

LabVIEW™ 업그레이드 노트

이 업그레이드 노트에서는 Windows, Mac OS, Linux 용 LabVIEW 를 8.6 버전으로 업그레이드하는 과정, 업그레이드에서의 유의사항 및 8.6 버전의 새로운 기능에 대해 설명합니다. 잠재적인 호환성 문제에 대해 더 배우고자 하는 경우, 이전 버전의 LabVIEW 에서 저장한 VI 를 새 LabVIEW 버전에 로드하기 전에 이 업그레이드 노트를 읽으십시오. 새 LabVIEW 버전에 파일을 로드하기 전에, 이전 버전의 LabVIEW 에서 저장한 모든 LabVIEW 파일에 대한 백업용 복사본을 생성하는 것이 좋습니다.

LabVIEW 7.1 또는 이전 버전에서 LabVIEW 8.6 으로 업그레이드하는 경우, National Instruments 는 LabVIEW 7.1 과 LabVIEW 8.6 사이에서 개선, 변경 및 추가된 기능에 대한 정보는 업그레이드 노트 외에 다음의 문서를 참조할 것을 권장합니다.

- **LabVIEW 8.0 업그레이드 노트** – *업그레이드와 호환성 문제* 섹션과 *LabVIEW 8.0의 특징과 변경 내역* 섹션이 업그레이드 사용자에게 중요한 정보를 제공합니다. National Instruments 웹 사이트 (ni.com/info) 를 방문하여 정보 코드 upnote8 을 입력하면 *LabVIEW 8.0 업그레이드 노트* 를 확인할 수 있습니다.
- **LabVIEW 8.2 업그레이드 노트** – *업그레이드와 호환성 문제* 섹션과 *LabVIEW 8.2의 특징과 변경 내역* 섹션이 업그레이드 사용자에게 중요한 정보를 제공합니다. National Instruments 웹 사이트 (ni.com/info) 를 방문하여 정보 코드 upnote82 를 입력하면 *LabVIEW 8.2 업그레이드 노트* 를 확인할 수 있습니다.
- **LabVIEW 8.5 업그레이드 노트** – *업그레이드와 호환성 문제* 섹션과 *LabVIEW 8.5의 특징과 변경 내역* 섹션이 업그레이드 사용자에게 중요한 정보를 제공합니다. National Instruments 웹 사이트 (ni.com/info) 를 방문하여 정보 코드 upnote85 를 입력하면 *LabVIEW 8.5 업그레이드 노트* 를 확인할 수 있습니다.

LabVIEW 프로그래밍 개념, LabVIEW 사용에 대한 단계별 설명, LabVIEW VI, 함수, 팔레트, 메뉴, 도구, 프로퍼티, 메소드, 이벤트, 대화 상자 등에 대한 참조 정보와 LabVIEW 8.6의 특징에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오. 또한, *LabVIEW 도움말*은 National Instruments 가 제공하는 LabVIEW 문서 리소스를 나열합니다. **도움말** » **LabVIEW 도움말 검색**을 선택하여 *LabVIEW 도움말*에 접근합니다.

목차

LabVIEW 8.6 으로 업그레이드하기	2
VI 변환하기	3
모듈, 툴킷, 인스트루먼트 드라이버 업그레이드하기	4
추가적인 National Instruments 소프트웨어 업그레이드하기	5
이전 버전의 LabVIEW 에서 업그레이드하기	6
업그레이드와 호환성 문제.....	7
LabVIEW 8.5 에서 업그레이드하기.....	7
리모트 프론트패널 라이선스 업그레이드하기 (Windows).....	9
어플리케이션 빌더의 공유 구성요소.....	9
LabVIEW 8.2 에서 업그레이드하기.....	10
LabVIEW 8.0 에서 업그레이드하기.....	15
LabVIEW 7.x에서 업그레이드하기	19
LabVIEW 6.x에서 업그레이드하기	33
LabVIEW 5.x 또는 이전 버전에서 업그레이드하기.....	36
LabVIEW 8.6 의 특징과 변경 내역	36
LabVIEW 설치하기	36
LabVIEW 문서	37
새 예제 VI.....	37
블록다이어그램 개선사항	37
프론트패널 개선사항	38
환경 개선사항	39
새로운 또는 변경된 VI, 함수, 노드 개선사항	42
새로운 클래스, 프로퍼티, 메소드, 이벤트	46
블록다이어그램 자동 정리	46
빠른 탐색을 사용하여 객체 놀기.....	46
여러 객체의 프로퍼티 편집	46
LabVIEW 웹 서비스 (Windows, Base Package 에 포함 안됨).....	46
NI 배포된 시스템 관리자.....	47
LabVIEW MathScript 개선사항 (Base Package 에 포함 안됨).....	47
고정 소수점 수에 대해 오버플로우 관리하기	49
LLB 병합하기 (Base Package 에 포함 안됨).....	50

LabVIEW 8.6 으로 업그레이드하기

LabVIEW 의 이전 버전에서 업그레이드하는 경우, 먼저 이 *LabVIEW 8.6 으로 업그레이드하기* 섹션과 이 문서의 *업그레이드와 호환성 문제* 섹션에서 *LabVIEW x.x 에서 업그레이드하기* 섹션을 참조하십시오. 이 때 *x.x*는 업그레이드할 LabVIEW 의 버전입니다.

다음의 과정은 새 LabVIEW 버전을 업그레이드할 때 관련 작업을 완료하는 순서와 작업을 완료한 후에 읽을 문서에 대해 설명합니다. National Instruments 는 새 LabVIEW 버전으로 업그레이드하기 전에 이 문서와 *LabVIEW 릴리스 노트* 모두를 읽기를 권장합니다.

1. LabVIEW 를 설치하기 전에 모든 호환성 문제를 충분히 인식하고 있는지 확인하려면, 새 LabVIEW 버전을 설치하기 이전에 이 문서에서 다음 섹션을 참조하십시오:
 - **LabVIEW 8.6 으로 업그레이드하기** - 이 섹션은 툴킷과 모듈 설치 제거하기, 이전 버전의 LabVIEW 에서 환경 셋팅과 user.lib 파일 복사하기, VI 를 LabVIEW 8.6 으로 변환하기에 대해 설명합니다.

- **업그레이드 및 호환성 문제**—이 섹션은 이전 버전의 LabVIEW 에서 새 LabVIEW 버전으로 업그레이드하는 VI 에 영향을 미칠 수 있는 호환성 문제에 대해 설명합니다 . 특히 , 업그레이드하려는 LabVIEW 버전과 관련된 하위 섹션을 참조하십시오 .



노트 National Instruments 웹 사이트 ni.com/info 를 방문하고 정보 코드 `exqmex` 를 입력하여 일부 호환성 문제에 대해 VI 를 평가할 수 있는 테스트를 다운로드할 수 있습니다 .

- (옵션) **LabVIEW 8.6 의 특징과 변경 내역**—이 섹션은 LabVIEW 8.6 버전의 새 기능에 대한 간략하게 설명합니다 . 이 기능을 사용하는 방법에 대한 전체 설명은 *LabVIEW 도움말* 을 참조하십시오 . **도움말** > **LabVIEW 도움말 검색** 을 선택하여 *LabVIEW 도움말* 에 접근합니다 .
2. (옵션) 이전 버전의 LabVIEW 를 설치 제거합니다 .
 3. LabVIEW 업그레이드 버전을 설치하고 정품인증합니다 . LabVIEW 와 관련된 모든 태스크를 완료했는지 확인하려면 , *LabVIEW 릴리즈 노트* 의 다음 섹션을 참조합니다 :
 - *시스템 사양*
 - *LabVIEW 8.6 설치하기* 와 LabVIEW 를 설치하는 플랫폼에 해당하는 하위 섹션
 - *LabVIEW 애드온 설치하기* (LabVIEW 플랫폼 DVD 가 아닌 CD 를 통해 LabVIEW 툴킷 또는 모듈을 설치하는 경우)
 - **(Windows) LabVIEW 라이선스 정품인증** 과 모든 하위 섹션
 - (옵션) *하드웨어 설치 및 설정하기* 와 하드웨어를 설치하는 플랫폼에 해당하는 하위 섹션
 - *여기서부터 어디로 가나요 ?*
 4. 새로운 버전의 LabVIEW 에서 수정된 문제 , 새로운 버전의 LabVIEW 에서 알려진 문제점과 *LabVIEW 도움말* 에 설명되지 않은 문서 추가 사항에 대한 정보에 대해서는 *LabVIEW Readme* 를 참조하십시오 . *LabVIEW Readme* 를 참조하려면 , `labview\readme` 디렉토리를 탐색하고 `readme.html` 파일을 찾습니다 .
 5. 이전 버전의 LabVIEW 에서 환경 셋팅을 복사합니다 . 환경 셋팅 복사에 대한 정보는 이 문서의 *이전 버전의 LabVIEW 에서 환경 셋팅 복사하기* 섹션을 참조하십시오 .
 6. 이전 버전의 LabVIEW 에서 `user.lib` 파일을 복사합니다 . `user.lib` 파일 복사에 대한 추가적인 정보는 이 문서의 *이전 버전의 LabVIEW 에서 user.lib 파일 복사하기* 섹션을 참조하십시오 .
 7. VI 를 LabVIEW 8.6 으로 변환합니다 . 이전 버전의 LabVIEW 에서 저장한 VI 의 변환에 대한 추가적인 문서는 이 문서의 *VI 변환하기* 섹션을 참조하십시오 .

VI 변환하기

LabVIEW 6.0 이전 버전에서 저장한 VI 를 열려면 NI 영업부에 문의하여 사용자 코드를 LabVIEW 8.6 과 호환가능한 VI 포맷으로 업그레이드하는 변환 소프트웨어를 얻어야합니다 . LabVIEW 6.0 또는 이후 버전에서 저장된 VI 를 열면 LabVIEW 8.6 은 자동으로 VI 를 변환하고 컴파일합니다 . 변환되고 컴파일 된 VI 를 반드시 LabVIEW 8.6 으로 저장해야 합니다 . 저장하지 않으면 추가적인 메모리 리소스를 사용하는 변환 프로세스가 VI 에 접근할 때마다 수행됩니다 . 또한 , 리컴파일과 같은 저장되지 않은 변경사항을 가지고 있는 VI 는 실행 성능을 크게 떨어뜨릴 수도 있습니다 .



