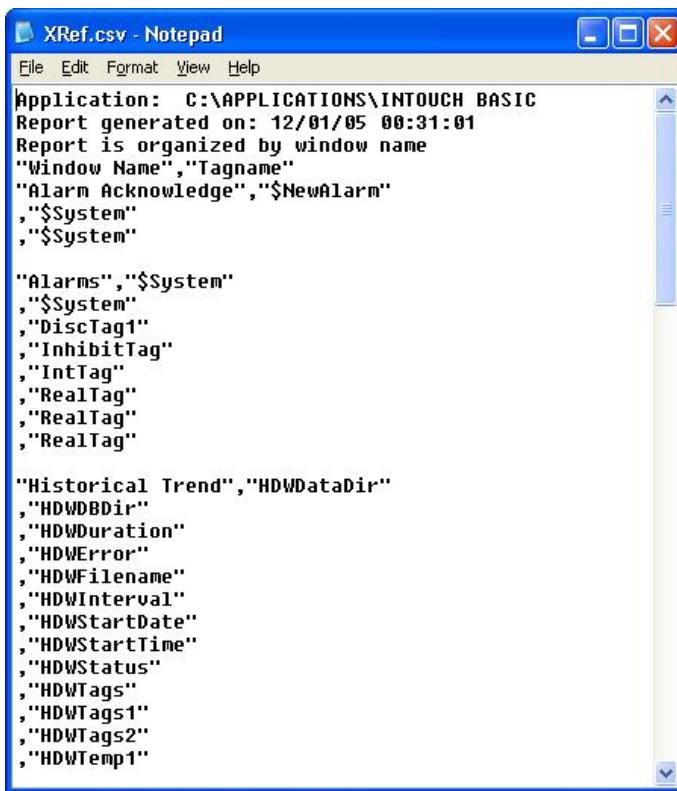


- b. File name 박스에 저장할 파일 이름을 입력합니다. 파일은 .CSV 파일 형식으로만 저장되어야 합니다.
- c. Save를 클릭합니다.

Cross Reference 파일 인쇄하기

Cross Reference 파일은 .CSV 파일 형식이 지원되는 Text 편집 프로그램에서 열려 인쇄 될 수 있습니다.

예를 들어, 메모장에 아래와 같이 파일을 열어봅니다.



The screenshot shows a Windows Notepad window titled "XRef.csv - Notepad". The window contains a large amount of text in a monospaced font. The text is a CSV report generated by the INTOUCH BASIC application. It includes header information like "Application: C:\APPLICATIONS\INTOUCH BASIC" and "Report generated on: 12/01/05 00:31:01". The main body of the text lists various tag names and their properties, such as "Window Name", "Tagname", "Alarm Acknowledge", "NewAlarm", and numerous entries for "Alarms", "System", and "RealTag" tags. There are also entries for historical trend data like "HDWDataDir", "HDWDDBdir", and "HDWDuration". The text continues with "HDWError", "HDWFilename", "HDWInterval", "HDWStartDate", "HDWStartTime", "HDWStatus", "HDWTags", "HDWTags1", "HDWTags2", and "HDWTemp1". The Notepad window has a standard Windows title bar with icons for minimize, maximize, and close, and a menu bar with File, Edit, Format, View, and Help.

File/ Print를 클릭하여 인쇄 합니다.

Updating Use Counts 와 사용되지 않은 태그 삭제하기

InTouch는 데이터 베이스에 각각의 item에 대해 Count를 사용하여 관리합니다. 이 Count는 애니메이션 링크나 Script의 변경, 윈도우의 삭제와 같은 내용에 대해 자동으로 업데이트 하지 않습니다. Application에서 Script나 애니메이션에 사용된 태그의 정보를 모두 삭제해도 Count가 업데이트되지 않으면, 그 태그는 삭제될 수 없습니다. 그러므로 태그를 삭제하기 위해서는 그 전에 Count를 업데이트 해야만 합니다.

Updating Use Counts

- a. WindowMaker에서 모든 윈도우를 닫습니다.
- b. Special/Update Use Counts를 클릭합니다.



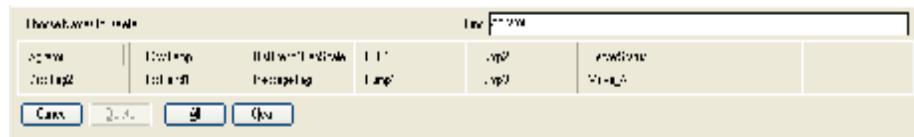
- c. Yes를 클릭합니다.
- d. 업데이트가 완료되면, 다음과 같은 장이 나타납니다.



- d. OK를 클릭합니다.

사용하지 않는 태그 삭제하기

- Special/ Delete Unused Tags를 클릭합니다.
Choose Names to Delete 대화상자가 나타납니다.

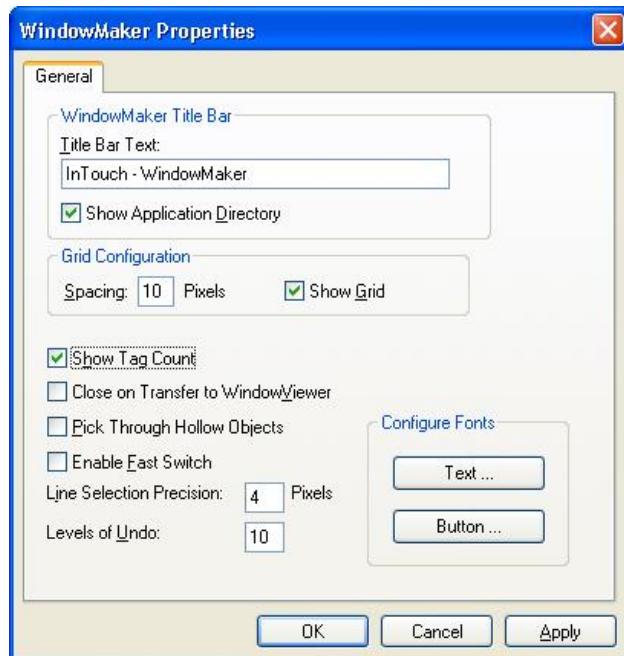


- 삭제할 태그를 선택하고, Delete를 클릭합니다. (Control Key를 사용하여 다중선택 가능)
- 모든 태그를 삭제하기 위해서는 All을 클릭합니다.

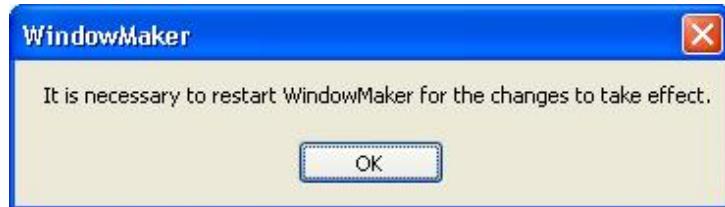
Tag Usage Count 표시하기

Tagname dictionary에 정의 된 로컬 태그의 수를 WindowMaker 메뉴 바에 표시할 수 있습니다. 태그 count는 내부 시스템 태그나 원격태그참조를 포함하지 않습니다. 또한 시스템 태그는 라이센스 태그 수에 포함되지 않습니다.

- Special/ Configure/ WindowMaker를 클릭합니다.
WindowMaker Properties 대화상자가 나타납니다.

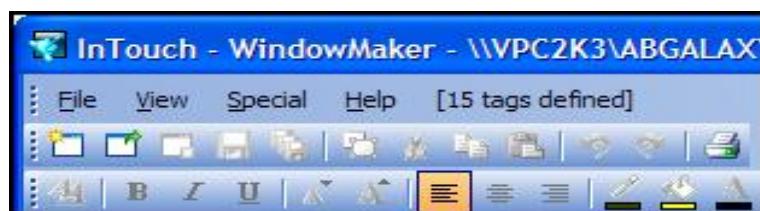


- Show Tag Count를 선택하고, OK를 클릭합니다.
다음과 같은 메시지 창이 나타납니다.



- c. OK를 클릭하고, WindowMaker를 재 시작합니다.

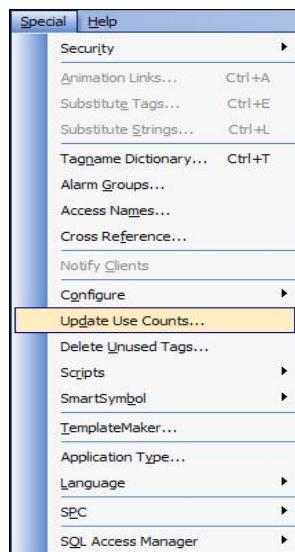
Tagname dictionary 정의 된 로컬 태그의 총 개수가 표시됩니다.



Lab 12 – 태그 삭제하기

Update Use Count

1. WindowMaker 의 모든 원도우를 닫습니다.
2. Special/ Update Use Counts를 선택합니다.



WindowMaker 는 Update 과정을 확인하기 위해 메시지 박스를 보여줍니다.



3. Yes를 클릭합니다.

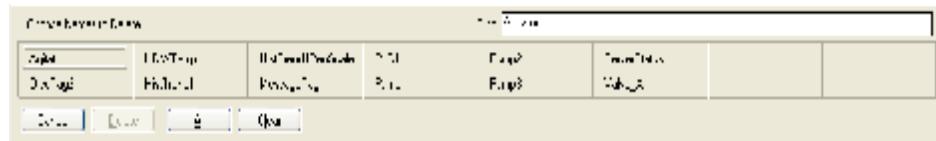


4. OK를 클릭합니다.

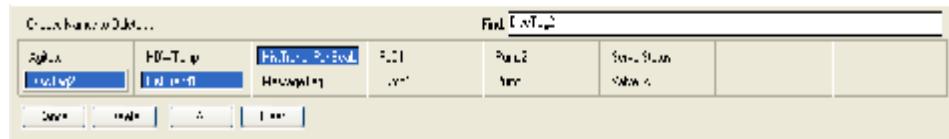
태그 삭제하기

5. WindowMaker 메뉴에서 Special/ Delete Unused Tags를 선택합니다.

Choose Names to Delete... 대화상자가 나타납니다.



6. 삭제할 태그를 선택합니다.



7. Delete를 클릭합니다.



8. OK를 클릭합니다.



9. OK를 클릭합니다.



Module 10

보안

Section 1 – Security Models

Lab 14 – Establishing Security

Section 2 – InTouch Security Features

Section 1 – Security Models

이 모듈에서는 InTouch 개발에서 가능한 3 가지 보안 모델의 개요를 설명합니다.

모든 InTouch 보안 방법은 Application에서 구성합니다. InTouch 개발자는 세가지 보안 모델을 선택할 수 있습니다.

ArchestrA Security : Industrial Application Server (IAS)을 위한 ArchestrA에 기반을 두는 보안으로써 NAD의 사용여부와 상관 없이 보안 체계를 중앙으로 집중시키게 됩니다.

IAS 보안에 관계된 실행은 외부의 Integrated Development Environment(IDE)에서 구성합니다. IDE에 관한 더 많은 정보는 Wonderware® ArchestrA Integrated Development Environment (IDE) Guide를 참조 하시기 바랍니다.

ArchestrA 인증과 권한부여를 선택하면 , Change Password, LogOn, Configure Users 또는 LogOff 등의 Spesial/Security 메뉴를 사용할 수 있게 됩니다.

InTouch Security : Application에 보안을 설정하므로, 운전자는 내부 태그에 함수들을 링크함으로써 실행을 허락한 특별한 Function들을 제어할 수 있습니다.

보안은 운전자가 Application에 이름과 암호를 넣고 로그온 하는 컨셉을 기본으로 합니다. 그러므로 각각의 운전자에 대한 사용자 이름, 암호, 권한 레벨을 구성해야 합니다.

Operating System Security : Operating System 인증모드는 사용자의 ID가 윈도우 도메인에 포함된 User name이 될 것 입니다. User name은 윈도우 네트워크 도메인/작업그룹에 포함된 사용자 목록으로부터 선택하실 수 있습니다. 각각의 User name은 주어지는 행동에 대한 사용자 인증을 결정하는 권한수준이 지정됩니다.

Operating system은 암호를 내부적으로 다루기 때문에, InTouch는 암호를 저장하지 않을 것입니다.

Operating System 보안은 사용자 목록과 그것에 대응하는 권한 레벨을 지정하기 위해 **Addpermission()** 함수를 사용합니다. Addpermission() 함수를 실행한 후에 생성되는 목록은 디스크에 쓰여집니다. 사용자의 세부적인 인증을 포함하는 이 파일은 NAD Client Node에는 복사되지 않습니다.

Lab 14 Establishing Security

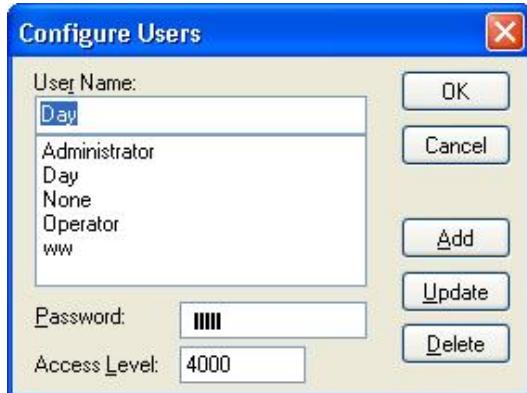
새로운 사용자 계정 생성

어플리케이션에 보안 설정을 위해, 어플리케이션에 보안유형 설정이 필요합니다.

1. 이전 과정을 실행하지 않았다면, 메인 WindowMaker로 부터 Special / Security / Select Security Type를 선택하고 InTouch를 선택합니다.
Administrator 와 암호를 wonderware 로 로그인합니다.
다음, 세개의 사용자 계정을 생성합니다.
2. Configure Users 대화상자를 보여주기 위해 Special / Security / Configure Users 명령을 클릭합니다.
3. 각 사용자 목록 밑에 계정을 추가하기 위해, User Name, password, 그리고 AccessLevel.을 입력합니다. 각 목록을 완료한 후 Add를 클릭합니다.

User Name	Password	Access Level
WW	WW	9999
Operator	HMI	8000
Day	Shift	4000

아래 그림과 같은 완료된 Configure Users 대화상자가 보입니다.



4. Ok를 클릭합니다.

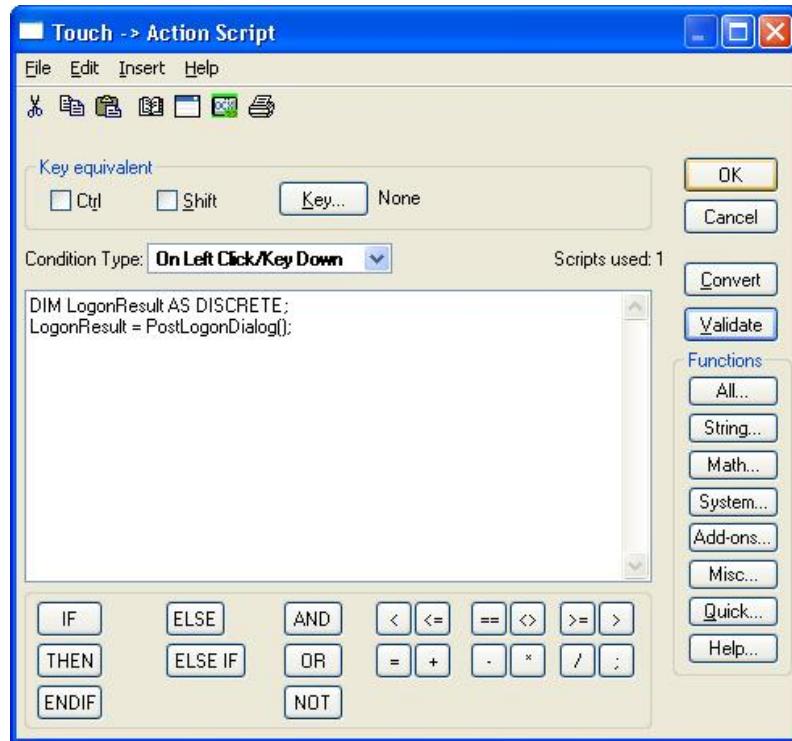
Logon Method 생성

5. 이전 **import**을 위해 사용된 **wwimport** 디렉토리로 부터 **Menu** 와 **Security** 윈도우를 **import**합니다.
6. **Security** 윈도우를 엽니다.

윈도우는 아래와 같은 그래픽 요소들을 포함합니다.



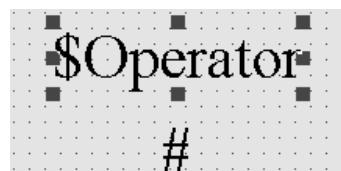
7. **Logon**이라는 버튼에 스크립트 함수 **PostLogDialog()**는 사용자가 **Application**에 로그인 할 수 있도록 하며 **DISCRETE** 값을 결과로 반환 합니다. 이번 예제에서는 반환된 값을 저장할 필요가 없으므로 로컬 로그인 **LogonResult**를 사용할 것입니다.
8. 다음 그래픽에서 보여진 것처럼 스크립트를 입력합니다.



9. Validate 버튼을 눌러 Script를 확인하고 OK를 클릭합니다.
10. Animation Links Selection 대화상자를 종료하기 위해 OK를 클릭합니다.

Configure the Operator Display Object

11. \$Operator display object에 Value Display String를 할당합니다.



12. 표현식 필드에 \$Operator를 입력합니다.



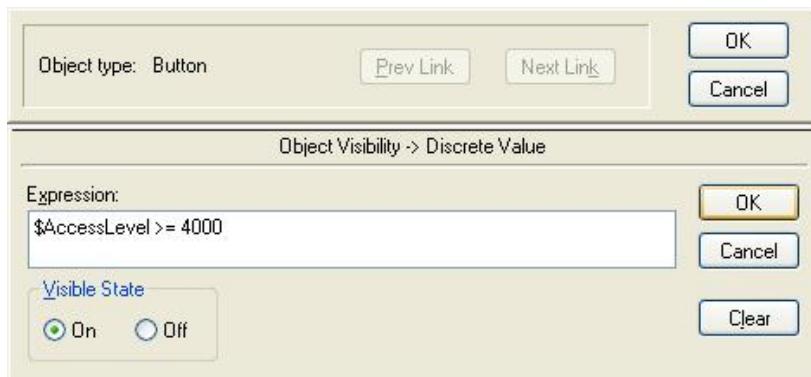
13. OK를 클릭합니다.

Configure the # Display Object

14. # 객체를 보여주기 위해 Value Display Analog 연결을 할당합니다.
 15. 표현식 필드에 \$AccessLevel를 입력합니다.
 16. OK를 클릭합니다.

Configure the Click to Continue button

17. Approved - Click to Continue라는 버튼에 Miscellaneous Visibility 연결을 할당 합니다.
 이 버튼은 시스템에 로그인 성공이 완료될때 보이게 될것입니다.
 18. Expression:에 \$AccessLevel에 맞는 값을 입력합니다.



19. OK를 클릭합니다.
 20. Menu 와 Links 윈도우를 보여주기 위해 Touch Link - Touch Pushbutton - Show Window 애니메이션 링크를 설정합니다.

Configure the Cancel Button

Cancel 버튼은 **logon** 과정을 취소할 때 사용하지만, 시스템에 로그인된 사용자가 없거나 **window** 가 처음 열려 질 때 보여집니다.

21. **Cancel** 버튼을 위해 **Miscellaneous Visibility** 연결을 설정합니다.
22. 다음과 같이 보여진 표현식을 입력합니다.



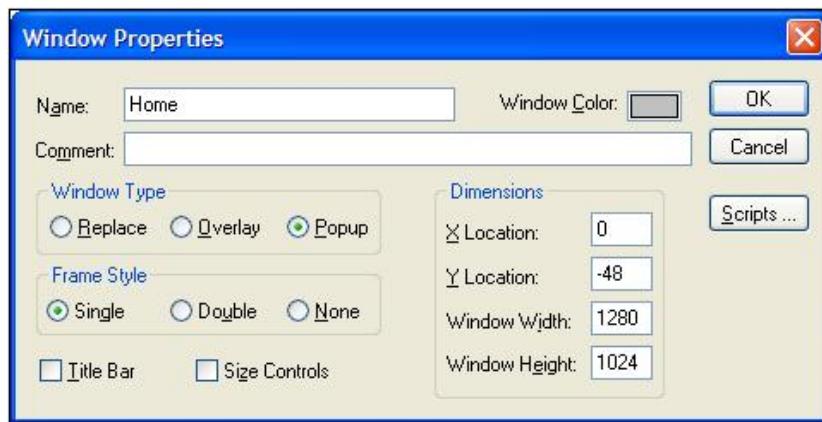
23. 아래의 **OK** 버튼을 클릭합니다.
24. 윈도우를 닫히지게 하기 위해 아래와 같이 **Touch Link - Touch Pushbuttons Action** 스크립트를 설정합니다.



25. 스크립트를 검사하고 **OK**를 클릭하고 **Animation Links Selection** 대화상자를 종료하기 위해 다시 **OK**를 클릭합니다.

Create a Home Window

26. Home이라는 새로운 윈도우를 생성하고 아래의 속성들을 할당합니다.

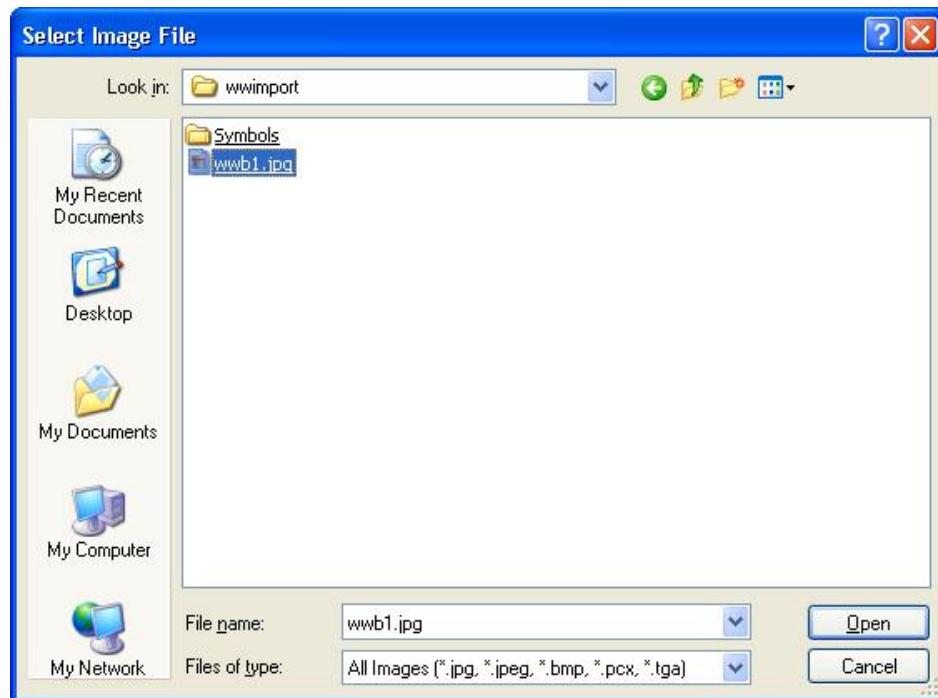


Note: 이전 예제는 **XP** 테마를 사용한 **windows 2003** 운영체제와 **1280 X 1024** 의 해상도를 사용 했습니다. 사용하는 운영체제는 해상도에 따라 값이 틀릴 수 도 있습니다.
Windows NT 에 정확하게 하려면 **Y 좌표**에 **-38** 을 입력합니다.

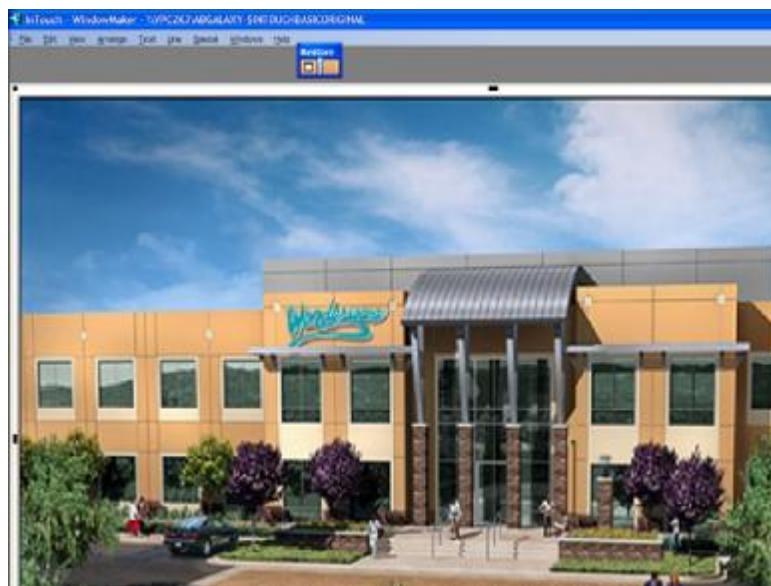
27. 윈도우에 비트맵 객체를 놓고 윈도우 전체를 덮기 위해 그것을 늘립니다.

28. 비트맵위에서 오른쪽 마우스를 클릭하고 Import Image 선택합니다.

29. c:\wwimport 디렉토리로 부터 wwb1.jpg 파일을 선택합니다.



30. 비트맵 객체에 이미지를 import하기 위해 OPEN을 클릭합니다.
31. 필요하다면 Home 원도우 전체를 채우기 위해 이미지를 크기를 재조정합니다.
아래 그림과 비슷하게 Home 원도우가 나타납니다.

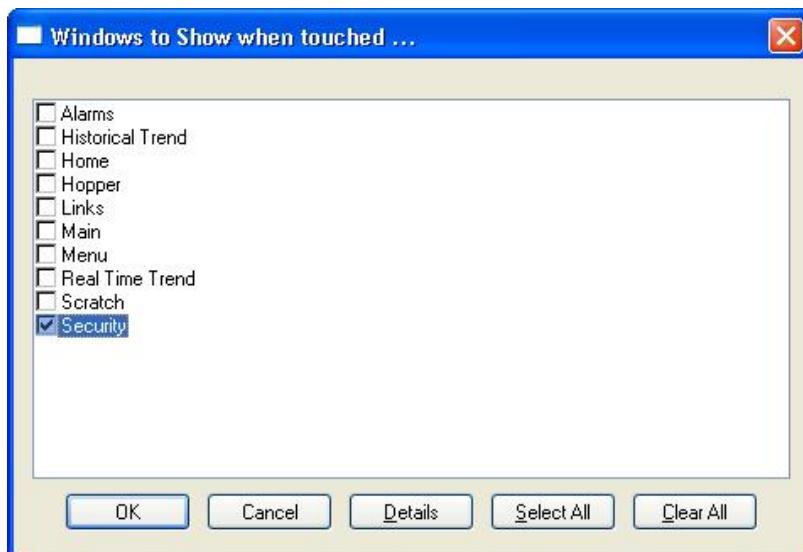


32. 빌딩 이미지의 정문위에 사각형 객체를 그립니다.



33. 사각형 객체를 더블 클릭합니다.

34. 사각형 객체에 아래의 Touch Pushbuttons - Show Window를 첨부합니다.



35. 사각형을 뒤로 보냅니다.

36. 저장하고 Home 윈도우를 닫습니다.

Configure Window Navigation

37. Menu 윈도우를 엽니다.

다음으로 Menu 상에 버튼에 연결이 필요한데, 각각의 윈도우에 Historical Trend 와 Real-time Trend 창을 연결하고, 스크립트를 설정 합니다..



38. Menu 윈도우의 내용들을 변환합니다.(F2, Special / Substitute Tags / Convert / Local).

Hint: F2 클릭전에 **Menu Window** 를 활성화 하려면 윈도우의 공백 영역을 클릭합니다.
이번 경우는 **Disk Space Free** 문자의 검은 지역을 클릭합니다..

39. 아래 테이블에 따라 애니메이션 링크를 설정합니다.

Button	Link Type	Window	Expression	Script
Home	Touch Pushbuttons - Action (On Left Click/KeyDown)			DIM LogOffResult as DISCRETE; LogOffResult=LogOff(); Show "Home";
Links	Touch Pushbuttons - Show Window	Links		
Alarms	Miscellaneous / Disable		\$AccessLevel < 5000 Disabled State = On	
	Touch Pushbuttons - Show Window	Alarms		
Trends	Touch Pushbuttons - Show Window	Historical Trend		
	Visibility		\$AccessLevel < 5000 Visible State = Off	
Hopper	Touch Pushbutton - Show Window	Hopper		
Main	Touch Pushbutton - Show Window	Main		

40. Menu 윈도우를 저장합니다.

41. Window Properties 대화상자에 Menu 윈도우 Y Location를 - 48로 변경합니다. 이것은 Menu 윈도우를 **WindowViewer program menu bars**위에 놓게 할 것 입니다.

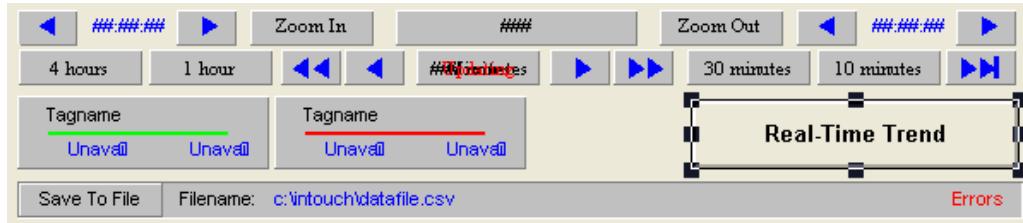
Note: 이전 예제는 XP 테마를 사용한 windows 2003 운영체제와 1280 X 1024 의 해상도를 사용 했습니다. 사용하는 운영체제는 해상도에 따라 값이 틀릴 수 도 있습니다.
Windows NT 에 정확하게 하려면 Y 좌표에 - 38 을 입력합니다..

Configure the Historical Trending Windows

42. Historical Trend윈도우를 엽니다.

43. Real-Time Trend(필요하다면) 버튼을 생성하고 Touch Pushbutton – Show Window

연결을 이용하여 Real-time Trend 윈도우에 연결합니다. 제시된 버튼 위치는 다음 그림과 같습니다.



44. 저장하고 윈도우를 닫습니다.

45. Real-Time Trend 윈도우를 열고 Historical Trend라고 명명한 버튼을 추가합니다.

46. Historical Trend윈도우에 Show Window링크를 설정합니다.

47. 저장하고 Real-Time Trend 윈도우를 닫습니다.

Selecting a Home Window

48. Special / Configure / WindowViewer를 선택합니다.

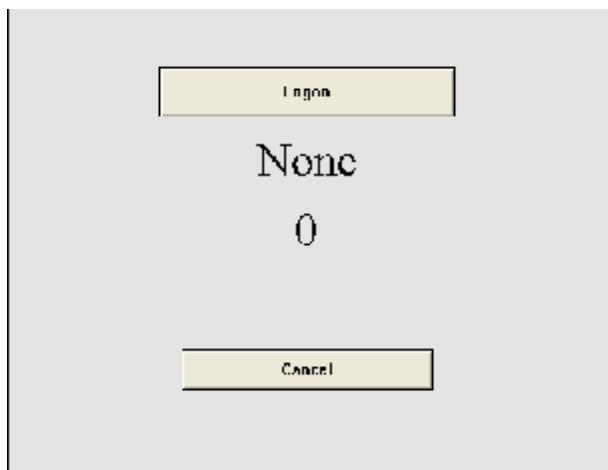
49. Home Windows탭을 선택하고 Home를 체크합니다.



50. OK를 클릭합니다.

로그인 / 암호가 보호된 어플리케이션 보기

51. Home 원도우를 열고 **Application**을 테스트하기 위해 Home 원도우를 열고, **Runtime**으로 전환합니다. WindowViewer에 Home 원도우가 보입니다.
52. 먼저 그래픽의 door를 클릭하여 Logon window를 나타나게 합니다.
53. Logon를 클릭하고 사용자 **ww** 암호 **ww**로 로그인합니다. 로그인 화면이 login name과 access level을 보여줄 것입니다.
54. Approved - Click to Continue 버튼을 클릭합니다. **Link**와 **Menu** 원도우는 모두 로그인되어야 보여지게 됩니다.
Logon 화면이 나타납니다.



55. Logon를 클릭합니다.

56. 사용자 ww 암호 ww로 로그인합니다.

로그인 후 메인 화면에는 로그인 이름과 접근 현재 로그인된 로그인명과 액세스 수준이 나타나게 됩니다.



57. Approved - Click to continue 버튼을 클릭합니다.

Links 와 Menu 윈도우가 보여 질것입니다. (모든 사용자에 대해서는 아래에 자세히 설명하였습니다.)

58. 생성된 사용자의 각 로그인:

User Name	Password	Access Level	Security Features
WW	WW	9999	Has access to all windows/links.
Operator	HMI	8000	Has access to all windows/links.
Day	Shift	4000	Links , Hopper , and Main windows. Alarms button is disabled, and Trend link is not visible.

Note: Application 을 변경 할 경우 InTouch Application Manager 를 통해 반드시
삭제하고 다시 배포 하여야 합니다.

Section 2 – InTouch Security Features

InTouch Security

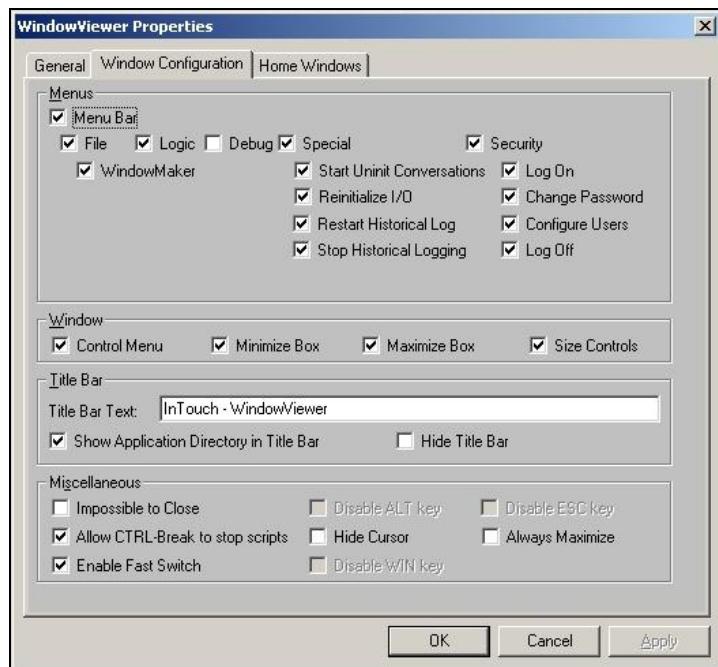
InTouch Application 을 보안한다는 것은 개인마다 다른 설정을 주는 것을 의미합니다.

Application 개발자는 Operating 시스템을 보안하거나 Operator 의 행동을 감시하거나 또는 Application 자체를 보안하기를 원합니다. 이런 다른 요구들을 적용 하기 위해서, InTouch 보안 체계는 여러가지 Application 에 대한 다양한 옵션을 포함하고 있습니다.