노트 LabVIEW 8.6 에서 저장한 VI 는 이전 버전의 LabVIEW 에서는 로드되지 않습니다 . VI 가 LabVIEW 8.5, 8.2 또는 8.0 에서 실행되도록 하려면 **파일** > **이전 버전으로 저장** 을 선택하여 VI 를 저장하십시오 . VI 를 변환한 후 LabVIEW 8.6 에서 저장하기 이전에 , LabVIEW 8.5, 8.2, 또는 8.0 에서 사용할 VI 의 백업 복사본을 남겨두십시오 .

컴퓨터에 메모리가 부족하여 모든 VI 를 한번에 변환하기 어려울 경우, VI 를 단계별로 나누어 변환하십시오. 변환하려는 VI 의 계층구조를 확인하고 가장 낮은 계층구조의 SubVI 부터 로딩하고 저장하기 시작합니다. 점차 상위 계층구조의 VI 까지 진행합니다. 마지막으로 최상위 레벨 VI 를 열고 변환합니다. 또한, 디렉토리 내에 위치한 VI 를 변환하려면 **도구»고급»매스컴 파일을** 선택할 수 있습니다. 그러나 매스 컴파일은 디렉토리 내의 VI 또는 LLB 를 알파벳 순서로 변환합니다. 변환 프로세스가 상위 레벨 VI 에 먼저 접근하게 되면, 매스 컴파일은 사용자가 상위 레벨 VI 를 먼저 열었을 때와 거의 동일한 양의 메모리를 필요로 합니다.

현재 사용하고 있는 메모리 양을 디스플레이하려면 **도움말» LabVIEW 정보**를 선택하여 메모리 사용량을 모니터링합니다.

모듈, 툴킷, 인스트루먼트 드라이버 업그레이드하기

이전 버전의 LabVIEW 에서 업그레이드하는 경우, 이전 버전의 LabVIEW 에서 설치한 모듈, 툴킷, 인스트루먼트 드라이버 모두와 현재 호환 가능한 버전을 설치해야 합니다. LabVIEW 플랫폼 DVD 는 LabVIEW 8.6 과 호환되는 대부분의 모듈과 툴킷을 포함합니다. LabVIEW 플랫폼 DVD 에 없는 모듈과 툴킷의 경우, 현재의 LabVIEW 버전과 호환 가능한 LabVIEW 모듈과 툴킷에 대한 정보는 National Instruments 웹 사이트 (ni.com/info) 에서 정보 코드 compat 를 입력하여 참조하십시오.

NI 모듈과 툴킷

다음 테이블은 사용자 OS 와 LabVIEW 애드온에 따라 LabVIEW 플랫폼 DVD 나 모듈 또는 툴킷 설치 CD 를 사용할 지 여부를 나열합니다.

OS	사용할 미디어	중요 정보
Windows	DVD	LabVIEW 8.6 및 LabVIEW 8.6 과 호환 가능한 모듈과 툴킷 버전을 설치할 때, LabVIEW 플랫폼 DVD 를 사용합니다. 추가적으로, 아직 구매하지 않은 모듈이나 툴킷의 평가판을 사용해볼 수 있습니다. LabVIEW 플랫폼 DVD 를 통해 이전 버전을 제거하거나 변경하지 않고 새 툴킷 버전을 LabVIEW 8.6 과 함께 설치할 수 있습니다. LabVIEW, 모듈, 툴킷을 설치하는 방법에 대한 정보는 <i>LabVIEW 릴리즈 노트</i> 를 참조하십시오.
Windows 에서 LabVIEW 플랫폼 DVD 가 모듈이나 툴킷을 포함하지 않는 경우	CD	모듈이나 툴킷을 구입할 때 받은 설치용 CD 를 사용합니다. 설치 CD 를 사용하기 전에, 설치하려는 모듈이나 툴킷이 호환 가능한 버전인지 확인합니다. 현재의 LabVIEW 버전과 호환 가능한 LabVIEW 모듈과 툴킷에 대한 정보는 National Instruments 웹 사이트 (ni.com/info) 에서 정보 코드 compat 를 입력하여 참조하십시오. 호환 가능한 모듈과 툴킷을 LabVIEW 8.6 디렉토리에 설치합니다. 이전 버전의 LabVIEW 에서 저장한 VI 를 모두 매스 컴파일합니다.
Mac/Linux		추가적인 정보는 이 문서의 <i>LabVIEW 매스 컴파일하기</i> 섹션을 참조하십시오.



노트 일부 툴킷 버전은 LabVIEW 8.6 과 호환되지 않습니다. 호환되지 않는 툴킷을 설치하면 툴킷이나 LabVIEW 의 일부 기능이 올바르게 동작할 수 없습니다. National Instruments 에서는 툴킷을 설치하기 전에 호환성을 확인할 것을 권장합니다. 현재의 LabVIEW 버전과 호환 가능한 LabVIEW 모듈과 툴킷에 대한 정보는 National Instruments

웹 사이트 (ni.com/info) 에서 정보 코드 compat 를 입력하여 참조하십시오 . 호환되지 않는 버전을 설치하여 LabVIEW 8.6 설치가 손상된 경우 , 우선 **프로그램 추가 / 제거** 를 사용하여 툴킷을 제거한 후에 LabVIEW 설치를 복구합니다 .

인스트루먼트 드라이버

사용하려는 인스트루먼트 하드웨어를 모두 컨트롤하고 통신하려면 현재 버전에서 인스트루먼트 드라이버를 설치해야 합니다 . 이전 버전의 LabVIEW 에서 인스트루먼트 드라이버를 설치한 경우 , 다음 방법 중 하나를 사용하여 LabVIEW 8.6 에서 해당 인스트루먼트 드라이버를 재설치해야 합니다 .

- **NI 모듈러 인스트루먼트 드라이버** – NI 디바이스 드라이버 DVD 또는 CD 를 사용하여 NI 모듈러 인스트루먼트 드라이버를 설치합니다 .
- **플러그 앤 플레이 인스트루먼트 드라이버** – NI 인스트루먼트 드라이버 검색기를 사용하여 LabVIEW 개발 환경 내에서 LabVIEW 플러그 앤 플레이 인스트루먼트 드라이버를 검색하고 설치합니다 .
- **IVI 드라이버 및 비인증 인스트루먼트 드라이버** – National Instruments 웹 사이트에서 [인스트루먼트 드라이버 네트워크](#) 를 사용하여 IVI 드라이버 및 비인증 인스트루먼트 드라이버를 검색하고 설치합니다 .



노트 NI 인스트루먼트 드라이버 검색기를 사용하여 인스트루먼트 드라이버를 재설치하는 경우 , National Instruments 는 labview\instr.lib 디렉토리를 매스 컴파일할 것을 권장합니다 .

타사 애드온

타사 LabVIEW 애드온 제조업체를 연락하여 현재 사용자 OS 시스템에서 LabVIEW 8.6 과 호환 가능한지 확인합니다 . 애드온과 관련된 모든 VI 를 반드시 매스 컴파일하십시오 .

추가적인 정보는 이 문서의 *LabVIEW 매스 컴파일하기* 섹션을 참조하십시오 .

LabVIEW 매스 컴파일하기

LabVIEW 의 이전 버전에서 저장한 VI 를 열게 되면 , LabVIEW 는 자동으로 VI 를 변환하고 그리고 컴파일합니다 . 이 VI 를 LabVIEW 현재 버전으로 저장하십시오 . 그렇지 않으면 VI 를 열 때마다 추가적인 메모리 리소스를 사용하는 변환 프로세스가 수행됩니다 . LabVIEW 플랫폼 DVD 에 없는 LabVIEW 모듈과 툴킷을 설치하거나 타사 애드온을 설치하는 경우 , National Instruments 는 이 같은 모듈과 툴킷 또는 타사 애드온이 설치한 VI 를 모두 매스 컴파일할 것을 권장합니다 .

VI 매스 컴파일에 대한 추가적인 정보는 **기본 >> VI 와 SubVI 생성하기 >> 사용법 >> VI 저장하기 >> VI 매스 컴파일하기** 를 참조하십시오 .

추가적인 National Instruments 소프트웨어 업그레이드하기

LabVIEW 8.x 에서는 NI TestStand 3.5 또는 이후 버전을 사용해야 합니다 . Upgrade Advisor 에 접근하고 NI TestStand 3.5 또는 이후 버전을 구매하려면 , National Instruments 웹 사이트 ni.com/info 에서 정보 코드 exd8yy 를 입력하십시오 .

NI TestStand 3.5 및 NI TestStand 4.0 은 다음의 LabVIEW 8.6 익스프레스 VI 를 설정하려하면 에러를 반환합니다 :

- 데이터 스토리지 닫기
- 수식
- 프로퍼티 얻기 (복수)
- 데이터 스토리지 열기
- 데이터 읽기

- 프로퍼티 설정 (복수)
- 스펙트럼 측정
- 데이터 쓰기

National Instruments 웹 사이트 (ni.com/info) 에서 정보 코드 rdtf10 을 입력하면 에러에 대한 추가적인 정보를 얻을 수 있습니다 .



노트 TestStand 4.1 및 이후 버전에서는 이 문제를 해결하였습니다 .

LabVIEW 와 TestStand 문제에 대한 보다 자세한 정보는 NI TestStand CD 와 <TestStand>\Doc 디렉토리에서 사용하는 TestStand 의 버전에 대한 Readme.html 파일을 참조하십시오 .

LabVIEW 8.x에서는 NI Spy 2.3 또는 이후 버전을 사용해야 합니다 . NI Spy 2.5 는 National Instruments 디바이스 드라이버 CD 에서 설치할 수 있습니다 .

LabVIEW 8.6 은 Measurement Studio 8.0 또는 이후 버전을 지원합니다 . Upgrade Advisor 에 접근하고 Measurement Studio 8.0 또는 이후 버전을 주문하려면 , National Instruments 웹 사이트 ni.com/info 에서 정보 코드 exd8yy 를 입력하십시오 .