첫번째 InTouch 보안은 로컬 노드의 Operating System 이나 Application 윈도우에 Operator 의 접근을 제한합니다.

스크립트 Function 은 <ALT> or <TAB> 와 같은 키의 사용을 못하게 할 수 있습니다.

또한, Special/Configure/WindowViewer 에서 사용자가 윈도우를 닫을 수 없게 메뉴를 숨기거나 항상 최대화시켜 놓는 방법이 있습니다.



두번째로 InTouch 는 사용자의 행동을 감시합니다.. (사용자가 언제 무엇을 했는지)

InTouch 이벤트나 inactivity 시스템 태그가 이 행동들을 추적할 수 있습니다.

세번째로 보안은 현재의 사용자나 그 사용자가 속해있는 그룹에 대해 설정되어있는 권리에 기반하여 Application 의 기능을 제한하는 것을 의미합니다. 이것은 기능적으로 제한된 접근으로 객체에 visibility 나 disable 를 설정 함으로써 구현할 수 있습니다.

InTouch Application Security

보안은 운전자가 Application에 이름과 암호를 넣고 로그온 하는 컨셉을 기본으로 합니다.

그러므로 사용자 이름, 암호 그리고 각 운전자별 권한 수준(Access Level)이 필요합니다.

새로운 Application이 만들어 졌을 때, 기본적으로, 사용자 이름이 권한 수준을 **9999** (모든 보안 명령에 접근이 가능 합니다)을 갖는 “Administrator”로 정해집니다. 한번 보안 목록에 새로운 사용자 이름이 만들어 지고, Windowmaker 또는 WindowViewer가 다시 시작되면, 기본 사용자 이름은 자동으로 권한 수준을 “0”(WindowMaker 그리고 WindowViewer에 Configure User 명령에 억세스 할 수 없는)을 갖는 “None”으로 재 설정 됩니다. 그러므로, Security user list에 접근 할 수 있기 위해서는, 9000보다 크거나 같은 권한 수준을 갖는 System Administrator를 위한 구성을 해야 합니다.

Using the Security Internal Tagnames

한번 보안이 설정이 되면, 버튼에서 사용되고, 애니메이션 링크의 식에서 또는 QuickScript 등에서 로그 온 한 운전자가 특별한 기능들을 사용할 수 있도록 허락 할 것인지를 컨트롤 하기 위 사용될 수 있는 두개의 내부 보안 태그명이 있습니다.

Tagname	Type Valid	Values	Access
\$AccessLevel	Integer	0-9999	Read Only
\$Operator	Message	16-characters max	Read Only

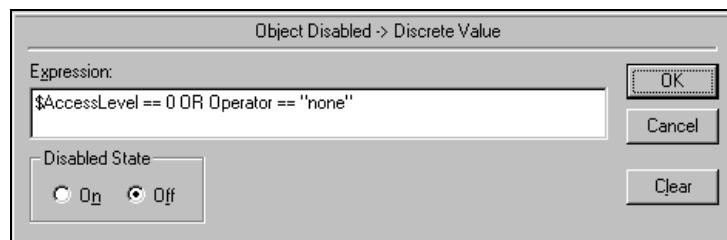
예를 들면, 로그 온 한 운전자의 권한 수준에 따라서 어떠한 객체를 보이도록 하기 위해서, 다음과 같은 문장이 visibility 애니메이션 링크의 식에 사용이 됩니다.

`$AccessLevel >= 2000`

또는, QuickScript은 IF 문장에 의해서 사용 됩니다.

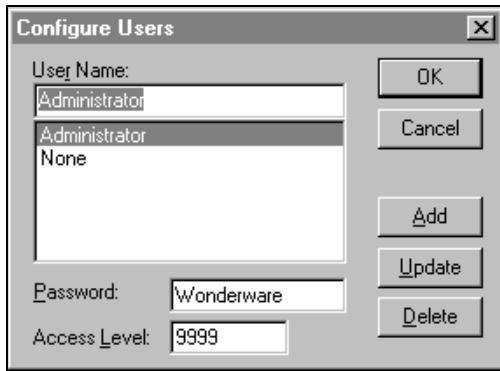
```
IF $Operator == "DayShift" THEN
    Show "Control Panel Window";
ENDIF;
```

또한 **Disable** 애니메이션 링크를 사용해서 내부 보안 태그명의 값에 근거한 객체의 터치 기능들을 조절 할 수 있습니다.



Configuring the Operator's Security Level

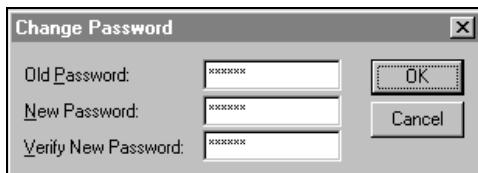
1. **Special/Security/Configure Users** 명령을 수행 합니다. **Configure Users** 대화 상자가 나타납니다.



2. **User Name** 상자에, 운전자로 정의할 이름을 넣습니다.
3. **Password** 상자에, 패스워드를 넣습니다.(32자까지 가능).
4. **Access Level** 상자에, 값을 넣습니다(lowest = 0 to highest = 9999).
5. **Add**를 클릭 합니다.
6. 존재하는 사용자 이름을 수정하기 위해서, **User Name** 목록에서 원하는 이름을 선택 합니다. 각각의 변화된 목록을 넣고 **Update**를 클릭 합니다.

보안접속 암호 변경

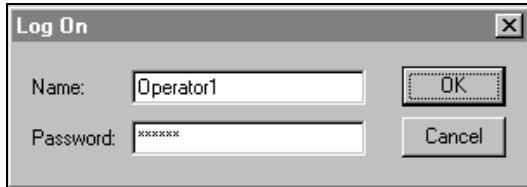
1. **Special/Security/Change Password** 명령을 클릭 하면 **Change Password** 대화 상자가 나타납니다:



2. **Old Password** 상자에, 이전 패스워드를 넣습니다.
3. **New Password** 상자에, 새로운 패스워드(up to 32 characters)를 넣습니다.
4. **Verify Password** 상자에, 새로운 패스워드를 다시 넣습니다.
5. **OK**를 클릭 합니다.

어플리케이션에 접속

1. Special/Security/Log On 명령을 클릭 하면, Log On 대화 상자가 나타납니다.



2. Name 상자 안에, 사용자 이름을 넣습니다
3. Password 상자안에, 패스워드를 넣습니다. 그리고 OK를 선택 합니다.

Creating a Custom Security Log On Window

만약 WindowViewer에서 Special 메뉴가 디스플레이 되지 않는다면, 운전자가 Application에서 로그 온 할 수 있는 로그 온 윈도우가 필요 합니다.

\$OperatorEntered 그리고 **\$PasswordEntered** 시스템 태그명을 사용자 입력 객체들에 연결하거나, QuickScript 내에서 “User Name” 그리고 “Password”를 설정 하기 위해서 사용 합니다.

예를 들면

```
Set the User Name string into      -> $OperatorEntered
Set the User Password string into -> $PasswordEntered
```

로그 온 이 허용 되면, **\$AccessLevel** 그리고 **\$Operator** 내부 태그명이 미리 설정 되어 있던 값으로 설정 됩니다.

또한, Special 메뉴가 WindowViewer에서 디스플레이가 되지 않을 때, **User Input-Discrete** 버튼은 **Change Password** 대화 상자를 보여주고, 운전자에게 패스워드를 변경 할 수 있도록 하기 위해서 **\$ChangePassword**에 연결 되어 질 수 있습니다.

운전자가 버튼을 클릭 했을 때, **\$ChangePassword** 태그명의 값은 1로 설정 되고, **Change Password** 대화 상자가 나타납니다. 운전자가 대화 상자를 닫을 때, 시스템을 값을 0으로 재설정 합니다.

User Input - Discrete 버튼에 승인된 운전자가 **Configure User** 대화 상자에 접근하고 security user name list 를 수정할 수 있는 **\$ConfigureUsers** 태그명과 연결 할 수 있습니다. 운전자가 그 버튼을 클릭 했을 때, **\$ConfigureUsers** 태그명의 값이 1로 설정이 됩니다. 그리고, **Configure Users** 대화 상자가 나타납니다. 운전자가 그 대화 상자를 닫으면 시스템은 값을 0으로 재설정 합니다.

관련된 태그들

\$ConfigureUsers

이 태그에 1 을 주면 WindowMaker Special/Security/Configure Users... 와 같이 Configure Users 대화상자를 보여줍니다.. 단, 권한수준이 9000 이상인 사용자로 로그인 되어 있을 때만 가능하다. Configure Users 대화상자를 닫으면 자동적으로 이 태그는 0 이 됩니다.

Data Type Discrete (write only)

\$ChangePassword

이 태그에 1 을 주면 WindowMaker Special/Security/Change Password...와 같이 Change Password 대화상자를 보여줍니다. Change Password 대화상자를 닫으면 자동적으로 이 태그는 0 이 됩니다.

Data Type Discrete (write only)

\$Operator

현재 로그온 된 사용자의 이름을 갖고 있습니다.

Data Type Message (read only)

\$AccessLevel

현재 로그온 된 사용자의 권한수준을 갖고 있습니다.

Valid Values 0 ~ 9999

Data Type Integer (read only)

\$OperatorEntered

로그온 할 때 사용자 이름을 넣는데 쓰입니다.

이 태그와 \$PasswordEntered 태그의 값이 맞으면 \$AccessLevel 태그와 \$Operator 태그가 정의된 값을 갖게 됩니다.

Data Type Message (read / write)

\$PasswordEntered

로그온 할 때 암호를 넣는데 쓰인다. 이 태그는 빈 문자열로 출력이 되기 때문에 값을 볼 수가 없습니다. 그러므로 이 태그로 Data Change 스크립트를 쓰면 결코 trigger 가 일어나지 않아 소용이 없을 것입니다. 이 태그와 \$OperatorEntered 태그의 값이 맞으면 \$AccessLevel 태그와 \$Operator 태그가 정의된 값을 갖게 됩니다.

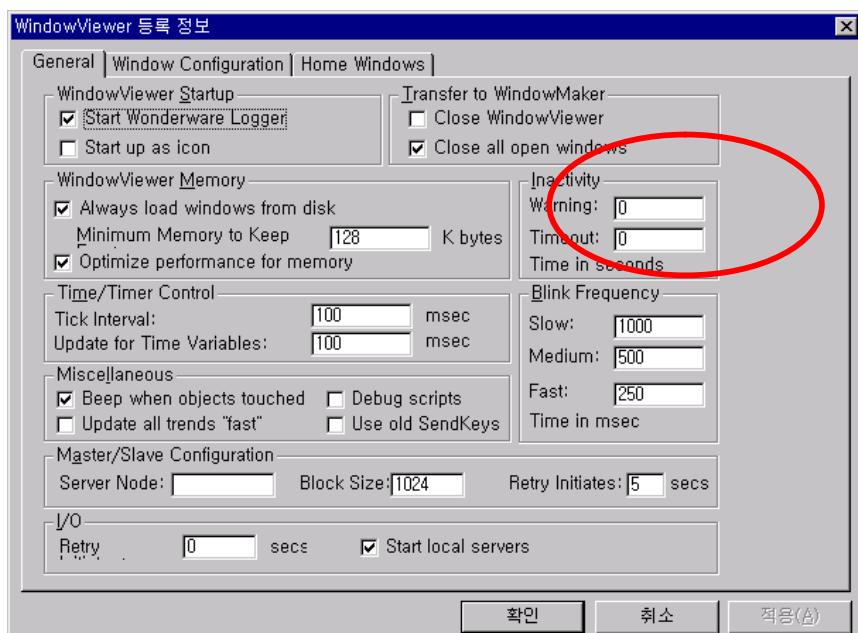
Data Type Message (read / write)

\$InactivityWarning

Special/Configure.../WindowsViewer... 메뉴 명령으로 아래의 WindowsViewer Configuration

대화상자에서 General 탭에 Inactivity 섹션의 Warning: 상자에 입력된 시간(초)이 로그온 후 경과된 시간과 맞으면 이 태그는 1을 유지합니다.

Data Type Discrete (read only)



\$InactivityTimeout

Special/Configure.../WindowsViewer... 메뉴 명령으로 위의 WindowsViewer Configuration

대화상자에서 General 탭에 Inactivity 섹션의 Timeout: 상자에 입력된 시간(초)이 로그온 후 경과된 시간과 맞으면 이 태그는 1을 유지하고 로그오프 됩니다. (자동 로그오프)

Data Type Discrete (read only)

Note: 로그오프 되면 사용자 이름은 “None”가 되고 권한수준은 0이 됩니다.

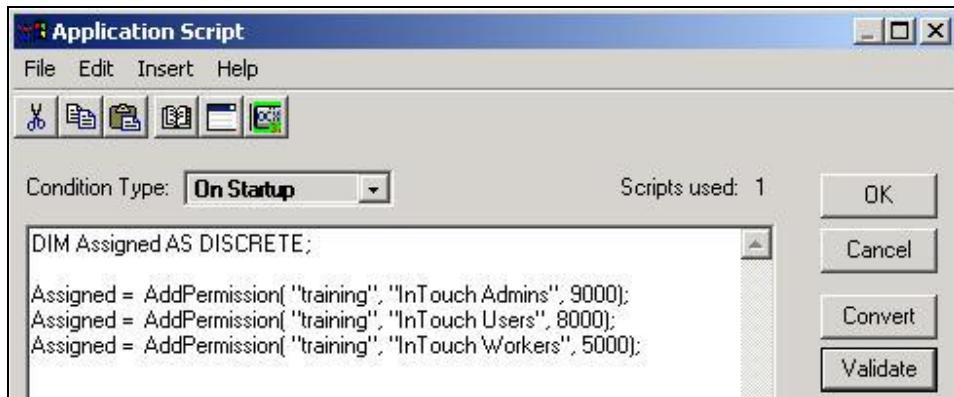
Operating System Security

- WindowViewer 가 실행중 이라면 정시시킵니다. **Special/Security>Select Security**

Type에서 **OS Security**를 선택합니다.



- Application/On Startup script에서 **AccessLevels**를 지정합니다.



Log On & Log Off 구성

- Window 를 생성하고 아래의 객체들을 그립니다.



- Log On 버튼을 더블 클릭합니다. Action 스크립트에 다음 문장을 넣습니다.



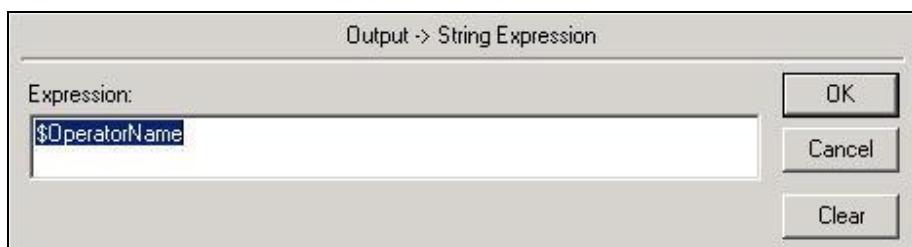
- Log Off 버튼을 더블 클릭합니다. Action 스크립트에 다음의 문장을 넣습니다.



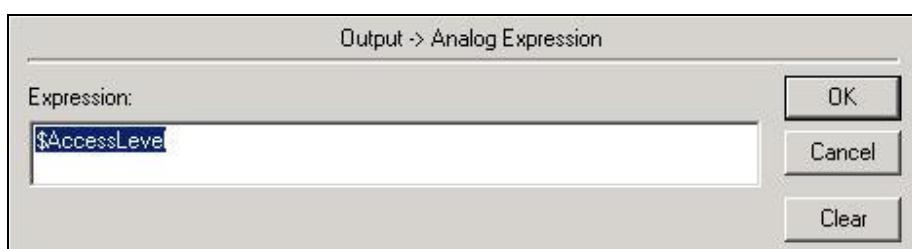
- \$Operator 다음의 문자 XXXXX를 더블클릭하고, Value Display/String 애니메이션 링크에 시스템 태그 \$Operator를 넣습니다.



- \$OperatorName 다음의 문자 XXXXXX를 더블클릭하고 Value Display/String 애니메이션링크에 시스템 태그 \$OperatorName을 넣습니다.



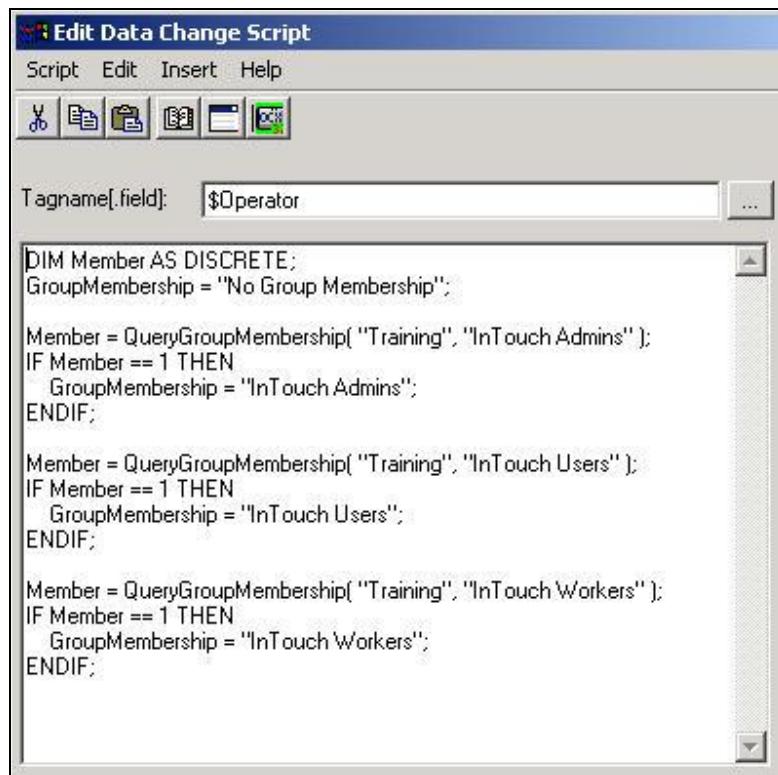
- \$AccessLevel 다음의 문자 #####를 더블 클릭하고, Value Display/Analog 애니메이션링크에 태그 \$AccessLevel를 넣습니다.



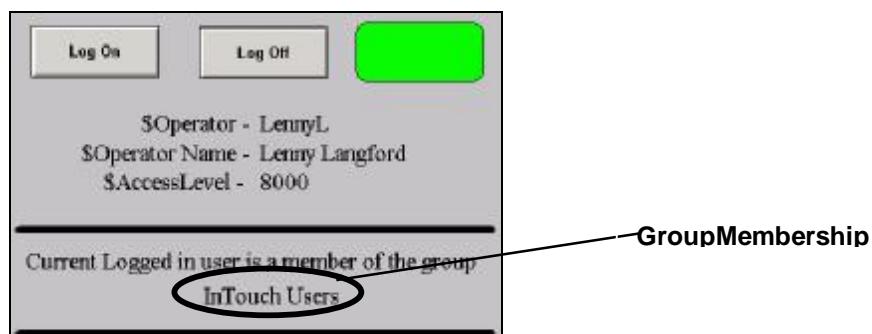
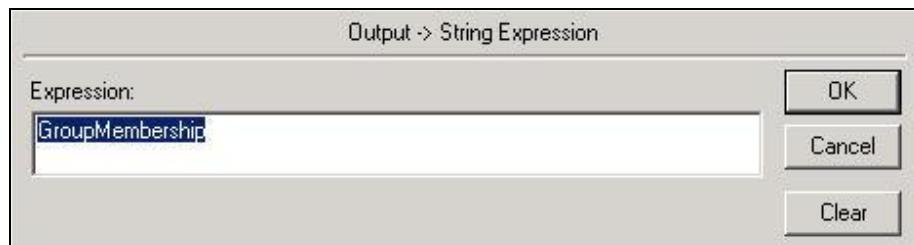
그룹 확인하기

보안의 목적을 위해 그룹 멤버를 확인할 필요가 있습니다.

- MemoryMessage 태그 (예: GroupMembership)를 생성합니다.
- Data Change 스크립트에 그룹 멤버를 확인하기 위해 \$Operator 태그를 넣습니다.



- 문자열 #####의 Value Display/String 애니메이션 링크에 GroupMembership 태그를 넣습니다.



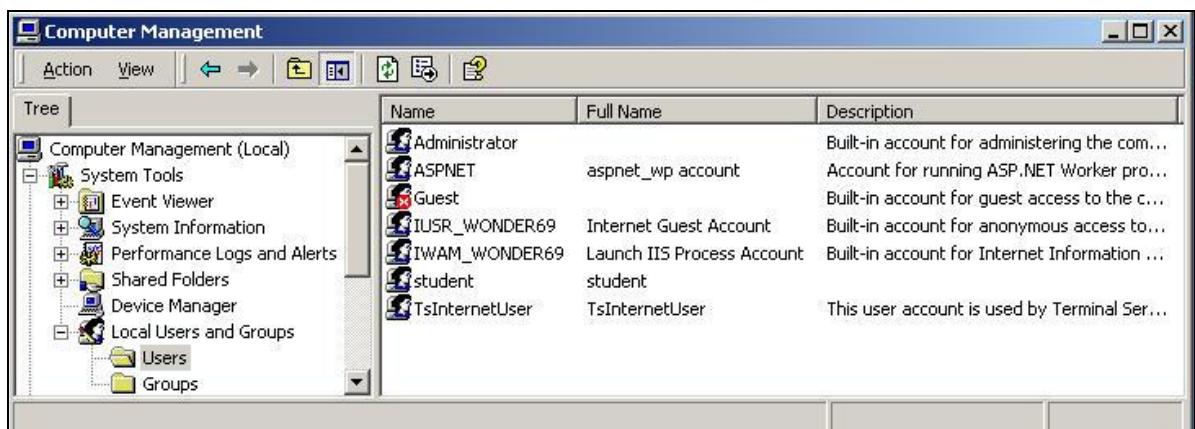
로컬 사용자 만들기

InTouch에서 OS 기본 보안은 도메인 사용자나 그룹 또는 로컬에 있는 사용자나 그룹을 모두 사용할 수 있습니다.

로컬 사용자 생성

- 윈도우 시작 메뉴에서 시작/설정/제어판/관리도구/컴퓨터 관리를 선택합니다.
- 시스템 도구/로컬 사용자 및 그룹을 확장합니다.
- 사용자를 선택합니다.

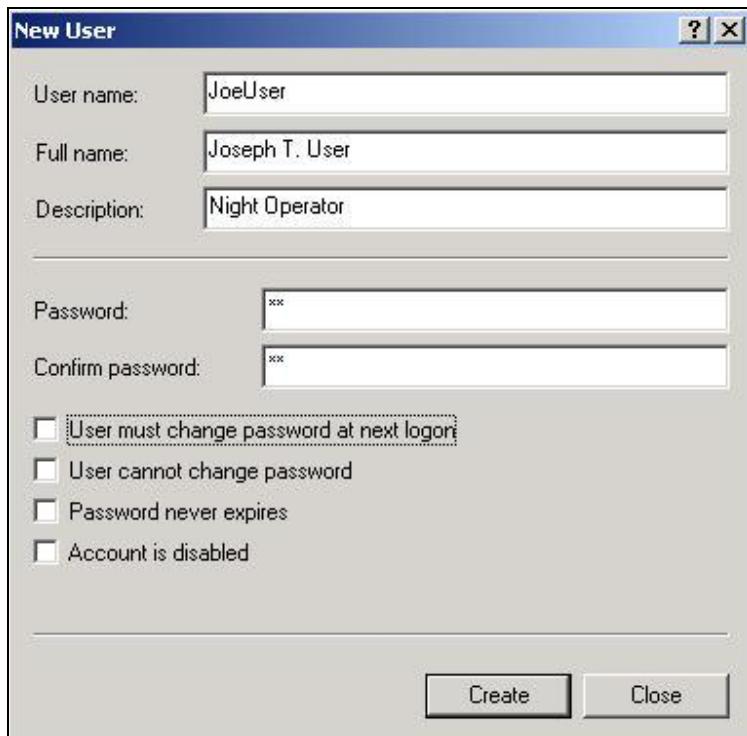
Note : Window XP에서는 약간 다를 것입니다.



- 사용자에 오른쪽 버튼을 클릭하고 새 사용자를 선택합니다.



- 새 사용자 대화상자가 나타납니다.
- 새 사용자에 대한 정보를 넣습니다 (사용자 이름, 전체이름, 설명, 암호) 만들기를 클릭합니다.



- '다음 로그온시 반드시 암호 변경(N)' 체크박스를 선택하지 않습니다.

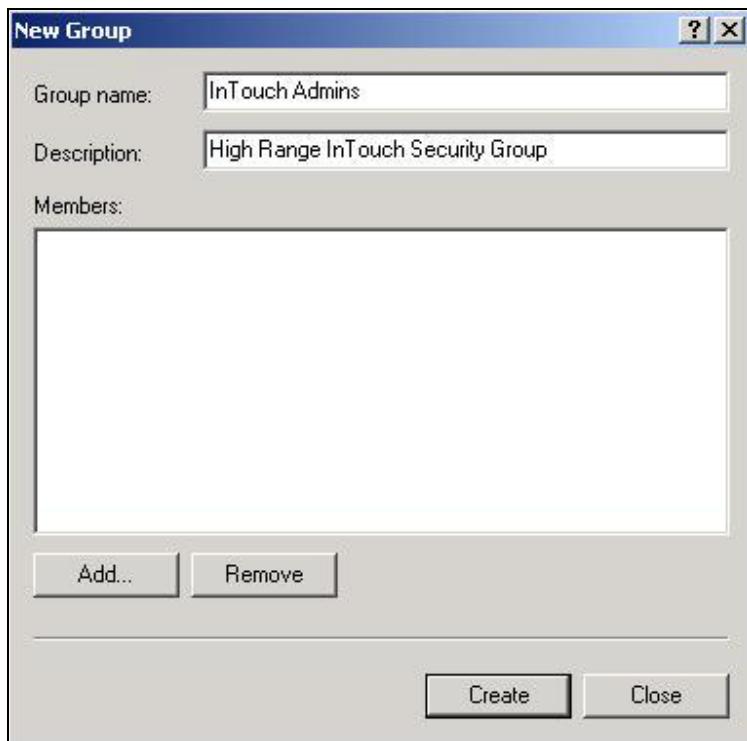
새로운 로컬 그룹만들기 & 사용자 지정하기

-로컬 사용자 및 그룹에서 그룹을 오른쪽 마우스 클릭합니다.

새 그룹을 선택합니다.



-그룹 이름에 InTouchAdmins 라고 넣습니다.



- 그룹을 선택하고 오른쪽 마우스 클릭을 합니다.

그룹에 추가를 선택합니다.,

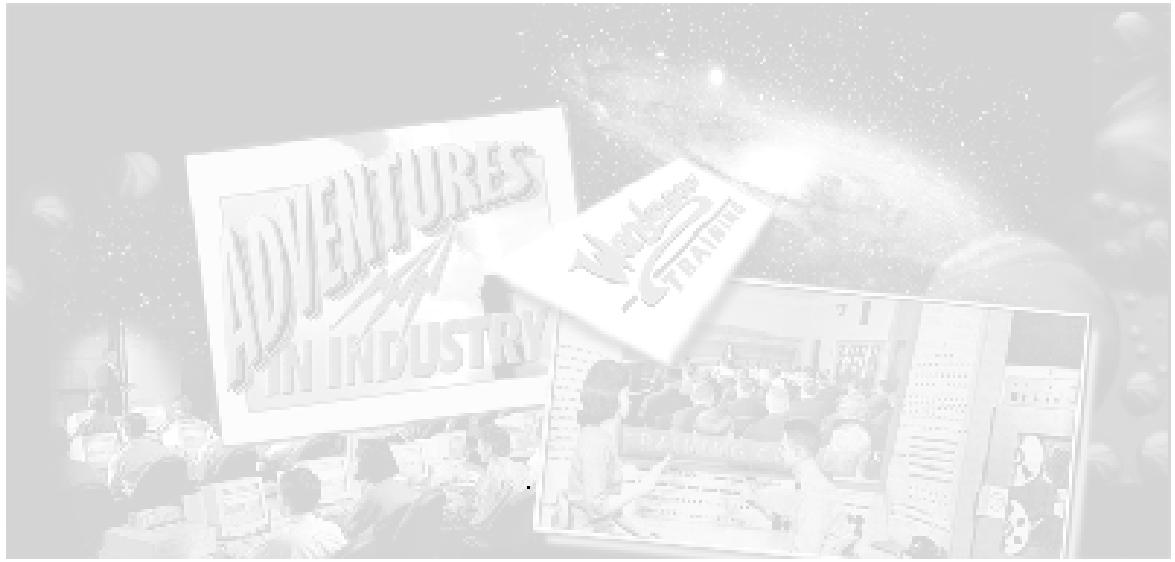


-추가를 클릭합니다.



- 사용자를 선택하고, 추가를 클릭합니다.





Module 11

Application Backup

Section 1 – Application Backup

Section 1 – Application Backup

이 장에서는 분산 개발 환경에서 Application을 어떻게 저장하고, 활용하는지에 대해 살펴봅니다.

소개

Application 을 수정하거나, 추가작업을 하기 전에, 반드시 Application 파일을 백업해 두어야 합니다.

InTouch Application 을 백업할 때는, WindowViewer 의 실행을 중지 시켜야 하며, WindowMaker 또한 닫고 백업 하기를 권장합니다.

백업된 파일들은 Application 개발에 관한 이력을 생성할 수 있고, 예상치 못한 문제가 발생하는 경우 작업을 되돌릴 수 있습니다.

백업작업은 Application을 수정하기 전이나, 다른 노드에 배포하기 전에 수행되어야 합니다.

InTouch File Content

Application 윈도우가 생성될 때, 정의된 윈도우가 WindowMaker와 WindowViewer 에 디스플레이 되기 위해서 필요로 되는 정보를 가진 멀티 파일들이 생성됩니다.

이것은 각각의 윈도우에 하나이상의 파일을 갖고 있다는 것입니다.

WindowMaker 에서 윈도우가 생성될 때, 생성되는 파일을 WIN****.WIN 이라 하고 이것은 윈도우 데이터를 가지며 생성됩니다.

WindowMaker 에서 윈도우를 수정하게 되면, 새로운 버전의 .WIN 파일이 생성되고 이전 파일은 .WBK 파일로 변환됩니다.

Application을 실행시키면 (Runtime) , WindowViewer 는 WIN 파일을 컴파일 하게 됩니다. 이것은 .WVW 확장자를 갖습니다.

마지막으로, 윈도우에 하나이상의 real-time Trend 가 있다면, .TRN이라는 파일을 생성하고 .WIN 파일과 함께 생성됩니다.

사용자가 Application을 다른 PC에 옮기려 한다면, 전체 Application 디렉토리를 복사해 넣습니다.

Checking Out and Checking In an IDE-Managed Application

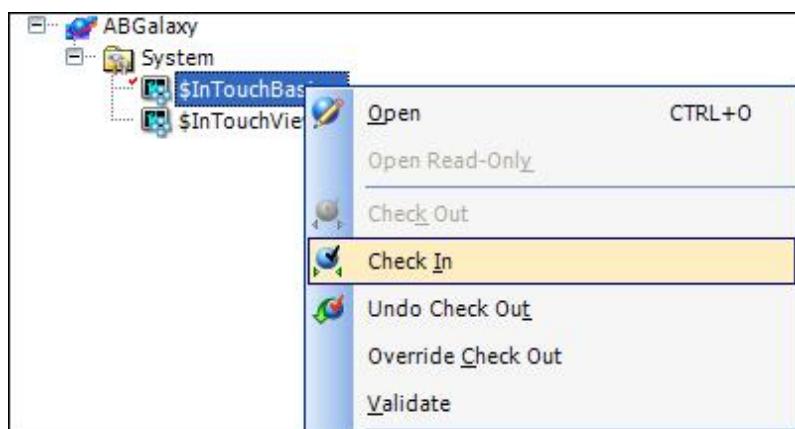
개발자가 IDE – Managed Application 을 열 때마다, Application 은 자동으로 Checked out 됩니다. 이것은 Application 에 Lock 이 걸린다는 의미이며, 다른 개발자가 Application에 접속하기 위해 Application 을 열면 Read-Only로 열리게 됩니다.

Application 작업이 끝나서 저장하고 닫으면, ArchesrA IDE 는 자동으로 Check-in 상태로 되돌아 가게 됩니다.

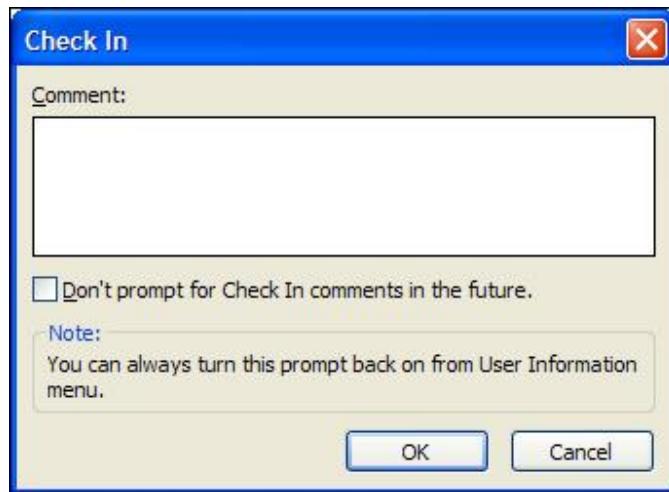


Note: InTouch Application 은 WindowMaker 가 실행될 때 Checked out 됩니다. 만약 아무 수정없이 InTouch 를 종료하면, Check-in 대화상자가 나타나지 않습니다.

- a. ArchesrA IDE Template Toolbox의 Application 객체에서 오른쪽마우스 클릭을 하고, Check in option 을 선택합니다.

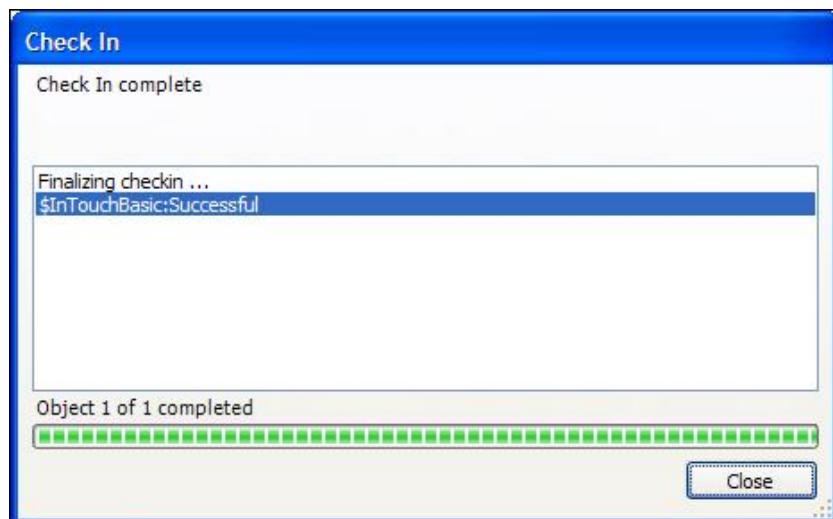


Check In 대화상자가 나타나면 Application의 변경사항에 대해 Comments부분에 설명을 입력할 수 있습니다.



b. OK를 클릭합니다.