이전 버전의 LabVIEW 에서 업그레이드하기

이전 버전의 LabVIEW 를 제거하지 않고 LabVIEW 8.6 을 설치할 수 있습니다 . LabVIEW 버전 은 구성요소를 공유하지만 , LabVIEW 를 새로운 버전으로 업그레이드해도 새로운 버전은 다른 디렉토리에 설치되므로 , 설치되어 있는 이전 버전의 LabVIEW 에는 영향을 주지 않습니다 . LabVIEW 5.x 및 이전 버전은 labview 디렉토리에 설치됩니다 . LabVIEW 6.0 및 이후 버전은 labview x.x 디렉토리에 설치됩니다 . 이 때 x.x 는 버전 번호입니다 .

현재 버전의 LabVIEW 대체하기

현재 버전의 LabVIEW 를 대체하려면 , 현재 버전을 제거하고 , LabVIEW 8.6 설치 프로그램을 실행한 후 , 설치 디렉토리를 이전 버전의 LabVIEW 가 설치된 labview 디렉토리로 설정하십시오 .

(Windows) 또한 프로그램 제어판의 프로그램 추가 / 제거를 사용하여 현재 버전의 LabVIEW 를 LabVIEW 8.6 으로 대체하여 현재 버전을 제거할 수 있습니다 . 설치 제거 프로그램은 labview 디렉토리에 생성한 파일은 삭제하지 않습니다 .



노트 LabVIEW 를 제거하거나 다시 설치할 때 , LabVIEW 는 .lib 파일에 저장한 모든 VI 와 컨트롤을 포함하여 vi.lib 디렉토리의 .lib 파일을 제거합니다 . 사용자 컨트롤과 VI 를 **컨트롤과 함수** 팔레트에 추가하려면 user.lib 디렉토리에 사용자 컨트롤과 VI 를 저장하십시오 .

이전 버전의 LabVIEW 에서 환경 셋팅 복사하기

이전 버전의 LabVIEW 의 환경 셋팅을 사용하려면 이전 버전이 설치된 labview 디렉토리에서 LabVIEW 환경 파일을 복사하십시오 .



주의 LabVIEW 8.6 의 환경 파일을 이전 버전의 환경 파일로 변경하면 , 이전 버전 이후에 LabVIEW 에 추가된 환경 설정을 덮어쓰게됩니다 .

LabVIEW 8.6 을 설치한 후 , LabVIEW 환경 파일을 LabVIEW 8.6 디렉토리로 복사합니다 .

(Windows) LabVIEW 는 환경을 labview 디렉토리의 labview.ini 파일에 저장합니다 .

(Mac OS) LabVIEW 는 환경을 홈 디렉토리 Library:Preferences 폴더의 LabVIEW Preferences 파일에 저장합니다 .

(Linux) LabVIEW 는 환경을 홈 디렉토리의 .labviewrc 파일에 저장합니다 .

이전 버전의 LabVIEW 에서 user.lib 파일 복사하기

이전 버전의 LabVIEW 의 user.lib 디렉토리에 있는 파일을 사용하려면 , 이전 버전이 설치된 labview 디렉토리에서 파일을 복사하십시오 . LabVIEW 8.6 을 설치한 후 , 이 파일을 LabVIEW 8.6 디렉토리의 user.lib 디렉토리에 복사하십시오 .

업그레이드와 호환성 문제

특정 LabVIEW 버전에 대한 업그레이드와 호환성과 관련된 사항은 다음 섹션을 참조하십시오 .

새로운 LabVIEW 버전에서의 유의사항 , 추가적인 호환성 문제 , LabVIEW 8.6 의 최신 추가 기능에 대한 자세한 내용은 labview 디렉토리의 readme.html 파일을 참조하십시오 .

LabVIEW 8.5 에서 업그레이드하기

LabVIEW 8.5 에서 LabVIEW 8.6 로 업그레이드할 때 다음과 같은 호환성 문제가 발생할 수 있습니다 .



노트 또한 National Instruments 웹 사이트 ni.com/info 를 방문하여 정보 코드 upnote85 를 입력하면 , LabVIEW 8.5.x에서 업그레이드할 때 발생할 수 있는 유의사항에 대한 보다 자세한 정보를 참조할 수 있습니다 .

지원하는 플랫폼

LabVIEW 8.6 의 지원 플랫폼에서 다음과 같은 부분이 변경되었습니다 :

- LabVIEW 8.6 은 PowerPC 프로세서를 사용하는 Macintosh 컴퓨터를 지원하지 않습니다 .

시스템 사양

(Windows) LabVIEW 8.6 은 LabVIEW 설치시 최소 1.6 GB 의 빈 디스크 공간이 필요합니다 .

(Mac OS) LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 262 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 828 MB 의 빈 디스크 공간이 필요합니다 .

(Linux) LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 365 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 651 MB 의 빈 디스크 공간이 필요합니다 .

인쇄된 문서

LabVIEW 8.6 에서 다음의 문서는 변경되지 않았습니다 . 따라서 LabVIEW 8.6 에서 변경한 사항이 문서 내용에 반영되지 않을 수 있습니다 .

- LabVIEW 시작하기 매뉴얼
- (LabVIEW 8.2, 8.5, 8.6) LabVIEW 기본 매뉴얼



노트 LabVIEW 기본 매뉴얼은 LabVIEW 도움말의 기본 모음의 일부분입니다 . 따라서 , 업데이트된 내용에 대한 정보는 LabVIEW 도움말의 목차 탭에 있는 기본 모음을 참조하십시오 .

VI 와 함수 동작 변경 내역

LabVIEW 8.6 에서 VI 와 함수의 동작이 다음과 같이 변경되었습니다 .

리포트 생성 VI

LabVIEW 클래스를 사용하여 리포트 생성 VI 가 재작성되었습니다 . 리포트 입력 컨트롤과 리포트 출력 인디케이터가 참조 번호 데이터 타입에서 LabVIEW 클래스 데이터 타입으로 변경되었습니다 . 타입 정의 참조 번호에서 마우스 오른쪽 버튼을 클릭하여 상수 , 컨트롤 , 인디케이터를 생성하지 않은 경우 , LabVIEW 가 이 객체를 업데이트할 수 없으므로 관련 VI 가 제대로 동작하지 않을 수도 있습니다 . 리포트 입력과 리포트 출력 파라미터의 이전 참조 번호 데이터 타입을 호출하는 모든 참조에 의한 호출 노드는 예상대로 동작하지 않습니다 .

타겟에서 실행되도록 리포트 생성 VI 를 사용하여 HTML 리포트를 생성하는 경우, 리포트를 생성할 때 해당 타겟을 참조하는지 반드시 확인하십시오. 호스트 컴퓨터에 HTML 리포트를 생성하고, 이후 타겟을 참조하지 않고 배포하면 VI 는 깨진 것으로 나타나며 실행되지 않습니다.

외부 코드 (DLLs 및 CINS)

메모리 관리자 관련 함수는 오직 하나의 메모리 영역, DS (data space) 를 포함합니다.

LabVIEW 메모리를 관리하는 C 또는 C++ CIN 또는 DLL 과 같이 동작하면, AZ (어플리케이션 영역) 메모리 함수에 대한 모든 참조를 DS 와 같은 함수의 것으로 대체합니다.

기타 VI 와 함수 동작 변경 사항

LabVIEW 8.6 의 VI 와 함수 동작에서 다음과 같이 변경되었습니다 :

- 수학과 신호 처리 VI 가 재호출이 아닌 VI 에서 재호출 VI 로 변경되었습니다. 이처럼 변경 되었으므로, 인스턴스 사이에 복제를 공유하도록 설정된 재호출 VI 에서 이 VI 를 호출해서는 안됩니다. 인스턴스 사이에 복제를 공유하도록 설정된 VI 에서 호출할 수 없는 VI 에 대한 자세한 정보는 National Instruments 웹 사이트인 ni.com/info 를 방문하고 정보 코드 `exrehi` 를 입력하여 참조하십시오.
- 이제 LabVIEW 는 단일 프로세스 공유 변수가 특정 타겟이 되도록 합니다. 단일 프로세스 공유 변수가 절대적이 되도록 설정할 수 없습니다.
- 빈 경로를 라이브러리 함수 호출 노드의 **경로 입력** 입력에 연결하면, LabVIEW 는 더 이상 에러를 반환하지 않습니다.

삭제된 VI 와 함수

LabVIEW 8.6 은 다음의 VI 와 함수를 지원하지 않습니다 :

- **세마포어 생성**—대신 (세마포어 참조 얻기) VI 를 사용합니다. (세마포어 참조 얻기) VI 는 (세마포어 생성) VI 와는 다릅니다. (세마포어 생성) VI 를 여러 번 사용하여 동일한 이름으로 다수의 세마포어를 생성하는 경우, LabVIEW 는 해당 세마포어에 대한 단일 참조의 복사본 여러 개를 생성합니다. 하지만, (세마포어 참조 얻기) VI 를 사용하여 동일한 세마포어에 대한 참조를 여러 개 얻는 경우, 각각의 고유한 참조 번호가 생성됩니다. LabVIEW 가 (세마포어 참조 얻기) VI 를 사용하도록 기존 VI 를 자동으로 변환하지 않기 때문에, 이전 LabVIEW 버전에서 저장한 VI 를 사용자가 직접 업데이트해야 합니다.
- **세마포어 삭제**—대신 (세마포어 참조 해제) VI 를 사용합니다. (세마포어 참조 해제) VI 는 (세마포어 삭제) VI 와는 다릅니다. (세마포어 삭제) VI 를 사용하여 세마포어를 제거하는 경우, LabVIEW 는 해당 세마포어에 대한 다른 참조의 복사본도 모두 제거합니다. 하지만, (세마포어 참조 해제) VI 를 사용하여 세마포어에 대한 참조를 해제하는 경우, 해당 세마포어에 대한 다른 참조는 여전히 유효하며 LabVIEW 는 다른 참조가 전혀 없는 경우에만 그 세마포어를 제거합니다. LabVIEW 가 (세마포어 참조 해제) VI 를 사용하도록 기존 VI 를 자동으로 변환하지 않기 때문에, 이전 LabVIEW 버전에서 저장한 VI 를 사용자가 직접 업데이트해야 합니다.