잠시 후, Check In이 완료된 메시지를 볼 수 있습니다.



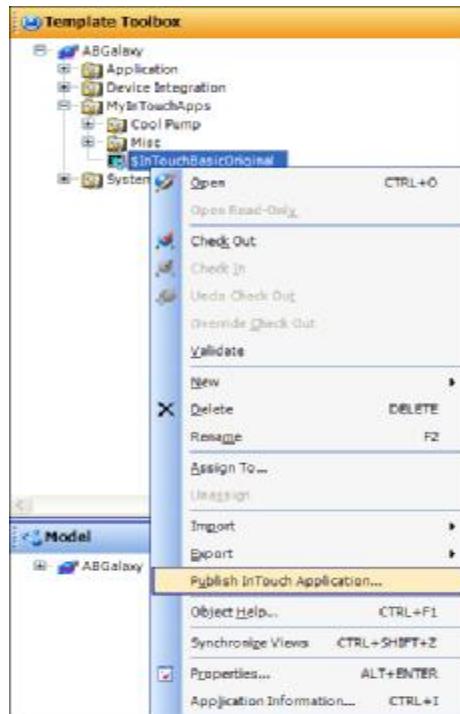
c. Close를 클릭합니다.



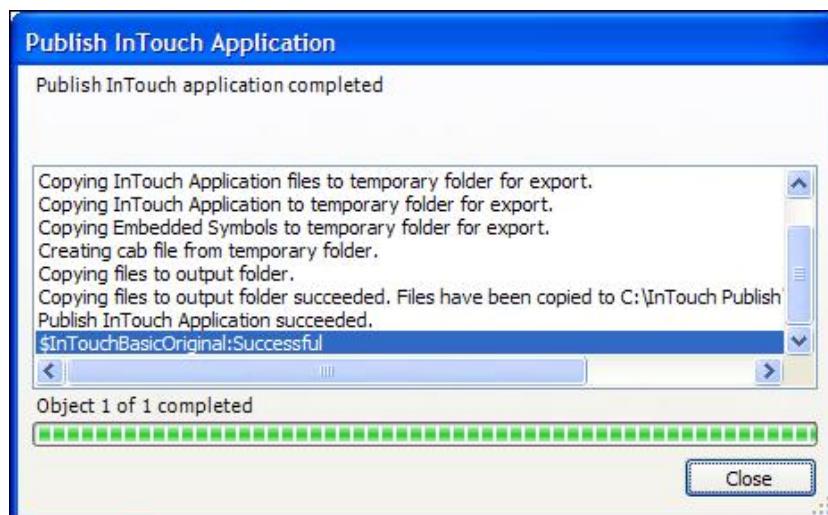
Application에 Checked out 표시가 사라진걸 확인할 수 있습니다.

Publish Your IDE-Managed InTouch Application

- a. Application 객체에서 오른쪽 마우스 클릭 후 Publish InTouch Application을 선택합니다.



- b. Browse for Folder 대화상자에서 Application folder를 선택하고 OK를 클릭합니다.
- c. 잠시 후 Publish InTouch Application 상자 안에서 Publish 과정을 성공했다는 메시지를 확인할 수 있습니다.

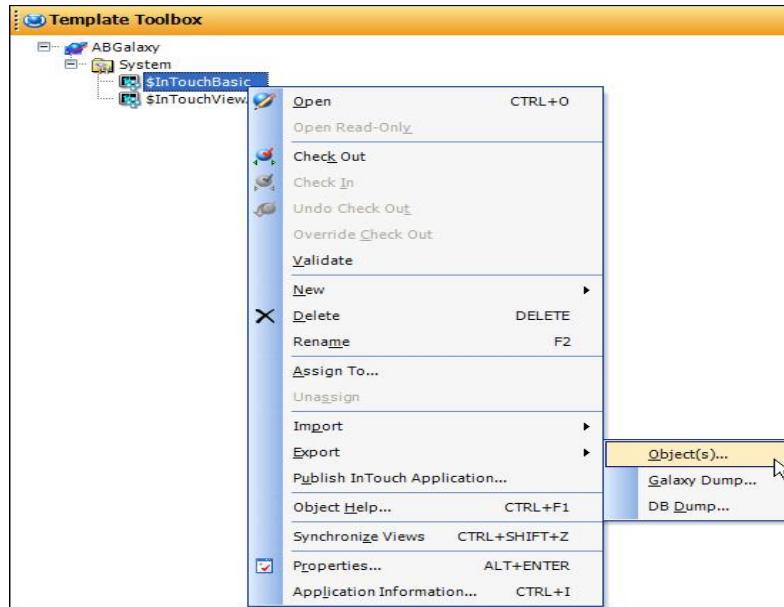


d. ArchestrA IDE를 최소화하거나, 종료합니다.

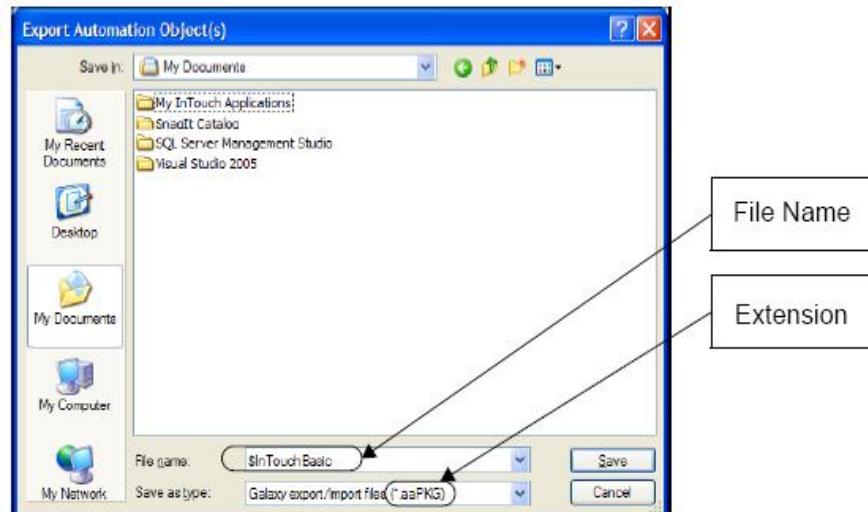
IDE-Managed InTouch Application 백업하기

IDE-Managed InTouch Application 은 Galaxy로부터 백업 받을 수 있습니다.

a. 백업 받으려고 하는 Application 객체에서 오른쪽 마우스 클릭을 합니다.

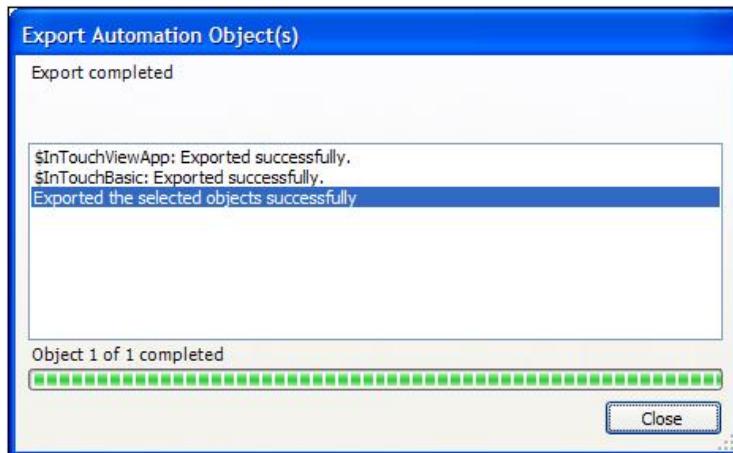


Export Automation(Object) 대화 상자가 나타납니다.



File name 입력란에 Application의 이름을 입력 (\$InTouchBasic) 하고, Save as Type에 file name의 확장자가 .aaPKG (IDE-Managed InTouch Application Backup file)인 것을 확인합니다.

- b. Save를 클릭합니다.



Export 과정이 완료되면, Export Complete 메시지가 표시 됩니다.

- c. Close를 클릭합니다.

Tip : 다른 Galaxy에 Application을 import하기 위해 .aaPKG 파일을 사용할 수 있습니다.

Stand Alone InTouch Backup – Application Publisher

Note : 이 기능은 IDE-Managed Application에 적용되는 것이 아닙니다.

InTouch Application Publisher는 InTouch Application을 다른 노드에 설치하기 위해 필요로 되는 모든 파일들과 설정과정을 포함한 자동 저장파일을 생성합니다.

Application Publisher는 인터넷상에서 다운로드 시간을 줄일 수 있도록 Application을 압축합니다.

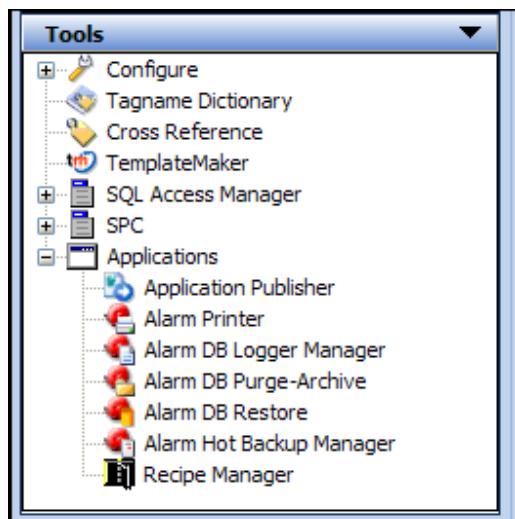
Note : Application Publisher로 작업을 끝낸 File 크기는 소스파일보다 훨씬 용량이 작아질 것입니다.

Application을 Publish하고, 다른 InTouch node에 Application file을 Load시키는 방법에는 몇 가지가 있습니다.

- | Floppy disk에 복사합니다.
- | 인트라넷망으로 file을 복사합니다.
- | 인터넷에 file을 복사합니다.
- | 특정 node에 프로그램으로 파일을 실행합니다.

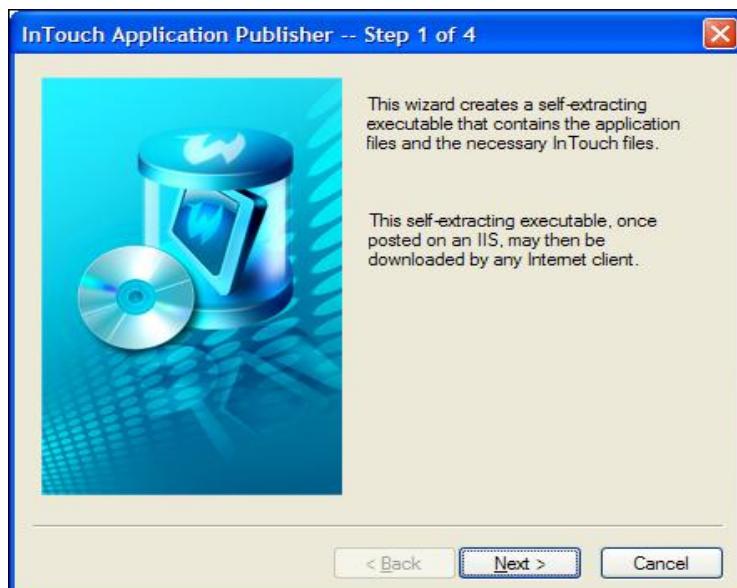
Publishing a Stand Alone InTouch application

- a. WindowViewer를 종료시키고, WindowMaker의 모든 윈도우를 닫습니다.
- b. WindowMaker에 Application Explorer의 Tool Panel에서 Application directory를 확장합니다.



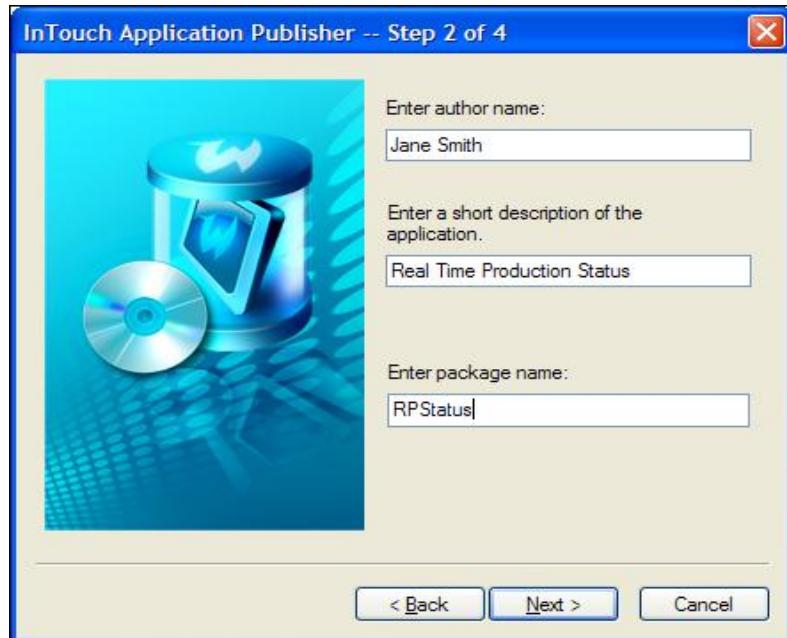
- c. Application Publisher를 더블 클릭합니다.

InTouch Application Publisher – Step 1~4 대화상자가 나타납니다.



- d. Next를 클릭합니다.

InTouch Application Publisher – Step 2 of 4 대화상자가 나타납니다.

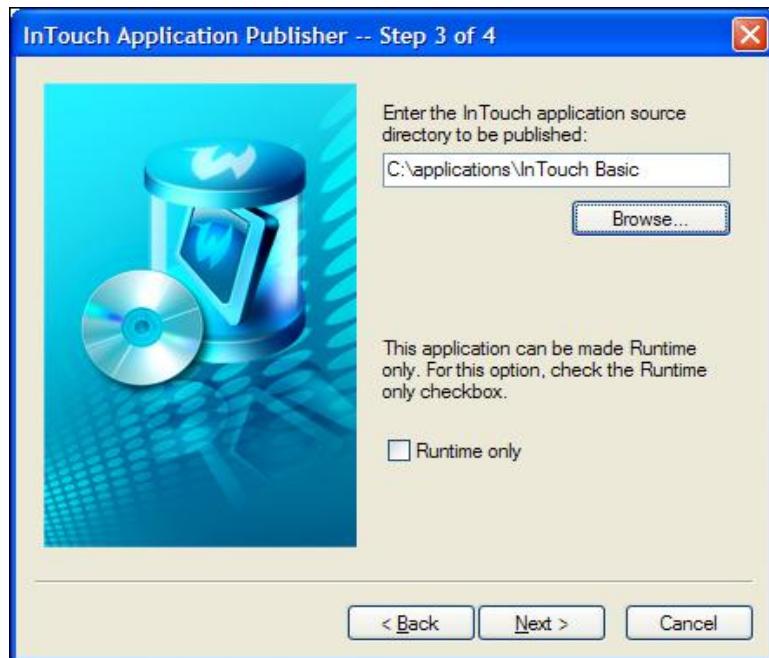


- e. Enter Author Name : Application 에 관계되어 연락할 수 있는 사람의 이름을 입력합니다. (256자 이하)
- f. Enter a short description : Application의 설명을 입력합니다. (256자 이하)
Enter Package name : Package의 이름을 입력합니다. (32자 이하)
- g. Next를 클릭합니다.

Note : 위의 정보는 SuiteVoyager 1.0 버전이나 FS WebServer 와 같은 IIS Client 를 위해 Publishing 할 때 사용되던 옵션입니다. 이 정보는 Packagename.inf 라는 파일에 저장되며, 자동저장파일 생성에서는 필요 없는 컴포넌트입니다.

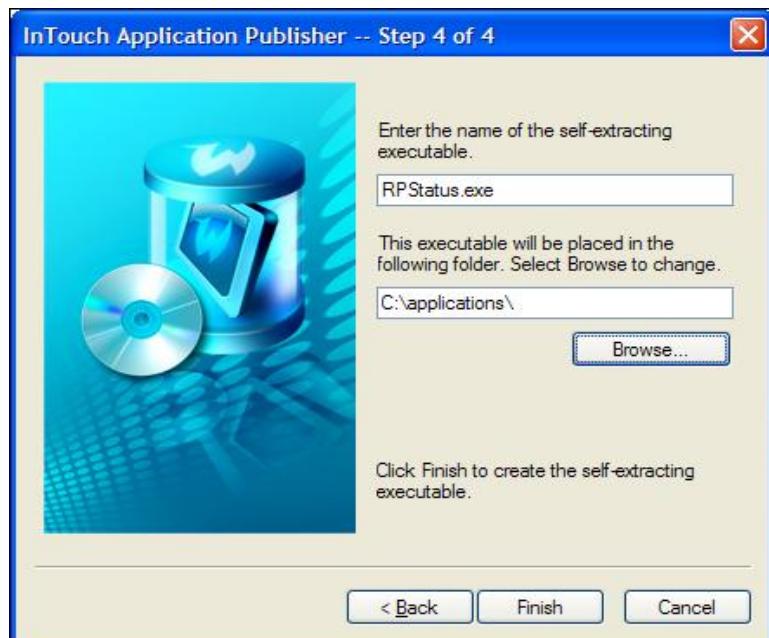
Application Publisher 는 IIS 가상 디렉토리를 Publish 할 수 있도록 지원합니다.

InTouch Application Publisher – Step 3 of 4 대화상자가 나타납니다.



- h. Enter the InTouch Application Source : InTouch Application 소스의 정확한 경로를 입력합니다. 기본값은 WindowMaker Application 디렉토리입니다 .
- i. Runtime only : Publish 된 파일에서 개발에 관련된 WindowMaker 파일을 막아줍니다.
- j. 이 옵션은 WindowMaker에서 Application을 수정할 수 없도록 합니다.
- k. Next를 클릭합니다.

InTouch Application Publisher – Step 4 of 4 대화상자가 나타납니다.



- k. 첫번째 입력란에 실행파일 이름을 확인합니다 (기본값은 앞에서 지정한 **PackageName**으로 표시 될 것입니다.)
- l. 두번째 입력란에 **Publish** 파일이 저장될 디렉토리의 경로를 입력하거나, **Browse**를 클릭하여 디렉토리를 정해 줍니다.

Note : 기본값은 Documents and Settings/AdministratorName/Local directory로 설정되어 있습니다.

- m. Floppy 드라이브 (A:\)에 직접 Publish 하지 않습니다. Publish 과정이 성공한 후, Floppy로 복사해 넣습니다.
- n. Finish를 클릭합니다.

Other Backup Scenarios

하드디스크 용량이 충분히 남지 않을 경우 Application의 크기를 줄이기 위해 Redundant나 백업 파일들을 지워야 하는 경우가 있습니다.
백업을 하기 전에는 WindowView를 반드시 종료해야 합니다.

InTouch Data Files

InTouch에 Application 디렉토리가 생성될 때, WindowMaker와 WindowView에서 사용될 설정 파일들이 생성됩니다. 또한 각각의 Application은 개별적인 디렉토리를 가져야 합니다.

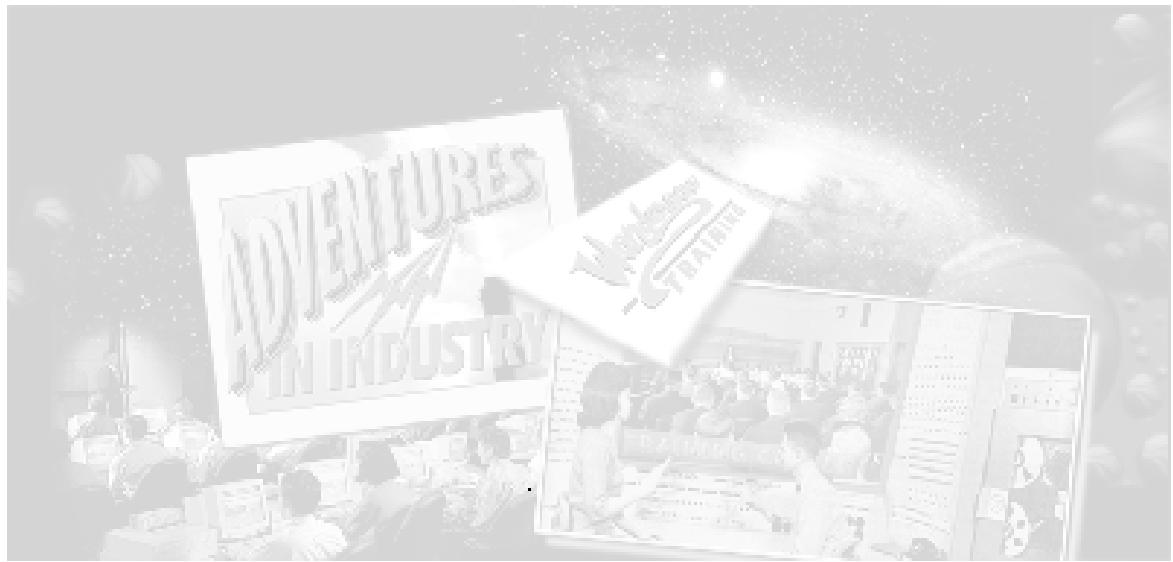
Wonderware에서 특별히 지시하지 않는 이상, 다음의 파일들은 절대로 수정해서는 안됩니다.

File	Description
ALARM.CFG	이 binary 파일은 알람 로깅과 Printing에 관한 정보를 포함합니다.
ALMGRP.INI	Txet 파일이며, Name Manager/Distributed Alarm 구성에 지정된 Alarm Group 리스트를 포함합니다.
APP.VER	이 binary 파일은 application 버전정보를 가지고 있습니다.
APPEDIT.LOK	WindowMaker에 의해 생성되는 Read-Only 파일이며 Application을 여러 노드에서 수정하는 것을 막아줍니다.
APPLIST.INI	WindowMaker palette에서 사용되는 Custom 및 Transparent color에 대한 정보를 포함하는 Text 파일입니다.
APPSTG.STG	WindowMaker palette에서 사용되는 Custom 및

	Transparent color에 대한 정보를 포함하는 binary 파일입니다
File	Description
APPLOGIC.LGC	Application Script (On Start, While Showing, On Shutdown)를 포함하는 binary 파일입니다.
COLORCH.INI	WindowMaker의 color selection box에 사용되는 32 colors를 포함하는 Text 파일입니다.
DB.INI	이 파일은 Tagname Dictionary 각각의 대화상자에 대한 기본값을 포함하고 있는 파일입니다.
DDE*.AVL, DDE*.BIN,DDE.CFG	이 파일들은 DDE Access name 과 node에 대한 정보를 포함합니다.
DHISTCFG.INI	분산 이력 시스템에 사용되는 파일입니다.
DS#####.CFG	SPC dataset 구성정보가 포함된 파일입니다.
DS#####.DAT	SPC를 사용하면서 Data 수집에 관련된 내용을 포함하는 binary 파일입니다.
DS.INF	Dataset 정보를 포함하는 binary 파일입니다.
GROUP.DEF	Tagname Alarm Group에 대한 정보를 포함하는 파일입니다.
HISTLOG.CFG	이력로깅에 사용되는 구성 정보를 포함하는 파일입니다.
HISTPRV.INI	Name Manager/Distributed History 구성에 정의된 분산이력 Provider 정보를 포함하는 Text 파일입니다.
INTOUCH.INI	InTouch에 사용되는 Application에 대한 구성 정보를 포함하는 파일입니다.
INTRACK.INI	InTrack 구성에 대한 정보를 포함하는 Text 파일입니다.
INTRACK.ITAL	InTrack Taglist 정보를 포함하는 Binary 파일입니다.
ITOCX.CFG	Application에 사용되는 ActiveX controls에 대한 Cross-reference 정보를 포함하는 Binary 파일입니다.
LINKDEFS.INI	WindowMaker 에 각 링크 대화상자에 대한 기본값을 포함하는 파일입니다.
OCX.INI	Globally Unique Identifier(GUID) 포함한 Text 파일이고, 각 ActiveX control의 설명을 Wizards/Active X 설치 메뉴로부터 InTouch에 등록합니다.
PASSWORD.BIN	Application의 보안에 사용되는 AccessLevel, UserName, Password 정보를 포함하고 있는 binary 파일입니다.
RETENTIV.X,RETENTIV.A RETENTIV.D,RETENTIV.H RETENTIV.S,RETENTIV.T	Application 의 retentive 값을 포함하고 있는 binary 파일입니다.
SECCONFIG.XML	Application에 정의된 보안 모델정보를 포함하는 파일입니다.

SERIAL.TRN	Application에 실시간 Trend에 대한 serial number를 포함하는 파일입니다.
SPC.CFG	SPC 구성에 대한 정보를 포함하는 binary 파일입니다.
File	Description
SPC.INI	SPC Dataset과 SPCPro data source name 구성에 대한 정보를 포함하는 Text 파일입니다.
SPCPASSWORD	SPCPro database에 이벤트기반, 혹은 자동 data 수집을 위한 user/password 구성을 포함하는 binary파일입니다.
SUPERTAG.DAT	Tagname dictionary에서 SuperTag를 생성할 때, WindowMaker에서 사용되는 정의된 템플릿 정보를 포함하는 binary 파일입니다.
SQL.DEF	SQL Access Manager에 정의된 템플릿과 Bindlist를 포함한 Text 파일입니다.
TAGBRW.DAT	Tag Browser에 사용된 마지막 구성정보를 포함한 Text 파일입니다.
TAGNAME.AVL TAGNAME.NDX,TAGNAME.E.X	Tagname dictionary의 item에 대한 정보를 포함하고 있는 파일들입니다.
TBOX.INI	ToolBox에 대한 초기설정을 포함하는 파일입니다. WindowMaker가 종료될 때마다 update 됩니다.
*.TRN	윈도우의 Trend 정보를 나타내는 파일입니다.
VERS_RES.INF	Application의 화면 해상도와 InTouch version number를 포함하는 파일입니다.
VIEW.INI	WindowViewer에 사용되는 기본 윈도우 선택정보를 포함하는 파일입니다.
*.WIN	각 윈도우 자세한 설명을 포함한 WindowMaker 파일이며, 이 파일은 win#####.win 형태로 보여집니다.
WIZ.INI	Wizards/ActiveX 설치 메뉴로부터 InTouch에 Wizard를 등록하는 것에 대한 정보를 포함하는 파일입니다.
WM.INI	WindowMaker가 시작될 때, 기본 윈도우 선택정보를 포함하는 파일입니다.
*.WVW	윈도우는 최초 WindowViewer를 실행할 때 Compile 되고, .www 파일로 저장하게 됩니다. 이 파일은 WindowViewer가 의 로딩시간을 향상시킬 수 있습니다.
WWREPORT.INI	InTrack report 기능을 사용할 때 증가되는 태그의 태그명을 포

	함하는 Text파일입니다.
WW_WDWS.NDX	Application 원도우 이름의 Cross-reference 정보를 포함하는 파일입니다.
*.AEH	Wonderware의 모든 제품에 대한 Error Log를 포함하고, WWLOGGER.EXE 프로그램에 의해 생성되는 Text파일입니다.
*.ALG	Alarm 파일입니다. InTouch 알람프로그램에 의해 생성되는 Text 파일이며, 모든 알람 이력을 포함합니다.
File	Description
*.LGH	InTouch의 로깅 데이터를 포함한 binary 파일입니다.
*.IDX	*.LGH 파일에 대한 인덱스 파일입니다.
*.CSC	Condition Script를 포함하는 binary 파일입니다.
*.KSC	Key Script를 포함하는 binary 파일입니다.
*.CSV	콤마분리형변수형태의 Text파일입니다.
*.TXT	Text 파일
*.BMP , *.JPG, *.PCX, *.TGA	Bitmap 객체에 포함되어있는 그래픽/이미지 파일입니다.
*.PAL	WindowMaker color palette나 3rd party 유ти리티로부터 Palette Export에 의해 생성되는 원도우 palette 파일입니다.
*.WAV	Microsoft 사운드 파일
*.NEH	16비트 Application 이 32비트를 변환될 때, Wonderware NT Conversion Logger에 의해 생성되는 Text 파일이며, 변환과정의 log를 포함하고 있습니다.
*.DCH	Data Change Script를 포함하는 binary 파일입니다.
*.NMD	Quick Function script를 포함하는 binary 파일입니다.
*.OES	ActiveX 이벤트 script를 포함하는 binary 파일입니다.
*.?BK	.?BK로 끝나는 모든 파일은 Backup 파일입니다. InTouch는 수정된 모든 파일에 대해 Backup 파일을 생성합니다.



참 조

Section 1 – Dotfields

Section 2– InTouch QuickScript Function

Section 1 – Dot Fields

DotFields	Description
.Ack	로컬 알람의 모든 타입에 대해 알람 인식(Ack) 상태를 감시하고 제어합니다.
.AckDev	로컬 Deviation 알람에 Ack 상태를 감시하고 제어합니다.
.AckDsc	로컬 Discrete 알람의 Ack 상태를 감시하고 제어합니다.
.AckROC	로컬 rate-of-change 알람의 Ack 상태를 감시하고 제어합니다.
.AckValue	로컬의 Value 알람의 Ack 상태를 감시하고 제어합니다.
.Alarm	명시된 태그나 알람 그룹에 알람 조건이 발생하면 1의 값을 갖습니다.
.AlarmAccess	알람 디스플레이 창에 선택된 알람과 연결된 태그명의 Access name 을 돌려줍니다.
.AlarmAckModel	태그에 연관되어있는 Ack Model을 보여줍니다. 0 => condition(default) 1 => event oriented 2 => expanded summary
.AlarmClass	알람 디스플레이 창에 선택된 알람과 연관된 태그의 class 를 돌려줍니다.
.AlarmComment	태그명이 아닌 알람에 대해 설명하는 읽고 쓸 수 있는 문자열
.AlarmDate	알람 디스플레이 창에 선택되어있는 알람과 관련된 태그의 날짜를 돌려줍니다.
.AlarmDev	명시된 태그나 알람 그룹에 deviation 알람이 발생하게 되면 1의 값을 갖습니다.
.AlarmDevCount	주어진 태그명이나 알람 그룹에 Deviation 알람이 발생하는 수를 돌려줍니다.
.AlarmDevDeadband	Minor, major deviation 알람에 대한 퍼센트 deadband 를 감시하고 제어합니다.
.AlarmDevUnAckCount	주어진 태그명이나 알람 그룹에 UnAck된 deviation 알람의 수를 Counting 하여 돌려줍니다. (minor , major deviation 알람 포함)
.AlarmEnable	태그명이나 알람 그룹의 알람 기능을 억제하거나 가능하게 합니다.
.AlarmGroup	분산 알람 디스플레이 화면표시에 현재의 쿼리 목록을 나타냅니다.
.AlarmGroupSel	분산 알람 디스플레이에 선택되어있는 알람에 연관된 태그명의 알람 그룹을 돌려줍니다.
.AlarmHiDisabled .AlarmHiEnable	High 상태의 이벤트나 알람을 억제하거나 가능하게 합니다.
.AlarmHiHiDisabled .AlarmHiHiEnabled	HiHi 상태의 이벤트나 알람을 억제하거나 가능하게 합니다.
.AlarmHiInhibitor	Hi 알람 상태에 연관된 알람 inhibitor 태그명을 돌려줍니다.
.AlarmLimit	분산 알람 디스플레이에 선택되어있는 알람에 연관된 태그명의 limit 값을 돌려줍니다.

.AlarmLoDisabled / .AlarmLoEnabled	Lo 상태의 이벤트나 알람을 억제하거나 가능하게 합니다.
.AlarmLoInhibitor	Lo 알람 상태에 연관된 알람 inhibitor 태그명을 돌려줍니다.
.AlarmLoLoDisabled / .AlarmLoLoEnabled	LoLo 상태의 이벤트나 알람을 억제하거나 가능하게 합니다.
.AlarmLoLoInhibitor	LoLo 알람 상태에 연관된 알람 inhibitor 태그명을 돌려줍니다.
.AlarmMajDevDisabled / .AlarmMajDevEnabled	Major deviation 이벤트나 알람을 억제하거나 가능하게 합니다.
.AlarmMajDevInhibitor	Major deviation 알람 상태에 연관된 알람 inhibitor 태그명을 돌려줍니다.
.AlarmMinDevDisabled / .AlarmMinDevEnabled	Minor deviation 이벤트나 알람을 억제하거나 가능하게 합니다.
.AlarmMinDevInhibitor	Minor deviation 알람 상태에 연관된 알람 inhibitor 태그명을 돌려줍니다.
.AlarmName	분산 알람 디스플레이에 선택된 알람의 name 을 돌려줍니다.
.AlarmOprName	분산 알람 디스플레이에 선택된 알람의 operator name 을 돌려줍니다.
.AlarmOprNode	분산 알람 디스플레이에 선택된 알람의 operator node 를 돌려줍니다.
.AlarmPri	분산 알람 디스플레이에 선택된 알람의 priority(1- 999) 를 돌려줍니다.
.AlarmProv	분산 알람 디스플레이에 선택된 알람의 provider 를 돌려줍니다.
.AlarmROC	명시된 태그에 rate- of- change 알람이 발생하면 1 의 값을 갖습니다.
.AlarmROCCount	주어진 태그명이나 알람 그룹에 rate- of- change 알람의 발생 수를 counting 합니다.
.AlarmROCDisabled / .AlarmROCEnabled	ROC 알람 상태의 이벤트나 알람을 억제하거나 가능하게 합니다.
.AlarmROCIInhibitor	ROC 알람 상태에 연관된 알람 inhibitor 태그명을 돌려줍니다.
.AlarmROCUnAckCount	주어진 태그명이나 알람 그룹의 Ack 되지않은 rate- of- change 알람의 수를 counting 합니다.
.AlarmState	분산 알람 디스플레이에 선택되어있는 알람과 연관된 태그의 state 를 돌려줍니다.
.AlarmTime	분산 알람 디스플레이에 선택되어있는 알람과 연관된 태그의 time 을 돌려줍니다.
.AlarmTotalCount	주어진 태그명이나 알람 그룹의 발생되는 알람의 전체 수를 counting 합니다.
.AlarmType	분산 알람 디스플레이에 선택된 알람의 type 을 돌려줍니다.
.AlarmUnAckCount	주어진 태그명이나 알람 그룹에 UnAck 된 알람의 수를 counting 합니다.
.AlarmUserDefNum1	특정 알람에 User 별로 값을 셋팅하여 값을 이용해 쿼리를 할 수 있습니다.
.AlarmUserDefNum2	특정 알람에 User 별로 값을 셋팅하여 값을 이용해 쿼리를 할 수 있습니다.