노트 예상치 못한 결과가 나오지 않도록, (세마포어 생성) VI 또는 (세마포어 삭제) VI 와 함께 사용하는 참조를 (세마포어 참조 얻기) VI 및 (세마포어 참조 해제) VI 에 전달하지 마십시오. 반대의 경우에도 마찬가지입니다.

- **Xmath 스크립트 노트**—대신 MathScript 노트를 사용합니다. MathScript 구문이 Xmath 구문과 다르기 때문에, MathScript 노트에서 작동할 수 있도록 현재 구문을 변경해야 합니다.

프로퍼티, 메소드, 이벤트 동작 변경 내역

LabVIEW 8.6 에서 프로퍼티, 메소드, 이벤트의 동작이 다음과 같이 변경되었습니다 :

- 장면 그래프 디스플레이 클래스의 카메라 컨트롤러 : 타입 프로퍼티는 **지향 (Oriented)** 값을 포함합니다.
- 장면 윈도우 클래스의 카메라 컨트롤러 : 타입 프로퍼티는 **지향 (Oriented)** 값을 포함합니다.

삭제된 프로퍼티, 메소드, 이벤트

LabVIEW 8.6 및 이후 버전에서는 다음의 프로퍼티, 메소드, 이벤트를 지원하지 않습니다:

- VI 클래스의 컨트롤 값 : 설정 (패턴화) 메소드 . 대신 컨트롤 값 : 설정 메소드를 사용합니다 .
- VI 클래스의 컨트롤 값 : 모두 얻기 (패턴화) 메소드 . 대신 컨트롤 값 : 모두 얻기 메소드를 사용합니다 .
- VI 클래스의 컨트롤 값 : 얻기 (패턴화) 메소드 . 대신 컨트롤 값 : 얻기 메소드를 사용합니다 .
- VI 프로퍼티 (ActiveX) 클래스의 VIModificationBitSet 프로퍼티 . 대신 VIModificationBitSet2 프로퍼티를 사용합니다 .
- VI 프로퍼티 클래스의 수정 :VI 수정 프로퍼티 . 대신 새 수정 :VI 수정 프로퍼티를 사용합니다 . LabVIEW 8.5 및 이전 버전에서 , 수정 :VI 수정 프로퍼티는 32- 비트 값을 반환합니다 . LabVIEW 8.6 에서 , 새 수정 :VI 수정 프로퍼티는 64 비트 값을 반환합니다 .

이름이 변경된 프로퍼티, 메소드, 이벤트

LabVIEW 8.6 및 이후 버전에서 다음의 프로퍼티, 메소드, 이벤트의 이름이 변경되었습니다 :

클래스	LabVIEW 8.5 이름	LabVIEW 8.6 이름	타입
G 객체	경계 : 높이	경계 :영역 높이	프로퍼티
G 객체	경계 : 폭	경계 :영역 폭	프로퍼티
프로젝트 아이템	디스크로부터 연결 끊기	자동 적용 정지	메소드
트리 컨트롤	확장 / 축소 기호 : 하위 레벨 0 에서 보이기	확장 / 축소 기호 : 루트에서 기호 보이기	프로퍼티
VI	컨트롤 값 : 설정 (배리언트)	컨트롤 값 : 설정	메소드
VI	컨트롤 값 : 얻기 (배리언트)	컨트롤 값 : 얻기	메소드
VI	컨트롤 값 : 모두 얻기 (배리언트)	컨트롤 값 : 모두 얻기	메소드

리모트 프론트패널 라이선스 업그레이드하기 (Windows)

LabVIEW를 사용하여 클라이언트가 원격에서 어플리케이션이나 프론트패널을 보도록 허용할 수 있습니다 . LabVIEW 는 5, 20, 50 명이 또는 무제한으로 클라이언트가 동시에 리모트 프론트패널에 연결할 수 있는 라이선스를 지원합니다 . 서버 컴퓨터에 오직 하나의 라이선스만 가질 수 있습니다 . 허용하려는 클라이언트 수를 지원할만큼 충분한 수의 연결을 인증할 수 있는 라이선스를 구입합니다 . LabVIEW 8.5.1 또는 이전 버전에서 리모트 프론트 패널 라이선스를 업그레이드했다면 , NI 라이선스 관리자에서 기존 시리얼 번호를 사용하여 같은 조건의 새 라이선스를 인증받아야 합니다 .

어플리케이션 빌더의 공유 구성요소

LabVIEW 는 로컬 컴퓨터의 모든 버전의 LabVIEW 와 공유되는 공유 라이브러리를 빌드하기 위한 구성 요소를 설치합니다 . LabVIEW 8.6 을 설치한 후 이전 버전의 LabVIEW 를 설치하면 , 이전 버전이 공유 구성요소를 대체합니다 . 이후 가장 최신 버전의 LabVIEW 에서 공유 라이브러리를 빌드하려하면 , 공유 구성요소에는 LabVIEW 8.6 이 사용하는 기능이 없으므로 에러가 발생합니다 . 이 문제를 해결하려면 , LabVIEW 8.6 을 다시 설치하십시오 .

LabVIEW 8.2 에서 업그레이드하기

LabVIEW 8.2 에서 LabVIEW 8.6 으로 업그레이드할 때 다음과 같은 호환성 문제가 발생할 수 있습니다. 업그레이드시 발생할 수 있는 다른 문제점에 대해서는 이 문서의 [LabVIEW 8.5 에서 업그레이드하기](#) 섹션을 참조하십시오 .



노트 또한 National Instruments 웹 사이트 ni.com/info 를 방문하여 정보 코드 ex5d8c 를 입력하면 , LabVIEW 8.2 에서 업그레이드할 때 발생할 수 있는 유의사항에 대한 보다 자세한 정보를 참조할 수 있습니다 .

지원하는 플랫폼

LabVIEW 8.5 및 이후 버전의 지원하는 플랫폼에서 다음과 같이 변경되었습니다 :

- LabVIEW 8.5 및 이후 버전은 Windows Vista x86 및 x64 를 지원합니다 .
- LabVIEW 8.5 및 이후 버전은 Intel 과 PowerPC 프로세서가 내장된 Macintosh 컴퓨터를 지원합니다 .

시스템 사양

(Windows) LabVIEW 8.5 는 LabVIEW 설치시 최소 1.2 GB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 LabVIEW 설치시 최소 1.6 GB 의 빈 디스크 공간이 필요합니다 .

(Mac OS) LabVIEW 8.5 는 LabVIEW 최소 설치시 최소 502 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 734 MB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 262 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 828 MB 의 빈 디스크 공간이 필요합니다 .

(Linux) LabVIEW 8.5 는 LabVIEW 최소 설치시 최소 450 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 640 MB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 365 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 651 MB 의 빈 디스크 공간이 필요합니다 .

Windows Vista 호환성 문제

LabVIEW 8.5 및 이후 버전은 다음의 기능상의 변경과 함께 32 비트와 64 비트 시스템에서 Windows Vista 를 지원합니다 .

특정주소에서 읽기와 특정주소에서 쓰기 VI 는 시스템의 모든 입 / 출력 포트에 대한 읽기 / 쓰기 권한을 허용하므로 , Vista OS 에서는 보안상의 이유로 장려되지 않습니다 . 따라서 이 VI 는 함수 팔레트에 없습니다 .

- **(Windows Vista x86)** VI 구성요소는 성공적으로 설치되지만 , Windows Defender 로그에서 "unsigned" 로 나타납니다 . 이 VI 는 정상적으로 실행됩니다 .
- **(Windows Vista x64)** 이 VI 는 에러 -4850 을 반환합니다 .

VI 와 함수 동작 변경 내역

LabVIEW 8.5 및 이후 버전에서 VI 와 함수의 동작이 다음과 같이 변경되었습니다 .

분석 VI 와 함수에서의 개선사항

각 LabVIEW 버전에서 , National Instruments 는 LabVIEW 와 C 함수에만 국한되지 않고 많은 알고리즘을 개선했습니다 . 또한 National Instruments 는 LabVIEW 가 최신 컴파일러를 사용하도록 업그레이드합니다 . 컴퓨터 하드웨어와 소프트웨어에서의 변경 사항과 이러한 개선 사항으로 인해 LabVIEW 8.2 및 이전 버전과 LabVIEW 8.5 및 이후 버전 사이에서 서로 다른 숫자형 결과가 나올 수 있습니다 . 배정도 부동소수를 비교하면 , 1E-16 에서 작은 차이를 발견할 수 있습니다 . National Instruments 웹 사이트 (ni.com/info) 에서 정보 코드 exiigr 를 입력하면 부동소수 비교에 대한 자세한 정보를 참조할 수 있습니다 .

수학 VI

LabVIEW 8.5 및 이후 버전에서 **수학 VI**가 다음과 같이 변경되었습니다 :

- **f(x)의 모든 제로 찾기** - f(x)의 모든 제로 (Zeroes) 찾기 VI에서 f(x)의 모든 제로 (Zeros) 찾기 VI로 변경되었습니다 .
- **f(x)의 제로와 극값** - f(x)의 제로 (Zeroes)와 극값 VI에서 f(x)의 제로 (Zeros)와 극값 VI로 변경되었습니다 .

숫자형 함수

LabVIEW 8.5 및 이후 버전에서 **숫자형** 함수가 다음과 같이 변경되었습니다 :

- **올림** 함수 - 올림 (Round To +Infinity) 함수가 올림 (Round Toward +Infinity) 함수로 변경되었습니다 .
- **버림** 함수 - 버림 (Round To -Infinity) 함수가 버림 (Round Toward -Infinity) 함수로 변경되었습니다 .

신호 처리 VI

(변이 측정) VI에서, **프리슛** 출력이 **변이 이전**으로 변경되었습니다 . 이 출력 또한 64 비트 배정도 부동소수 숫자형 데이터 타입에서 클러스터 데이터 타입으로 변경되었습니다 . **오버슛** 출력이 **변이 이후**로 변경되었습니다 . 이 출력 또한 64 비트 배정도 부동소수 숫자형 데이터 타입에서 클러스터 데이터 타입으로 변경되었습니다 .