.AlarmUserDefStr	특정 알람에 User 별로 문자를 셋팅하여 문자를 이용해 쿼리를 할 수 있습니다.
.AlarmValDeadband	알람의 deadband 값을 감시하고 제어합니다.
.AlarmValue	분산 알람 디스플레이에 선택된 알람에 연관된 태그명의 알람 값을 돌려줍니다.
.AlarmValueCount	주어진 태그나 알람 그룹에 value 알람이 발생한 수를 counting 합니다.
.AlarmValueUnAckCount	주어진 태그나 알람 그룹에 UnAck 된 value 알람의 수를 counting 합니다.
.Caption	Check box 와 함께 디스플레이 될 메시지를 결정합니다.
.ChartLength	이력트랜드 차트에 표시될 시간의 범위를 제어합니다.
.ChartStart	이력트랜드 차트에 날짜와 시간의 시작을 관리합니다.
.Comment	태그명 사전의 태그에 대한 Comment 필드의 값을 나타냅니다.
.DevTarget	Mimor,major deviation 알람에 대한 target 을 감시하고 제어합니다.
.DisplayMode	트랜드에 값을 표현하는데 사용되는 방법을 결정합니다.
.Enabled	제어개체가 사용자에 의해 발생한 사건에 응할지 여부를 결정합니다.
.Freeze	분산 알람 디스플레이 개체의 freeze 상태를 읽고 씁니다.
.HiHiLimit	경보 확인값의 HiHi 한계를 감시하고 제어합니다.
.HiHiSet	Analog 태그에 HiHi 알람 상태를 설정할지 여부를 0,1 의 값으로 돌려줍니다.
.HiHIStatus	HiHi 한계경보가 있는지 없는지를 알아냅니다.
.HiLimit	경보 확인값의 Hi 한계를 감시하고 제어합니다.
.HiSet	Analog 태그에 Hi 알람 상태를 설정할지 여부를 0,1 의 값으로 돌려줍니다.
.HiStatus	Hi 한계경보가 있는지 없는지를 알아냅니다.
.ListChanged	분산 알람 디스플레이에 대한 업데이트나 새로운 알람 발생을 나타냅니다.
.ListCount	목록 상자나 콤보 상자 안의 항목 숫자를 결정합니다.
.Listindex	목록에서 현재 선택된 항목의 지수를 결정합니다. 지수는 목록에서 특정한 항목을 정의하는 번호입니다.
.LoLimit	경보 확인 값의 Low 한계를 감시하고 관리합니다.
.LoLoLimit	경보 확인 값의 LoLo 한계를 감시하고 관리합니다.
.LoSet	Analog 태그에 Low 알람 상태를 설정할지 여부를 0,1 의 값으로 돌려줍니다.
.LoLoSet	Analog 태그에 LoLo 알람 상태를 설정할지 여부를 0,1 의 값으로 돌려줍니다.
.LoLoStatus	LoLo 한계경보가 있는지 없는지를 알아냅니다.

.LoStatus	Low 한계경보가 있는지 없는지를 알아냅니다.
.MajorDevPct	경보 조사에서 편차의 퍼센트를 감시하고 제어합니다.
.MajorDevSet	태그에 최대편차 퍼센트가 설정이 되었는지 여부를 0 또는 1 의 값으로 나타냅니다.
.MajorDevStatus	명시된 태그에 주 편차 경보가 발생했는지를 결정합니다.
.MaxEU	명시된 태그에 최대값 (엔지니어링 단위). 이 Dotfield 는 DDE 유형 태그에서만 유효합니다.
.MaxRange	트랜드를 기록하고 있는 각 태그의 화면표시되는 이력 트랜드 범위의 퍼센트를 나타냅니다.
.MaxRow	WindowViewer 가 I/O Server 로부터 받은 실제 Row 값에 대한 최대 값
.MinEU	명시된 태그에 최소값 (엔지니어링 단위). 이 Dotfield 는 DDE 유형 태그에서만 유효합니다.
.MinorDevPct	경보 조사에서 편차의 퍼센트를 감시하고 제어합니다.
.MinorDevSet	태그에 최소편차 퍼센트가 설정이 되었는지 여부를 0 또는 1 의 값으로 나타냅니다.
.MinorDevStatus	명시된 태그에 주 편차 경보가 발생했는지를 결정합니다.
.MinRange	트랜드를 기록하고 있는 각 태그의 화면 표시되는 이력 트랜드 범위의 퍼센트를 나타냅니다.
.MinRaw	WindowViewer 가 I/O Server 로부터 받은 실제 Row 값에 대한 최소 값
.Name	String 형식에 명시된 태그의 이름을 나타냅니다.
.NewIndex	WcAddItem() 나 wcInsertItem() 를 통해 목록상자나 콤보 상자에 추가된 마지막 항목의 정수 지수를 돌려줍니다.
.NextPage	이 속성이 1 에서 0 으로 바뀔 때 경보 화면을 한 페이지 아래로 이동합니다.
.Normal	명시된 태그 명을 위한 경보가 없을 때 1 입니다.
.NumAlarms	경보 개체 안에 있는 경보의 숫자를 갖습니다.
.OffMsg	태그 명 사전 안에 명시된 이산 태그의 off 메시지에 Access 할 수 있도록 합니다.
.OnMsg	태그 명 사전 안에 명시된 이산 태그의 on 메시지에 Access 할 수 있도록 합니다.
.PageNum	경보 개체 안에 있는 현재 화면 표시된 페이지의 번호를 갖습니다.
.Pen1- .Pen8	각 펜에 의해 이력 트랜드가 기록되는 태그 명을 관리합니다.
.PendingUpdates	
.PrevPage	이 속성이 1 에서 0 으로 바뀔 때 경보 화면을 한 페이지위로 이동합니다.
.PriFrom	현재 쿼리 우선 순위의 하위 필터 값을 갖습니다.
.PriTo	현재 쿼리 우선 순위의 상위 필터 값을 갖습니다.

.ProviderReq	현재 쿼리에 필요한 경보 제공자의 숫자를 나타냅니다.
.ProviderRet	쿼리 결과를 성공적으로 돌려준 경보 제공자의 숫자를 나타냅니다.
.Quality	현재 통신하고 있는 아이템의 통신품질상태(Quality State)를 나타내며 정수값에 따라 통신품질상태를 나타낼 수 있습니다.
.QualityLimit	I/O Server 와 통신하고 있는 태그의 통신 Quality Limit 를 정수로 표시합니다.
.QualityLimitString	I/O Server 와 통신하고 있는 태그의 통신 Quality Limit 를 문자로 표시합니다.
.QualityStatus	통신품질상태(Quality State)를 정수값으로 나타내어 줍니다.
.QualityStatusString	통신품질상태(Quality State)를 문자로 나타내어 줍니다.
.QualitySubstatus	Quality Substatus 를 정수값으로 나타내어 줍니다.
.QualitySubstatusString	Quality Substatus 를 문자로 나타내어 줍니다.
.QueryState	특정 Alarm Object 를 쿼리하여 Alarm Status (Ack,UnAck) 를 보여줍니다.
.QueryType	특정 Alarm Object 를 쿼리하여 Alarm Type (Historical,Summary) 을 나타내줍니다.
.RawValue	Client로써 WindowViewer 가 I/O Server 로부터 실제의 값을 받습니다.
.ReadOnly	Text box 의 목록이 읽기 전용인지 읽기/쓰기 요인지를 결정합니다.
.Reference	런타임시 Access Name 이나 Item Name 을 동적으로 변경할 수 있도록 해 줍니다.
.ReferenceComplete	요청한 item 이 .Value 필드에 나타난 것과 같을 경우 확인을 돌려줍니다.
.ROCPct	경보 감시의 변경율을 감시하고 관리합니다.
.ROCSet	Analog 태그에 변경율 알람을 설정할지 여부를 0,1 의 값으로 돌려줍니다.
.ROCStatus	명시된 태그에 변경율 경보가 있는지 없는지를 결정합니다.
.ScooterLockLeft	이 도트 필드를 1 로 설정하면 오른쪽 스쿠터가 왼쪽 스쿠터 위치보다 왼쪽으로 움직이지 못하게 합니다.
.ScooterLockRight	아 도트 필드를 1 로 설정하면 왼쪽 스쿠터가 오른쪽 스쿠터 위치보다 오른쪽으로 움직이지 못하게 합니다.
.ScooterPosLeft	왼쪽 스쿠터의 위치를 감시하고 관리합니다.
.ScooterPosRight	오른쪽 스쿠터의 위치를 감시하고 관리합니다.
.Successful	현재 쿼리가 성공했는지의 여부를 나타냅니다.
.SuppressRetain	분산 알람 디스플레이 개체에 대한 Suppression 기능이 남아 있는지의 상태를 읽고 줍니다.
,TagID	펜에 의해 트랜드 기록되는 태그명을 감시하고 관리하기 위해 이력 트랜드 .Pen1~Pen8 TagID 태그명과 함께 사용합니다.
.TimeDate	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 정수로 표시해 줍니다.

.TimeDateString	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 문자로 표시해 줍니다.
.TimeDateTime	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 Real 값으로 표시해 줍니다.
.TimeDay	해당 태그가 I/O Server 와 통신을 한지 날짜수로 표시해 줍니다.
.TimeHour	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 시간 값으로 표시해 줍니다
.TimeMinute	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 분의 값으로 표시해 줍니다
.TimeMonth	해당 태그가 I/O Server 와 통신을 한지 얼마나 달이 지났는지를 월단위 값으로 표시해 줍니다
.TimeMsec	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 시간의 msec 값으로 표시해 줍니다
.TimeSecond	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 시간의 second 값으로 표시해 줍니다
.TimeTime	해당 태그가 I/O Server 와 통신을 한지 얼마나 시간이 지났는지를 시간의 msec 값으로 표시해 줍니다
.TimeTimeString	해당 태그가 I/O Server 와 통신을 한지 날짜를 문자값으로 표시해 줍니다.
.TopIndex	목록 상자에서 최상단의 항목의 정수 지수를 정합니다.
.TotalPages	경보개체에 있는 총 페이지의 숫자를 나타냅니다.
.UnAck	로컬 알람의 확인 상태를 제어합니다.
.UpdateCount	연관된 이력 트랜드에 업데이트가 일어날 때마다 증가 됩니다.
.UpdateInProgress	이력 읽기가 진행 중이면 1 이고 아니면 0 입니다.
.UpdateTrend	이력 트랜드 차트로 하여금 모든 현재의 값을 사용하여 업데이트를 하게끔 합니다.
.Value	명시된 태그명의 값을 갖습니다.
.Value	이는 모든 InTouch 윈도우 제어의 기본 속성입니다. 이 속성의 변화는 InTouch 데이터 베이스의 태그명과 원도우 제어 개체를 같게 합니다.
.Visible	원도우 제어가 보이는지 여부를 결정합니다.

Section 2 – InTouch QuickScript Functions

Abs()	명시된 숫자의 절대 값을 돌려줍니다.
Ack()	지역 InTouch 경보를 확인합니다.
ActiveApp()	현재 실행 중인 다른 윈도우 응용프로그램을 활성화 합니다.
Addpermission()	도메인과 그룹에 사용자 액세스레벨을 지정합니다.
AlmAckDisplay()	현재 경보화면에 보이는 경보만을 확인합니다.
AlmAckPriority()	같은 명시된 Priority 를 가진 알람을 확인합니다.
AlmAckRecent()	가장 최근에 발생한 경보를 확인합니다.
AlmAckSelect()	경보화면에 선택된 경보들만 확인합니다.
AlmAckSelectedGroup()	선택된 그룹의 알람을 확인합니다.
AlmAckSelectedPriority()	선택된 Priority 에 대한 알람을 확인합니다.
AlmAckSelectedTag()	선택된 태그에 대한 알람을 확인합니다.
AlmAckTag()	명시된 태그에 대해 알람을 확인합니다.
AlmDefQuery()	기본 설정으로 분산 알람디스플레이를 쿼리 업데이트 합니다.
AlmMoveWindow()	명시된 분산 알람디스플레이의 스크롤
AlmQuery()	명시된 분산 알람디스플레이를 쿼리합니다.
AlmSelectAll()	모든 선택된 알람을 선택합니다.
AlmSelectGroup()	명시된 그룹의 알람을 선택합니다.
AlmSelectionCount()	조작자에 의해 선택된 알람의 수를 나타냅니다.
AlmSelectPriority()	명시된 Priority 에 대한 알람을 선택합니다.
AlmSelectTag()	명시된 태그에 대한 알람을 선택합니다.
AlmSetQueryByName()	정의된 QueryName 으로 알람을 새로 쿼리 합니다.
AlmShowStats()	경보화면 통계 스크린을 나타냅니다.
AlmSuppressAll()	모든 알람을 Suppress 합니다.
AlmSuppressGroup()	주어진 그룹에 속해있는 알람을 Suppress 합니다.
AlmSuppressDisplay()	디스플레이 되어있는 알람을 Suppress 합니다.
AlmSuppressPriority()	주어진 Priority 에 속해있는 알람을 Suppress 합니다.
AlmSuppressRetain()	
AlmSuppressSelected()	선택된 알람을 Suppress 합니다

AlmSuppressTag()	선택된 태그를 Suppress 합니다.
AlmSuppressSelectedGroup()	선택된 그룹을 Suppress 합니다.
AlmSuppressSelectedPriority()	선택된 우선순위에 대해 Suppress 합니다.
AlmSuppressSelectedTag()	선택된 태그에 대한 알람을 Suppress 시킵니다.
AlmUnSelectAll()	선택된 알람을 모두 해제 시킵니다.
AlmUnSuppressAll()	Suppress 된 알람을 해제 시킵니다.
ArcCos()	주어진 -1 부터 1 사이의 값에 이 함수는 그 cosine 의 값이 이와 동일한 각도를 -90 부터 90 사이의 도로 돌려줍니다.
ArcSin()	주어진 -1 부터 1 사이의 값에 이 함수는 그 sine 의 값이 이와 동일한 각도를 -90 부터 90 사이의 도로 돌려줍니다.
ArcTan()	주어진 값에 이 함수는 그 tangent 의 값이 이와 동일한 각도를 -90 부터 90 사이의 도로 돌려줍니다.
AttemptInvisibleLogon()	명시된 사용자나 암호를 사용하여 InTouch 에 로그온을 시도합니다.
ChangePassword()	암호 변경 대화 상자를 나타냅니다.
Cos()	각도로 주어진 값의 cosine 값을 돌려줍니다.
DialogStringEntry()	알파뉴메릭 키보드를 스크린에 나타내어 사용자로 하여금 태그명 사전에 있는 메시지 태그의 현재 문자열 값을 변경하도록 해 줍니다.
DialogValueEntry()	숫자 키 패들을 스크린에 나타내어 사용자로 하여금 태그명 사전에 있는 이산, 정수 및 실수 태그의 현재 값을 바꿀 수 있게 해줍니다.
Dtext()	동적으로 이산 태그명에 따라 메시지 태그명을 바꿉니다.
EnableDisablekeys()	Windows Keys 나 Alt ,Esc 키를 활성화시키거나 억제시킵니다.
Exp()	E 의 누승을 돌려줍니다.
FileCopy()	SourceFile 을 DestFile 로 복사합니다.
FileDelete()	필요 없거나 원하지 않는 파일을 삭제합니다.
FileMove()	복사를 하는 대신 파일을 한 지역에서 다른 곳으로 움직이는 것을 제외하고는 Filecopy() 함수와 비슷합니다.
FileReadFields()	명시된 파일에서 쉼표 구분 변수 (CSV) 레코드를 읽습니다.
FileReadMessage()	명시된 파일로부터 명시된 숫자의 바이트를 읽습니다.
FileWriteFields()	쉼표 구분된 변수 (CSV) 레코드를 명시된 파일에 씁니다.
FileWriteMessage()	명시된 숫자의 바이트를 명시된 파일에 씁니다.
GetnodeName()	NetDDE 노드 이름을 문자열 변수에 돌려줍니다.