쌍곡형 함수

LabVIEW 8.5 및 이후 버전에서 쌍곡형 함수가 다음과 같이 변경되었습니다 :

- 역 쌍곡형 코사인 함수는 입력값이 함수 범위밖의 실수일 때 NaN를 반환합니다 .
- 역 쌍곡형 시컨트 함수는 입력값이 함수 범위밖의 실수일 때 NaN를 반환합니다 .

라이브러리 & 실행 파일 VI와 함수

라이브러리 함수 호출 노드에서 파스칼 문자열 포인터를 설정할 때 , 블록 다이어그램에서 값을 반드시 문자열 입력에 연결해야 합니다 . C 문자열 포인터를 설정할 때 , 값을 문자열 입력에 연결하거나 **라이브러리 함수 호출** 대화 상자의 **파라미터** 탭의 **최소 크기** 폴다운 메뉴에서 문자열 크기를 지정해야 합니다 . 문자열의 값을 지정해야 VI를 실행할 수 있습니다 .

VI 참조 열기 함수

LabVIEW 8.0 및 8.2에서 , **VI 경로** 입력의 VI 이름이 타겟의 메모리상의 VI 이름과 일치하는 경우 , LabVIEW는 메모리상의 VI에 대한 참조를 반환하고 **VI 경로** 입력에서 지정된 VI를 로드하지 않습니다 . LabVIEW 8.5 및 이후 버전에서 , **VI 경로** 입력의 VI 이름이 타겟 메모리 상의 VI 이름과 일치하지만 경로가 다른 경우 , (VI 참조 열기) 함수는 에러를 반환합니다 .

64 비트와 배정도 숫자형 데이터 타입을 지원하는 다형성 VI

배정도 숫자형 및 64 비트 정수 타입 모두를 지원하는 다형성 VI의 터미널에 연결한 경우 , LabVIEW는 확장형 숫자형 데이터를 배정도 숫자형 데이터로 강제 변환합니다 . 이 강제 변환은 원래 데이터의 일부 소수 부분을 보존합니다 .

기타 VI와 함수 동작 변경 사항

LabVIEW 8.5 및 이후 버전의 VI와 함수 동작이 다음과 같이 부수적으로 변경되었습니다 :

- (Instr 속성 얻기) VI와 (Instr 속성 설정) VI는 더이상 LabVIEW와 함께 배송되지 않습니다 . 어플리케이션에서 이 VI 중 하나를 사용하는 경우 , 이 VI를 동일한 기능에 대한 **VISA 고급** 팔레트의 프로퍼티 노드로 대체하십시오 .
- (순환적 파일 리스트) VI의 **모든 폴더** 파라미터는 폴더 바로 가기를 포함할 수 있지만 , 순환하지 않습니다 .

프로퍼티, 메소드, 이벤트 동작 변경 내역

LabVIEW 8.5 및 이후 버전에서 프로퍼티, 메소드, 이벤트의 동작이 다음과 같이 변경되었습니다 :

- VI 가 실행 중일때 컨트롤 클래스의 데이터 연결 : 경로 프로퍼티는 읽기 / 쓰기이며 설정 가능합니다 . 이 프로퍼티를 쓰려면 , 쓰기 전에 컨트롤을 NI Publish-Subscribe-Protocol URL 에 연결해야 합니다 .
- 어플리케이션 클래스의 타겟 : CPU 프로퍼티는 값 AMD/Intel x64 를 포함합니다 .
- 어플리케이션 클래스의 타겟 : OS 프로퍼티는 값 Windows x64 및 Linux x64 를 포함합니다 .
- 트리의 열 헤더안에 있는 포인트를 연결하면 트리 컨트롤 클래스의 포인트를 행과 열로 메소드가 TREE_COLUMN_HEADERS 태그를 반환합니다 .
- LabVIEW 클래스 : 생성 메소드는 이름 입력을 포함합니다 . **이름** 입력을 연결하지 않으면 , LabVIEW 는 실행시 클래스의 이름을 입력하도록 요청합니다 .
- 컨트롤 값 : 읽기 (배리언트) , 컨트롤 값 : 읽기 (패턴화) , 컨트롤 값 : 설정 (배리언트) 및 컨트롤 값 : 설정 (패턴화) 메소드가 컨트롤 검색시 더이상 앞과 뒤의 화이트스페이스를 제거하지 않습니다 .

삭제된 프로퍼티, 메소드, 이벤트

LabVIEW 8.5 및 이후 버전에서는 다음의 프로퍼티, 메소드, 이벤트를 지원하지 않습니다 :

- LV 클래스 라이브러리 클래스의 기본 인스턴스 프로퍼티 . 대신 LV 클래스 기본값 읽기 VI 를 사용하십시오 .
- 장면 객체 클래스의 기하 프로퍼티 . 대신 그리기 가능 프로퍼티를 사용하십시오 .
- 그래프 차트 클래스의 눈금 색 프로퍼티 . 대신 그래픽 스케일 클래스의 눈금 색 프로퍼티를 사용하십시오 .
- 그래프 차트 클래스의 눈금 색 : X 색 프로퍼티 . 대신 그래픽 스케일 클래스의 눈금 색 : 주요 색 및 눈금 색 : 보조 색을 사용하십시오 .
- 그래프 차트 클래스의 눈금 색 : Y 색 프로퍼티 . 대신 그래픽 스케일 클래스의 눈금 색 : 주요 색 및 눈금 색 : 보조 색을 사용하십시오 .
- 웨이브폼 차트 클래스의 범례 : 플롯 보이기 프로퍼티 . 대신 범례 : 행의 개수 프로퍼티를 사용하십시오 .
- 웨이브폼 그래프 클래스의 범례 : 플롯 보이기 프로퍼티 . 대신 범례 : 행의 개수 프로퍼티를 사용하십시오 .
- 리스트박스 클래스의 픽셀 폭 프로퍼티 . 대신 경계 : 영역 폭 프로퍼티를 사용하십시오 .
- 그림 클래스의 스크롤 막대 보이기 프로퍼티 . 대신 수평 스크롤 막대 보이기 및 수직 스크롤 막대 보이기 프로퍼티를 사용하십시오 .
- 장면 객체 클래스의 기하 설정 메소드 . 대신 그리기 가능 설정 메소드를 사용하십시오 .
- 어플리케이션 클래스의 장면 : 기하 : 새 메시 메소드 . 대신 장면 : 그리기 가능 : 기하 : 새 메시 메소드를 사용하십시오 .
- 컨트롤 클래스의 끝기 시작 이벤트 . 대신 해당 컨트롤 클래스의 끝기 시작 이벤트를 사용하십시오 .
- 컨트롤 클래스의 끝기 시작 ? 이벤트 . 대신 해당 컨트롤 클래스의 끝기 시작 ? 이벤트를 사용하십시오 .

이름이 변경된 프로퍼티, 메소드, 이벤트

LabVIEW 8.5 및 이후 버전에서 다음의 프로퍼티, 메소드, 이벤트의 이름이 변경되었습니다 :

클래스	LabVIEW 8.2 이름	LabVIEW 8.5 이름	타입
절대 시간, 숫자형	데이터 범위	데이터 입력 제한	프로퍼티
절대 시간, 숫자형	데이터 범위 : 증가	데이터 입력 제한 : 증가	프로퍼티
절대 시간, 숫자형	데이터 범위 : 최대값	데이터 입력 제한 : 최대	프로퍼티
절대 시간, 숫자형	데이터 범위 : 최소값	데이터 입력 제한 : 최소	프로퍼티
절대 시간, 숫자형	범위 이탈시 동작	리미트 밖의 값에 응답	프로퍼티
절대 시간, 숫자형	범위 이탈시 동작 : 증가	리미트 밖의 값에 응답 : 증가	프로퍼티
절대 시간, 숫자형	범위 이탈시 동작 : 최대값	리미트 밖의 값에 응답 : 최대	프로퍼티
절대 시간, 숫자형	범위 이탈시 동작 : 최소값	리미트 밖의 값에 응답 : 최소	프로퍼티
어플리케이션	라이브러리 : 프로젝트 라이브러리 파일 버전 얻기	라이브러리 : LabVIEW 파일 버전 얻기	메소드
어플리케이션	장면 : 기하 : 새 상자형	장면 : 그리기 가능 : 기하 : 새 상자형	메소드
어플리케이션	장면 : 기하 : 새 원뿔형	장면 : 그리기 가능 : 기하 : 새 원뿔형	메소드
어플리케이션	장면 : 기하 : 새 원기둥형	장면 : 그리기 가능 : 기하 : 새 원기둥형	메소드
어플리케이션	장면 : 기하 : 새 높이 필드	장면 : 그리기 가능 : 기하 : 새 높이 필드	메소드
어플리케이션	장면 : 기하 : 새 메시	장면 : 그리기 가능 : 기하 : 새 메시	메소드
어플리케이션	장면 : 기하 : 새 구형	장면 : 그리기 가능 : 기하 : 새 구형	메소드
어플리케이션 (ActiveX)	LibraryGetProjectLibFileVersion	LibraryGetFileLVVersion	메소드
디지털, 숫자형 텍스트, 스케일	포맷 & 정밀도	디스플레이 포맷	프로퍼티
디지털, 숫자형 텍스트, 스케일	포맷 & 정밀도 : 포맷	디스플레이 포맷 : 포맷	프로퍼티
디지털, 숫자형 텍스트, 스케일	포맷 & 정밀도 : 정밀도	디스플레이 포맷 : 정밀도	프로퍼티
디지털 테이블	열 헤더 보이기	신호 번호 보이기	프로퍼티
디지털 테이블	행 헤더 보이기	변환 보이기	프로퍼티

클래스	LabVIEW 8.2 이름	LabVIEW 8.5 이름	타입
장면 그래프 디스플레이와 장면 윈도우	색 삭제	배경색	프로퍼티
장면 객체	기하 설정	그리기 가능 설정	메소드
VI	커넥터 팬	커넥터 팬 : 설정	프로퍼티

LabVIEW MathScript 동작 변경 내역 (Windows, Base Package 에 포함 안됨)

LabVIEW 8.5 및 이후 버전에서 LabVIEW MathScript 가 다음과 같이 변경되었습니다 :

- 다음의 MathScript 함수를 사용하여 검색 경로 리스트 또는 작업 중인 디렉토리에서 변경한 경우, 이 변경은 함수를 호출하는 **LabVIEW MathScript 윈도우** 또는 MathScript 노트의 현재 인스턴스에만 적용됩니다 .