GetPropertyD0	실행시간 중 명시된 속성의 이산 값을 읽습니다.
GetPropertyI0	실행시간 중 명시된 속성의 정수 값을 읽습니다.
GetPropertyM0	실행시간 중 명시된 속성의 메시지 값을 읽습니다.
Hide0	스크립트 안에서 각종 윈도우를 숨깁니다.
HideSelf0	현재 활성중인 윈도우를 숨깁니다.
HTGetLastError0	명시된 펜에 마지막 읽기에서 오류가 발생 했는가를 결정합니다.
HTGetPenName0	명시된 트랜드의 명시된 펜 번호에 현재 사용되는 태그의 태그명을 돌려줍니다.
HTGetTimeAtScooter0	ScootNum 과 ScootLoc 에 의해 명시된 스쿠터 위치의 표현에 대해 GMT 1970년 00:00:00 부터의 시간을 초로 돌려줍니다.
HTGetTimeStringAtScooter0	ScootNum 과 ScootLoc 에 의해 명시된 스쿠터 위치에 있는 표본의 시간/날짜를 포함한 문자열을 돌려줍니다.
HTGetValue0	전체 트랜드의 명시된 펜을 위한 요청된 유형의 값을 돌려줍니다.
HTGetValueAtScooter0	명시된 스쿠터 위치, 트랜드 그리고 펜 번호 표본의 요청한 유형 값을 돌려줍니다.
HTGetValueAtZone0	트랜드의 명시된 펜의 왼쪽과 오른쪽 스쿠터 위치에 포함 된 데이터의 요청된 유형 값을 돌려줍니다.
HTScrollLeft0	트랜드의 시작 시간을 트랜드 너비의 퍼센트만큼 현재 시간 보다 옛날 시간으로 설정합니다.
HTScrollRight0	트랜드의 시작 시간을 트랜드 너비의 퍼센트만큼 현재 시간 보다 앞의 시간으로 설정합니다.
HTSelectTag0	Select Tag 대화상자를 나타냅니다.
HTSetPenName0	트랜드의 펜에 다른 태그명을 지정합니다.
HTUpdateToCurrentTime0	데이터를 읽어 끝 시간을 현재의 시간과 동일하게 하여 나타냅니다.
HTZoomIn0	새로운 차트의 너비와 시작 시간을 계산합니다.
HTZoomOut0	새로운 차트의 너비와 시작 시간을 계산합니다. 새 차트의 너비는 옛 차트의 너비를 2로 곱한 것과 같습니다.
InfoAppActive0	응용프로그램이 활성 중인지를 실험합니다.
InfoAppTitle0	응용프로그램 제목이나 현재 실행 중인 명시된 프로그램의 Windows 작업 목록이름을 돌려줍니다.
InforDisk0	특정한 지역 디스크 드라이브에 대한 정보를 돌려줍니다.
InfoFile0	특정한 파일이나 하위 디렉토리에 관한 정보를 돌려줍니다.
InfoInTouchAppDir0	현재 InTouch 의 응용프로그램 디렉토리를 돌려줍니다.
InfoResources0	각종 시스템 자원 값을 돌려줍니다.
Int0	명시된 숫자와 같거나 적은 다음의 정수를 돌려줍니다.
InvisibleVerifyCredentials0	

IOGetApplication()	명시된 태그에 정의된 액세스 명에 대해 정의되어 있는 Application name 을 돌려줍니다.
IOGetNode()	명시된 태그에 정의된 액세스 명에 대해 정의되어 있는 node name 을 돌려줍니다.
IOGetTopic()	명시된 태그에 정의된 액세스 명에 대해 정의되어 있는 Topic name 을 돌려줍니다.
IOReinitialize()	존재하는 모든 통신을 다시 초기화로 시작시킵니다.
IOSetAccessName()	어플리케이션 실행 중에 AccessName 의 속성을 수정할 수 있습니다.
IOSetItem()	어플리케이션 실행중에 I/O 태그의 Accessname 과 item 을 변경할 수 있습니다.
IOStartUninitConversations()	
IsAnyAsynchFunctionBusy()	
IsAssignedRole()	
Log()	숫자의 10진 로그 값을 돌려줍니다.
LogMessage()	WWLogger 에 사용자 정의 메시지를 기록합니다.
LogN()	X의 베이스 n로그를 돌려줍니다.
LogOff()	InTouch 사용자를 LogOff 시킵니다.
Pi()	Pi 값을 돌려줍니다.
PlaySound()	.wav 파일이름에 의해 명시된 웨이브 유형 소리나 WIN.INI 파일의 [sounds]부분의 항목으로 명시된 윈도우의 사운드 장치를 통해 들려줍니다.
PostLogonDialog()	InTouch 로그 온 대화상자를 가져옵니다.
PrintHT()	명시된 이력 트랜드 유형 태그명과 관련된 이력 트랜드 차트를 인쇄하기 위한 버튼을 작성합니다. 이 함수를 사용할 때는 이력 트랜드가 화면에 보여야 합니다.
PrintScreen()	명시된 화면을 인쇄합니다.
PrintWindow()	명시된 윈도우를 인쇄합니다.
QueryGroupMembership()	OS 보안 모드에서 현재 로그온 되어있는 사용자 그룹을 확인합니다.
RecipeDelete()	현재 정의된 Recipe 이름을 명시된 recipe 서식 파일에서 삭제합니다.
RecipeGetMessage()	처리된 함수의 오류코드를 아날로그 태그명에 그리고 오류 코드에 해당하는 메시지 태그명을 씁니다.
RecipeLoad()	특정한 recipe 를 특정한 태그명 단위에 읽어 들입니다.
RecipeSave()	새로 작성된 recipe 를 저장하거나 기존의 recipe 에 일어난 변화를 명시된 recipe 서식 파일에 저장합니다.
RecipeSelectNextRecipe()	Recipe 서식 파일에 현재 정의된 다음 recipe 이름을 선택합니다.
RecipeSelectPreviousRecipe()	Recipe 서식 파일에 현재 정의 된 앞의 Recipe 이름을

	선택합니다.
RecipeSelectRecipe()	Recipe 서식 파일 상에 현재 정의된 특정한 Recipe 이름을 선택합니다.
RecipeSelectUnit()	현재 Recipe 값이 읽혀질 태그명의 단위를 선택합니다.
ReloadWindowViewer()	
RestartWindowViewer()	사용자로 하여금 WindowViewer 를 종료했다가 다시 시작하는 것의 관리를 허용합니다.
Round()	실수를 명시된 정밀도로 반올림합니다.
SendKeys	키를 다른 응용프로그램에 보냅니다. 다른 응용프로그램에게는 키가 키보드에 입력된 것처럼 보입니다. 이 기능은 응용프로그램에 데이터를 입력하거나 응용프로그램에 명령을 주는데 사용될 수 있습니다.
SetPropertyD()	실행시간 중 쓰여질 속성의 이산 값을 명시합니다.
SetPropertyI()	실행시간 중 쓰여질 속성의 정수 값을 명시합니다.
SetpropertyM()	실행시간 중 쓰여질 속성의 메시지 값을 명시합니다.
Sgn()	값의 부호를 알아냅니다. (양수, 0 혹은 음수)
Show	명시된 윈도우를 화면표시 합니다.
ShowAt()	윈도우가 보일 때, 수평 및 수직 픽셀 위치를 명시합니다.
ShowHome()	“홈” 윈도우를 화면표시 합니다. “홈” 윈도우는 특수(S)/설정(O).../ 홈 윈도우 보기(M).. 명령을 통해 설정된 윈도우들입니다.
ShowTopLeftAt()	윈도우가 보일 때 그 상단 왼쪽 모서리의 수평 및 수직 픽셀 위치를 명시합니다.
Sin()	각의 sine 값을 돌려줍니다.
SQLAppendStatement()	String 의 값으로 SQL 명령을 계속 이어 갑니다. 오류는 함수가 종료할 때 주어집니다.
SQLClearParam()	특정한 매개변수의 값을 지웁니다. SQLExecute() 를 실행하기 전에 SQLSetParam() 을 다시 실행 해야합니다.
SQLClearStatement()	SQLHandle 에 명시된 명령문과 관계된 자원을 풀어 줍니다.
SQLClearTable()	데이터베이스의 표 내의 모든 레코드를 삭제하여 표만 유지합니다.
SQLCommit()	SQLCommit() 명령은 일련의 업무 명령을 일단락 짓습니다.
SQLConnect()	CommetString 에 명시된 데이터베이스에 InTouch 를 연결합니다.
SQLCreateTable()	명시된 표 서식의 매개변수를 사용하여 데이터베이스 내에 표를 작성합니다. 표 서식(SQL.DEF 파일에 정의됨)은 데이터베이스 표의 구조를 결정합니다.
SQLDelete()	하나 또는 복수의 레코드를 삭제합니다.
SQLDisconnect()	데이터베이스로부터의 연결을 종료합니다.

SQLDropTable()	표를 삭제합니다.
SQLEnd()	SQLSelect() 후에 이 함수를 사용하여 결과 표를 저장하는데 소용되는 자원을 풀어 줍니다.
SQLErrorMsg()	특정한 resultCode 와 연관된 텍스트 오류메시지를 읽습니다.
SQLExecute()	SQL 명령을 실행합니다.
SQLFirst()	마지막에 실행된 SQLSelect() 에 의해 작성된 결과표의 첫 레코드를 선택합니다. 이 명령문을 사용하기 전에 SQLSelect() 가 이미 실행되었어야 합니다.
SQLGetRecord()	현재의 선택 임시저장소에서부터 RecordNumber 로 명시된 레코드를 읽습니다.
SQLInsert()	주어진 BindList 의 태그명의 값을 사용하여 언급된 표에 새 레코드를 삽입합니다.
SQLInsertEnd()	명령문을 풀어줍니다.
SQLInsertExecute()	이미 준비된 명령문을 실행합니다.
SQLInsertPrepare()	실행할 삽입 명령문을 작성하고 준비합니다.
SQLLast()	마지막 SQLSelect() 에 의해 작성된 결과 표의 마지막 레코드를 선택합니다. 이 명령문을 사용하려면 SQLSelect() 가 이미 실행되었어야 합니다.
SQLLoadStatement()	Filename 에 있는 명령문을 읽습니다. 이 시점에서 명령문은 SQLSetStatement() 에 의해 작성된 명령문과 같으며 SQLAppendStatement() 를 통해 덧붙일 수 있으며 혹은 SQLExecute() 를 통해 실행할 수 있습니다. 한 파일에 한명령문만 허용합니다. 그러나 SQLPrepareStatement() 나 SQLExecute() 가 아직 실행되지 않았으면 SQLAppendStatement() 를 사용하여 명령문에 덧붙일 수 있습니다.
SQLManageDSN()	마이크로소프트의 ODBC 관리자 설정 프로그램을 실행합니다. 이것은 모든 데이터 원본 이름을 추가,삭제 혹은 변경하는데 사용할 수 있습니다.
SQLNext()	마지막 SQLSelect() 에 의해 작성된 결과 표에서 다음 레코드를 선택합니다. 이 명령문을 사용하려면 SQLSelect() 가 이미 실행되었어야 합니다.
SQLNumRows()	마지막 SQLSelect() 명령문에 명시된 선택기준에 의해 선택된 열의 숫자를 가리킵니다.
SQLPrepareStatement()	이 명령문을 사용하기 전에 SQLSelect() 가 이미 실행되어 있어야 합니다. SQLSetStatement() SQLLoadStatement() 를 통해 작성된 SQL 명령문을 준비합니다.
SQLPrev()	마지막 SQLSelect() 에 의해 작성된 결과 표에서 앞의 레코드를 선택합니다.
SQLRollback()	SQLRollback() 명령은 가장 최근에 적용된 작업 집합을 취소합니다.
SQLSelect()	데이터베이스로 하여금 표에서 정보를 읽게 합니다. SQLSelect() 함수가 실행되면, 임시 결과 표가 메모리에 작성되는데 이 표는 SQLFirst() , SQLLast() , SQLNext()

	및 SQLPrev() 등으로 훑어 볼 수 있는 레코드를 포함하고 있습니다.
SQLSetParamChar()	명시된 매개변수의 값을 명시된 문자열에 설정합니다. SQLSetParaChar() 는 실제로 실행되기 전에 몇 번이고 부를 수있으며 모든 주어진 값을 모두 연결아 연결한 매개변수 값을 작성합니다. 길이가 0인것은 무시합니다.
SQLSetParamDate()	명시된 날짜 매개변수의 값을 명시된 문자열에 설정합니다.
SQLSetParamDateTime()	명시된 날짜- 시간 매개변수의 값을 명시된 문자열에 설정합니다.
SQLSetParamDecimal()	명시된 소수점숫자 매개변수의 값을 명시된 문자열에 설정합니다. Precision 은 값의 숫자 자리 수이며 Scale 은 소수점 오른쪽의 숫자 자리 수를 나타냅니다.
SQLSetParamFloat()	명시된 매개변수의 값을 명시된 ParameterValue 에 설정합니다.
SQLSetParamInt()	명시된 매개변수의 값을 명시된 ParameterValue 에 설정합니다.
SQLSetParamLong()	명시된 매개변수의 값을 명시된 ParameterValue 에 설정합니다.
SQLSetParamNull()	명시된 매개변수의 값을 NULL 로 설정합니다.
SQLSetParamTime()	명시된 시간 매개변수의 값을 명시된 문자열에 설정합니다.
SQLSetStatement()	이미 설정된 연결 ConnectionID 에 String 의 내용을 사용하여 SQL 명령문 임시 저장을 시작합니다. ConnectionID 당 하나의 임시저장소가 있을 수 있습니다. 함수가 돌아올 때 오류를 돌려줍니다.
SQLTransact()	SQLTransact() 명령은 업무명령 그룹의 시작을 정의합니다. SQLTransact() 명령한 SQLCommit() 명령 사이에 실행되는 일련의 명령들을 업무 집합이라 합니다. 한 업무 집합은 하나의 업무처럼 처리합니다. SQLTransact() 명령이 실행된 후 그 후의 모든 업무는 SQLCommit() 명령이 떨어질 때까지는 데이터베이스에 적용되지 않습니다.
SQLUpdate()	현재 태그명의 값으로 레코드를 업데이트 합니다.
SQLUpdateCurrent()	현재 선택된 레코드를 InTouch 의 새 값으로 업데이트합니다.
Sqrt()	자동으로 InTouch 로 하여금 명령문 뒤의 값의 제곱근을 계산하게 합니다.
StartAPP	자동으로 다른 Window 응용프로그램을 시작합니다.
StringASCII()	명시된 메시지 태그의 첫 글자의 ASCII 값을 돌려줍니다.
StringChar()	명시된 ASCII 코드에 상당하는 글자를 돌려줍니다.
StringFromIntg()	정수를 다른 진법으로 변환하여 문자열로 나타냅니다.
StringFromReal()	실수 값을 부동 소수점 숫자 혹은 지수 표기법의 문자열 표시로 변환합니다.

StringFromTime()	시간 값(1970년 1월 1일 이후의 시간 경과를 초로 나타냄)을 특정한 문자열 표시로 변환합니다.
StringFromTimeLocal()	시간 값(1970년 1월 1일 이후의 시간 경과를 초로 나타냄)을 특정한 문자열 표시로 변환합니다.
StringInString()	Text 안에서 SearchFor 가 처음 나타나는 위치를 돌려줍니다.
StringLeft()	명시된 메시지 태그의 첫(혹은 맨 왼쪽의)글자를 돌려줍니다.
StringLen()	명시된 메시지 태그의 길이를 돌려줍니다.
StringLower()	명시된 메시지 태그의 모든 대문자를 소문자로 바꿉니다.
StringMid()	메시지 태그에서 명시된 위치부터 시작하여 명시된 숫자의 글자를 돌려줍니다.
StringReplace()	제공한 문자열의 특정한 부분을 변경하거나 교체합니다. 이 함수는 문자열 태그, 글자, 단어 혹은 문장을 교체할 수 있습니다.
StringRight()	명시된 메시지 태그의 마지막 글자(들)을 돌려줍니다.
StringSpace()	메시지 태그나 식 안에 간격 문자의 문자열을 만듭니다.
StringTest()	어느 유형인가를 알기 위해 메시지 태그의 첫 글자를 테스트 합니다.
StringToIntg()	메시지 태그명의 숫자값을 수학적 계산을 하 수 있는 정수 값으로 변환합니다.
StringToReal()	메시지 태그명의 숫자 값을 수학적 계산을 할 수 있는 실수 값으로 변환합니다.
StringTrim()	메시지 태그에서 원하지 않는 간격 문자를 제거 합니다.
StringUpper()	명시된 메시지 태그의 모든 소문자를 대문자로 바꿉니다.
Tan()	주어진 각의 tangent값을 돌려줍니다.
Text()	메시지 유형의 태그명으로 하여금 아날로그태그명의 값을 FormatText 에 명시된 대로 화면 표시하게 합니다.
Trunc()	실수를 소수점 오른쪽을 제거합니다.
TseGetClientId()	Terminal server client 에서 어플리케이션이 실행되고 있다면 client ID 를 문자열로 돌려줍니다 (client 의 TCP/IP 주소)
TseQueryRunningOnConsole()	Terminal server console 에서 어플리케이션이 실행되고 있으면 1의 integer 값을 돌려줍니다.
TseQueryRunningOnClient()	Terminal server client 에서 어플리케이션이 실행되고 있으면 1의 integer 값을 돌려줍니다.
wcAddItem()	목록상자나 콤보상자에 제공된 Message 문자열을 추가합니다.
wcClear()	목록상자나 콤보상자에서 모든 항목을 삭제합니다.
WcDeleteItem()	목록상자나 콤보상자에서 지수 매개변수와 관련된 항목을 삭제합니다.
WcDeleteSelection()	목록에서 현재 선택된 항목을 삭제합니다.

WcErrorMessage()	오류를 설명하는 메시지 문자열을 돌려줍니다.
WcFindItem()	목록 상자나 콤보상자에서 주어진 Message 문자열과 동일한 첫 항목의 지수를 결정합니다.
WcGetItem()	목록 상자나 콤보상자에서 해당되는 ItemIndex 와 연관된 항목 문자열의 값 속성을 돌려줍니다.
WcGetItemData()	ItemIndex 매개변수에 의해 명시되는 항목과 관계된 정수 값을 정합니다.
WcInsertItem()	목록에 메시지 문자열을 삽입합니다. ItemIndex 는 문자열을 받을 목록 상자의 위치번호에 해당하는 지수입니다.
WcLoadList()	목록 상자나 콤보 상자의 내용을 Message 파일에 있는 항목들로 교체합니다.
WcLoadText()	텍스트 상자의 내용을 Message 파일 내용으로 교체합니다.
WcSaveList()	Message 파일의 내용을 목록 상자나 콤보 상자 개체에 있는 항목들로 교체합니다.
WcSaveText()	텍스트 상자안에 있는 텍스트를 Message 에 명시된 파일에 저장합니다. 파일이 없으면 새로 작성됩니다. 파일이 있으면 읽기/쓰기 이어야 합니다.
WcSetItemData()	항목 (Number)의 정수 값을 ItemIndex 로 명시된 목록의 항목에 지정합니다. 이 함수는 문자열에 숫자를 지정하는 것을 허용합니다.
Wwcontrol()	사용자로 하여금 InTouch 에서 화면 되살리기, 화면 축소, 전체화면 보기 및 응용프로그램 닫기 등의 기능을 줌으로써 다른 응용프로그램을 관리할 수 있게끔 해줍니다.
Wwexecute()	명시된 응용프로그램과 토픽에 명령을 보냅니다.
Wwpoke()	명시된 응용프로그램, 토픽 그리고 항목에 값을 보냅니다.
WwRequest()	특정한 응용프로그램, 토픽, 그리고 항목에서 단 한번의 값을 요청합니다.