- addpath
- cd
- path
- rmpath

사용자가 **LabVIEW MathScript 윈도우**를 닫거나 또는 MathScript 노트를 포함하는 VI 가 실행을 중지할 때 LabVIEW 는 검색 경로 리스트와 작업 중인 디렉토리를 기본으로 리셋 합니다 .

- qz 함수의 구문이 $[q, z, \alpha, \beta, \text{evec}] = \text{qz}(a, b)$ 에서 $[S, T, Q, Z, R, L] = \text{qz}(A, B, \text{type})$ 로 변경되었습니다 .

LabVIEW 클래스 아이콘

LabVIEW 8.2 에서 LabVIEW 클래스를 생성하고 블록 다이어그램에 클래스 컨트롤이나 인디케이터를 놓을 때 생성한 아이콘을 디스플레이하려면, 클래스 마스크가 클래스 아이콘을 가리지 않도록 클래스 아이콘이 보다 적은 공간을 차지하도록 업데이트해야 합니다 . 32 픽셀 (가로) × 19 픽셀 (세로) 보다 작은 이미지를 사용하십시오 .

LabVIEW 에서 LLB 열기

옵션 대화 상자의 **환경** 페이지에서의 **LLB 파일의 Windows 탐색기 활성화** 옵션이 없어졌습니다 . LabVIEW 는 **LLB 관리자** 윈도우에서 LLB 를 엽니다 . National Instruments 웹 사이트 (ni.com/info) 에서 정보 코드 `exvfc5` 를 입력하면 LLB 열기에 대한 보다 자세한 정보를 참조할 수 있습니다 .

Timed 루프 우선순위 레벨 제한

LabVIEW 8.2.x 및 이전 버전에서는 Timed 루프의 우선순위 레벨로 32 의 2 거듭제곱까지 선택할 수 있습니다 . LabVIEW 8.5 및 이후 버전은 65,535 미만의 우선 순위 레벨만을 지원합니다 .

웨이브폼 데이터 타입

웨이브폼 배열의 경계 밖에서 인덱스할 때, 결과로 나타나는 웨이브폼은 값이 0(부적합한 웨이브폼) 이 아닌 1 인 적합한 기본 웨이브폼입니다 . 스칼라 출력 터널에서 제로 횟수로 For 루프를 실행할 때에도 참입니다 .

LabVIEW 8.0 에서 업그레이드하기

LabVIEW 8.0 에서 LabVIEW 8.6 으로 업그레이드할 때 다음과 같은 호환성 문제가 발생할 수 있습니다 . 업그레이드할 때 발생할 수 있는 문제에 대한 정보는 이 문서의 [LabVIEW 8.2 에서 업그레이드하기](#)와 [LabVIEW 8.5 에서 업그레이드하기](#) 섹션을 참조하십시오 .

각 버전의 새로운 기능 및 변경 내역에 대한 추가적인 정보는 ni.com/manuals 에서 버전 8.0 과 8.5 까지 각 LabVIEW 버전에 대한 [LabVIEW 업그레이드 노트](#)를 참조하십시오 .

지원하는 플랫폼

LabVIEW 8.2 및 이후 버전의 지원하는 플랫폼에서 다음과 같이 변경되었습니다 :

- LabVIEW 8.2 및 이후 버전은 Windows XP x64 를 지원하지 않습니다 .
- LabVIEW 8.2 및 이후 버전은 Mac OS X 10.3.8 또는 이전 버전을 지원하지 않습니다 .
- LabVIEW 8.2 는 Intel 프로세서가 내장된 Macintosh 컴퓨터를 일부 지원합니다 .
National Instruments 웹 사이트 (ni.com/info) 에서 정보 코드 macintel 을 입력하면 Macintosh 지원에 대한 보다 자세한 정보를 얻을 수 있습니다 . LabVIEW 8.5 는 Intel 및 PowerPC 프로세서 모두가 내장된 Macintosh 컴퓨터를 지원합니다 .

시스템 사양

(Windows) LabVIEW 8.2 는 LabVIEW 설치시 최소 1.2 GB 가 필요합니다 . LabVIEW 8.6 은 LabVIEW 설치시 최소 1.6 GB 가 필요합니다 .

(Mac OS) LabVIEW 8.2 는 LabVIEW 최소 설치시 최소 500 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 700 MB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 262 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 828 MB 의 빈 디스크 공간이 필요합니다 .

(Linux) LabVIEW 8.2 는 LabVIEW 최소 설치시 최소 430 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시에는 620 MB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 365 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 651 MB 의 빈 디스크 공간이 필요합니다 .

인쇄된 문서

LabVIEW 8.2 에서 다음의 문서는 변경되지 않았습니다 . 따라서 LabVIEW 8.2 에서 변경한 사항이 문서 내용에 반영되지 않을 수 있습니다 .

- *LabVIEW 도움 카드*
- *LabVIEW 기본 매뉴얼 - LabVIEW 기본 매뉴얼은 LabVIEW 도움말에서 기본* 모음의 일부 분입니다 . 따라서 , 업데이트된 내용에 대한 정보는 *LabVIEW 도움말의 목차* 탭에서 **기본** 모음을 참조하십시오 .

VI 와 함수 동작 변경 내역

LabVIEW 8.2 및 이후 버전에서 VI 와 함수의 동작이 다음과 같이 변경되었습니다 .

어플리케이션 인스턴스 사이에서 통신하기

LabVIEW 8.2 및 이후 버전에서는 (큐 얻기) , (알림자 얻기) , (사용자 이벤트 생성) , (세마포어 생성) , (랑대부 생성) 함수를 사용하여 LabVIEW 어플리케이션 인스턴스 사이에서 통신할 수 없습니다 . 하나의 어플리케이션 인스턴스에서 큐 , 알림자 , 사용자 이벤트 , 세마포어 , 또는 랑대부 참조를 구하거나 생성하면 , 다른 어플리케이션 인스턴스에서 해당 참조를 사용할 수 없습니다 .

원래의 고유벡터로 변환 VI

(원래의 고유벡터로 변환) VI 의 **낮은 인덱스** , **높은 인덱스** 및 **스케일** 입력은 필수 입력입니다 .

DataSocket 쓰기 함수

LabVIEW 8.0.1 에서 (DataSocket 쓰기) 함수의 기본 동작은 비동기형으로 변경되었습니다. LabVIEW 8.0 및 LabVIEW 8.2 또는 이후 버전이 컴퓨터에 설치되어 있는 경우, labview\examples\Shared Variable 디렉토리의 DataSocket API Client VI 예제는 사용자가 해당 VI 를 정지시킬 때 에러를 반환합니다. LabVIEW 8.2 또는 이후 버전에서 이 예제를 사용하려면 LabVIEW 8.0 을 LabVIEW 8.0.1 로 업데이트해야 합니다 .

파일 I/O VI

(스프레드시트 파일에 쓰기) VI 와 (스프레드시트 파일로부터 읽기) VI 는 다형성 VI 입니다. (스프레드시트 파일에 쓰기) VI 는 사용자가 **포맷** 입력에 연결한 값에 맞춥니다. (스프레드시트 파일로부터 읽기) VI 는 다음의 인스턴스를 포함합니다 : DBL, I64, 문자열 .

GPIO 상태 함수

LabVIEW 8.0 에서, (GPIO 상태) 함수는 **에러 입력**에서 에러를 수신하는 경우 실행되지 않았 습니다. LabVIEW 8.2 또는 이후 버전에서, (GPIO 상태) 함수는 **에러 입력** 입력에서 에러를 수 신하더라도 항상 실행됩니다 .

히스토그램 VI

(히스토그램) VI 의 기본 **간격** 입력이 10 으로 변경되었습니다 .

VI 참조 열기 함수

(VI 참조 열기) 함수의 **옵션** 입력의 기본 동작은 참조된 VI 에서 찾을 수 없는 SubVI 를 검색할 때 로딩 대화 상자를 디스플레이하는 것입니다. LabVIEW 가 자동으로 찾을 수 없는 VI 를 찾도 록 입력 요청이 나타나지 않습니다. 새 값인 0x20 은 참조된 VI 의 찾을 수 없는 SubVI 를 검색 하도록 로딩 대화 상자를 디스플레이하지 않도록 지정합니다. 사용자에게 찾을 수 없는 SubVI 를 찾도록 입력 요청하려면, 옵션 0x10 값을 사용합니다 .

다항식 근 VI

P(x) 가 제로가 아닌 상수일 때, (다항식 근) VI 는 에러를 반환하지 않습니다. 그러나, **P(x)** 가 0 이면, (다항식 근) VI 는 에러 -20111 을 반환합니다. 이 VI 의 입력 다항식 계수의 일부는 제로가 아니어야 합니다 .

램프 패턴 VI

(램프 패턴) VI 에서 **샘플이 1 이고 끝 제외?** 가 참인 경우, VI 는 **시작**의 한 원소를 가진 배열 을 에러 없이 반환합니다. LabVIEW 8.0 에서 VI 는 이러한 조건에서 에러를 반환합니다 .

레지스트리 값 읽기 VI

LabVIEW 8.0 은 VI 가 패턴화된 문자열 배열에 사용하는 REG_MULTI_SZ 문자열 포맷팅을 을 바르지 않게 처리했습니다. 이 문제 때문에 사용자는 (레지스트리 값 읽기) VI 에서 이 데이터 타입을 처리하기 위해 분석기를 작성해야 했습니다. LabVIEW 8.2 및 이후 버전에서, (레지스 트리 값 읽기) VI 가 (레지스트리 값 쓰기) VI 에서 사용되는 포맷과 같은 포맷으로 이 데이터 타입을 반환합니다. 이제 사용자가 직접 분석기를 추가하지 않아도 됩니다. LabVIEW 8.2 및 이후 버전에서, 이러한 VI 와 사용자 분석기를 함께 사용하면 (레지스트리 값 읽기) VI 가 잘못된 데이터를 반환하게 됩니다 .

웨이브폼 리샘플 (한번) VI

(웨이브폼 리샘플 (한번)) VI 의 **열린 구간?** 입력의 기본값이 참에서 거짓으로 변경되었으며, 기본으로 닫힌 구간이 선택됩니다. 따라서 기존의 코드를 업데이트하지 않으면, VI 는 예상되 는 결과를 반환하지 않을 수도 있습니다 .

사운드 VI

(사운드 입력 읽기) 와 (단순한 사운드 파일 읽기) VI 에서, 데이터 출력의 t0 구성요소는 읽 은 첫번째 샘플의 타임스탬프를 반환합니다. LabVIEW 는 첫번째 샘플을 읽는 최초 시간의 근 사치를 구합니다 .

이제 연속 사운드 태스크에서 사운드를 정지하기 위해 (사운드 출력 정지) VI 를 호출하지 않아도 됩니다 .

(사운드 출력 기다림) VI 는 **연속 샘플** 모드와 **유한 샘플** 모드에서 작동합니다 .

웨이브폼 VI

LabVIEW 8.2 및 이후 버전에서 웨이브폼 VI 가 다음과 같이 변경되었습니다 :

- **기본 레벨 트리거 검출 VI**—이 VI 의 양쪽 인스턴스에서 , **기울기 입력이 트리거 기울기** 로 변경되었습니다 .
- **웨이브폼 부분 얻기 VI**—다음의 인스턴스를 포함합니다 : WDT 웨이브폼 부분 얻기 DBL, WDT 웨이브폼 부분 얻기 CDB, WDT 웨이브폼 부분 얻기 EXT, WDT 웨이브폼 부분 얻기 I16, WDT 웨이브폼 부분 얻기 I32, WDT 웨이브폼 부분 얻기 I8, WDT 웨이브폼 부분 얻기 SGL. 시작 / 지속기간 포맷 입력은 더 이상 **절대 시간** 옵션을 포함하지 않습니다 . 시작 입력이 시작 샘플 / 시간으로 변경되었으며 , 실제 시작 출력은 실제 시작 샘플 / 시간으로 변경되었습니다 .
- **웨이브폼 시간 배열 얻기 VI**—X 배열 출력이 배열도 부동소수 데이터 타입에서 타임스탬프 데이터 타입으로 변경되었습니다 .
- **Y 값 얻기 VI**—이 VI 와 그에 대응하는 다형성 인스턴스는 XY 값 얻기로 변경되었습니다 . (XY 값 얻기) VI 에는 이제 X 값 출력을 포함하며 , **데이터 값** 출력은 Y 값으로 변경되었습니다 .
- **웨이브폼 샘플 개수 VI**—이 VI 는 다음의 인스턴스가 있는 다형성 VI 입니다 : WDT 웨이브폼 샘플 개수 DBL, WDT 웨이브폼 샘플 개수 CDB, WDT 웨이브폼 샘플 개수 EXT, WDT 웨이브폼 샘플 개수 I16, WDT 웨이브폼 샘플 개수 I32, WDT 웨이브폼 샘플 개수 I8, WDT 웨이브폼 샘플 개수 SGL.
- **파일로부터 웨이브폼 읽기 VI**—에러가 파일 끝 에러인 경우 , **에러 출력** 출력의 에러 상태를 참으로 반환합니다 .
- **부분 대체 VI**—시작 입력이 시작 샘플 / 시간으로 변경되었으며 , 실제 시작 출력은 실제 시작 샘플 / 시간으로 변경되었습니다 .
- **디지털 패턴 찾기 VI**—시작 입력은 시작 인덱스 / 시간으로 변경되었습니다 .
- **웨이브폼 찾기 VI**—최적 피팅 시간 및 피팅 시간 출력은 배열도 부동소수 데이터 타입에서 타임스탬프 데이터 타입으로 변경되었습니다 .
- **웨이브폼 최소 최대 VI**—최소 시간 및 최대 시간 출력은 배열도 부동소수 데이터 타입에서 타임스탬프 데이터 타입으로 변경되었습니다 .
- **웨이브폼을 XY 쌍으로 VI**—XY 쌍 출력의 x 원소는 배열도 부동소수 데이터 타입에서 타임스탬프 데이터 타입으로 변경되었습니다 .

프로퍼티 , 메소드 , 이벤트 동작 변경 내역

LabVIEW 8.2 및 이후 버전에서 프로퍼티 , 메소드 , 이벤트의 동작이 다음과 같이 변경되었습니다 :

- ActiveX VI 참조 얻기 메소드의 **옵션** 입력의 기본 동작은 사용자에게 참조된 VI 의 찾을 수 없는 subVI 를 검색하도록 입력 요청을 하는 것입니다 . 새 값인 0x20 은 **찾기** 대화 상자를 디스플레이하지 않거나 또는 참조된 VI 의 찾을 수 없는 SubVI 를 검색하도록 입력 요청하지 않도록 지정합니다 .
- 프로젝트에서 열려 있지 않은 라이브러리에 공유 변수를 추가하려는 경우 , 프로젝트 아이 템 클래스의 아이템 추가 메소드가 에러를 반환합니다 .
- VI 실행 메소드의 **참조 자동 삭제** 입력이 참이고 메소드가 에러를 반환하는 경우 , LabVIEW 는 참조를 삭제하지 않습니다 .

- 어플리케이션 : 언어 프로퍼티의 유효한 값에 zh-cn 이 포함되었습니다 . 이는 중국어 간체가 LabVIEW 환경 언어임을 나타냅니다 .
- LabVIEW 8.0 에서 , 참조에 의해 배열 데이터 타입을 전달하는 .NET 메소드는 모든 데이터를 참조 번호 데이터 타입으로 전달합니다 . LabVIEW 8.2 및 이후 버전에서 , 참조에 의해 배열 데이터 타입을 전달하는 .NET 메소드는 데이터를 실제 데이터 타입으로 전달합니다 .
- 디지털 테이블 , 여러 열 리스트박스 , 테이블 , 트리 컨트롤 클래스의 편집 위치 프로퍼티는 (-2, -2) 의 값을 반환하여 사용자가 컨트롤 텍스트를 편집하는 중이 아님을 나타냅니다 . 리스트박스 클래스의 행 편집 프로퍼티는 -2 의 값을 반환하여 사용자가 컨트롤 텍스트를 편집하는 중이 아님을 나타냅니다 .
- LabVIEW 8.0 에서 , 패널 업데이트 연기 프로퍼티는 서브패널에 있는 프론트패널의 업데이트를 연기하지 않았습니다 . LabVIEW 8.2 및 이후 버전에서 , 패널 업데이트 연기 프로퍼티가 서브패널에도 작동합니다 .
- 어플리케이션 인스턴스 닫기 및 어플리케이션 인스턴스 닫기 ? 이벤트는 어플리케이션 종료 및 어플리케이션 종료 ? 이벤트를 대체합니다 . LabVIEW 프로젝트 밖에서 실행 중인 VI 에서 어플리케이션 인스턴스 닫기 이벤트를 사용하는 경우 , 사용자 인터페이스를 통해서 또는 프로그램적으로 LabVIEW 를 종료할 때 LabVIEW 는 이벤트를 생성합니다 . LabVIEW 는 사용자 인터페이스를 통해 LabVIEW 를 종료할 때 어플리케이션 인스턴스 닫기 ? 이벤트를 생성합니다 . 사용자가 LabVIEW 프로젝트 내에서 실행 중인 VI 를 위해 어플리케이션 인스턴스 닫기 및 어플리케이션 인스턴스 닫기 ? 이벤트를 등록하면 , LabVIEW 는 어플리케이션 인스턴스를 닫거나 LabVIEW 를 종료할 때 이벤트를 생성합니다 .

삭제된 프로퍼티 , 메소드 , 이벤트

LabVIEW 8.2 및 이후 버전에서는 다음의 프로퍼티 , 메소드 , 이벤트를 지원하지 않습니다 .

- LabVIEW 8.2 및 이후 버전은 커백터 팬 프로퍼티를 지원하지 않습니다 .
- LabVIEW 8.x는 변수 클래스에서 데이터 타입 프로퍼티를 지원하지 않습니다 . 대신에 변수 클래스의 데이터 타입 (배리언트) 프로퍼티를 사용합니다 .

이름이 변경된 프로퍼티 , 메소드 , 이벤트

LabVIEW 8.2 및 이후 버전에서 다음의 프로퍼티 , 메소드 , 이벤트의 이름이 변경되었습니다 :

클래스	LabVIEW 8.0 이름	LabVIEW 8.2 및 이후 버전 이름	타입
어플리케이션	슬레이브 연결 끊기	LVRT: 슬레이브 연결 끊기	메소드
어플리케이션	어플리케이션 종료	어플리케이션 인스턴스 닫기	이벤트
어플리케이션	어플리케이션 종료 ?	어플리케이션 인스턴스 닫기 ?	이벤트
강도 그래프 , 혼합 신호 그래프 , 웨이브폼 그래프	커서 팔레트 보이기	커서 범례 보이기	프로퍼티
라이브러리	라이브러리 태그 삭제	라이브러리 태그 : 삭제	메소드
라이브러리	아이콘 열기	아이콘 : 열기	메소드
라이브러리	라이브러리 태그 열기	라이브러리 태그 : 열기	메소드
라이브러리	라이브러리 태그 이름 열기	라이브러리 태그 : 이름 열기	메소드
라이브러리	잠금 상태 열기	잠금 상태 : 열기	메소드
라이브러리	소스 영역 열기	소스 영역 : 열기	메소드

클래스	LabVIEW 8.0 이름	LabVIEW 8.2 및 이후 버전 이름	타입
라이브러리	저장	저장 : 라이브러리	메소드
라이브러리	복사본 저장	저장 : 복사본	메소드
라이브러리	아이콘 설정	아이콘 : 설정	메소드
라이브러리	라이브러리 태그 설정	라이브러리 태그 : 설정	메소드
라이브러리	잠금 상태 설정	잠금 상태 : 설정	메소드
라이브러리	소스 영역 설정	소스 영역 : 설정	메소드
리스트박스, 여러 열 리스트박스, 트리 컨트롤	끌기 / 놓기 : 아이템 끌기 허용	끌기 / 놓기 : 끌기 허용	프로퍼티
경로 및 문자열	놓기 허용	놓기 허용	프로퍼티
프로젝트 아이템	태그 삭제	태그 : 삭제	프로퍼티
프로젝트 아이템	태그 얻기	태그 : 태그 얻기	프로퍼티
프로젝트 아이템	태그 이름 얻기	태그 : 이름 얻기	프로퍼티
프로젝트 아이템	XML 태그 얻기	태그 : XML 태그 얻기	프로퍼티
프로젝트 아이템	태그 설정	태그 : 태그 설정	프로퍼티
프로젝트 아이템	XML 태그 설정	태그 : XML 태그 설정	프로퍼티
프로젝트 아이템	라이브러리 아이템 타입 문자열	라이브러리 아이템 타입 : 문자열	프로퍼티
프로젝트 아이템	라이브러리 아이템 타입	라이브러리 아이템 : 타입	프로퍼티

어플리케이션 빌드 변경사항

LabVIEW 8.2 및 이후 버전에서, 어플리케이션이나 공유 라이브러리의 이름을 .11b 파일 확장자를 갖도록 변경하여 독립 어플리케이션 (EXE) 또는 공유 라이브러리 (DLL)의 내용을 볼 수 없습니다. 또한 어플리케이션이나 공유 라이브러리 외부에서 VI 경로를 지정하여 독립 어플리케이션이나 공유 라이브러리 안에 있는 VI에 접근할 수도 없습니다. National Instruments 웹 사이트 ni.com/info에서 정보 코드 exjk3b를 입력하여 어플리케이션과 공유 라이브러리 보기와 접근에 대한 보다 자세한 정보를 확인할 수 있습니다.

LabVIEW 7.x에서 업그레이드하기

LabVIEW 7.x에서 LabVIEW 8.6으로 업그레이드할 때 다음과 같은 호환성 문제가 발생할 수 있습니다. 업그레이드할 때 발생할 수 있는 문제에 대한 정보를 이 문서의 [LabVIEW 8.0에서 업그레이드하기](#), [LabVIEW 8.2에서 업그레이드하기](#), [LabVIEW 8.5에서 업그레이드하기](#) 섹션을 참조하십시오.

각 버전의 새로운 기능 및 변경 내역에 대한 추가적인 정보는 ni.com/manuals에서 LabVIEW 버전 7.x부터 8.5까지 각각의 버전에 대한 *LabVIEW 업그레이드 노트*를 참조하십시오.



노트 LabVIEW 8.6에서 *LabVIEW 기본 매뉴얼*은 변경되지 않았습니다. 해당 문서의 PDF 버전은 labview\manuals 디렉토리에서 접근할 수 있습니다. *LabVIEW 기본 매뉴얼*이 *LabVIEW 도움말*에서 **기본** 모음의 일부입니다. 따라서, 업데이트된 내용에 대한 정보는 *LabVIEW 도움말*의 **목차** 탭에 있는 **기본** 모음에서 참조하십시오.

지원하는 플랫폼

LabVIEW 8.x에서 지원하는 플랫폼이 다음과 같이 변경되었습니다 :

- LabVIEW 7.1 및 이후 버전은 Windows Me/98/95 를 지원하지 않습니다 . LabVIEW 8.x는 Windows NT 를 지원하지 않습니다 .
- LabVIEW 8.x는 Mac OS X 10.2 또는 이전 버전을 지원하지 않습니다 .
- LabVIEW 8.x는 Sun Solaris 를 지원하지 않습니다 .

시스템 사양

LabVIEW 7.x의 최소 RAM 사양은 128 MB 이지만 , National Instruments 는 256 MB 의 RAM 을 권장합니다 . LabVIEW 8.5 는 최소 256 MB 의 RAM 이 필요하지만 , National Instruments 는 1 GB 의 RAM 을 권장합니다 .

LabVIEW 7.x는 800 × 600 픽셀의 화면 해상도가 필요하지만 , National Instruments 는 1,024 × 768 픽셀의 화면 해상도를 권장합니다 . LabVIEW 8.x는 1,024 × 768 픽셀의 화면 해상도가 필요합니다 .

Windows

LabVIEW 7.x는 최소 Pentium III 또는 그 이상 , Celeron 600 MHz 또는 동등한 프로세서가 필요하지만 , National Instruments 는 Pentium 4 또는 이와 동등한 프로세서를 권장합니다 . LabVIEW 8.x는 최소 Pentium III, Celeron 866 MHz, 또는 동등한 프로세서가 필요하지만 , National Instruments 는 Pentium 4/M 또는 동등한 프로세서를 권장합니다 .

LabVIEW 7.x는 LabVIEW 최소 설치시 최소 130 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 550 MB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 전체 설치시 1.6 GB 의 디스크 공간이 필요합니다 .

Mac OS

LabVIEW 7.x는 LabVIEW 최소 설치시 최소 280 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 350 MB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 262 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 828 MB 의 빈 디스크 공간이 필요합니다 .

Linux

LabVIEW 7.x는 최소 Pentium III 또는 그 이상 , Celeron 600 MHz 또는 동등한 프로세서가 필요하지만 , National Instruments 는 Pentium 4 또는 이와 동등한 프로세서를 권장합니다 . LabVIEW 8.x는 최소 Pentium III, Celeron 866 MHz, 또는 동등한 프로세서가 필요하지만 , National Instruments 는 Pentium 4/M 또는 동등한 프로세서를 권장합니다 .

LabVIEW 7.x는 LabVIEW 최소 설치시 최소 200 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 300 MB 의 빈 디스크 공간이 필요합니다 . LabVIEW 8.6 은 LabVIEW 최소 설치시 최소 365 MB 의 빈 디스크 공간이 필요하며 , LabVIEW 전체 설치시 651 MB 의 빈 디스크 공간이 필요합니다 .

LabVIEW 7.x는 GNU C 라이브러리 (glibc) 2.1.3 또는 이후 버전이 필요하지만 , National Instruments 는 GNU C 라이브러리 2.2.4 또는 이후 버전을 권장합니다 . LabVIEW 8.x는 GNU C 라이브러리 2.2.4 또는 이후 버전이 필요합니다 .

LabVIEW 7.x는 Red Hat Linux 7.0 또는 이후 버전 , Mandrake Linux 8.0 또는 이후 버전 , SuSE Linux 7.1 또는 이후 버전 , Debian Linux 3.0 또는 이후 버전에서 동작합니다 . LabVIEW 8.x는 Red Hat Enterprise Linux WS 3 또는 이후 버전 , MandrakeLinux/Mandriva 10.0 또는 이후 버전 , SuSE Linux 9.1 또는 이후 버전에서 동작합니다 .

사용자 팔레트 보기

LabVIEW 8.x는 사용자 팔레트 보기를 지원하지 않습니다. 사용자 팔레트 보기를 사용하지 않고도 팔레트 세트를 편집할 수 있습니다. LabVIEW 8.0에서 팔레트 변경 내역에 대한 추가적인 정보는 National Instruments 웹 사이트 ni.com/info에서 정보 코드 1v8palette를 입력합니다.

VI와 함수 동작 변경 내역

LabVIEW 7.1 또는 8.0에서 VI와 함수의 동작이 다음과 같이 변경되었습니다.

.NET VI와 어플리케이션

LabVIEW 8.x에서 .NET 함수와 어플리케이션을 사용하려면 .NET Framework 1.1 Service Pack 1 또는 이후 버전이 있어야 합니다. .NET Framework 1.1 Service Pack 1을 설치하기 전에 반드시 Microsoft .NET Framework 1.1 Hotfix KB886904를 제거해야 합니다.

LabVIEW 7.x에서 마지막으로 저장한 .NET VI를 로드하는 경우, LabVIEW 8.x는 어셈블리 파일이 VI와 동일한 디렉토리에 있거나 LabVIEW 7.x에서 **Tools >> Advanced >> .NET Assembly References**를 선택하여 어셈블리를 등록한 경우에도, VI가 참조하는 어셈블리를 찾도록 입력 요청할 수 있습니다.

분석 VI 알고리즘

LabVIEW 7.1 및 이후 버전에서, 분석 VI는 BLAS/LAPACK 알고리즘을 사용합니다. 이 VI는 더욱 정확한 결과를 생성합니다. LabVIEW 8.x에서, 이 VI는 **수학**과 **신호 처리** 팔레트에 있습니다.

신호 추가 익스프레스 VI

LabVIEW 7.x에서, Append Signals Express VI의 **Input Signal A**가 비어있거나 연결되지 않은 상태에서 단일 신호 또는 결합된 신호를 **Input Signal B**에 연결하는 경우, **Appended Signals** 출력은 비어 있습니다. LabVIEW 8.x에서, **입력 신호 A**가 비어있거나 연결되지 않은 상태에서 단일 신호를 **입력 신호 B**에 연결한 경우, 익스프레스 VI는 **입력 신호 B**를 반환합니다. 결합된 신호만을 **입력 신호 B**에 연결하는 경우, 결합된 신호의 각 신호는 다음 신호를 추가하여 결과로 하나의 신호를 생성합니다.

비교 함수

LabVIEW 7.x 또는 이전 버전에서, 배리언트 데이터를 비교하기 위해 Comparison 함수를 사용할 때, LabVIEW는 먼저 두 배리언트의 전체 길이를 비교한 후, 배리언트를 비트 단위로 비교합니다. LabVIEW 8.x는 배리언트의 실제 타입 정보를 인코딩하는 타입 코드로 배리언트 데이터를 비교한 후, 다른 타입별 속성을 비교합니다.

내적 VI

LabVIEW 7.0에서 Dot Product VI는 다음 방정식을 사용하여 입력 벡터 X 와 Y 의 내적을 계산합니다:

$$X * Y = \sum_{i=0}^{n-1} X_i Y_i$$

LabVIEW 7.1 및 이후 버전에서, Dot Product VI는 다음 방정식을 사용하여 복소수 입력의 내적을 계산합니다:

$$X^* Y = \sum_{i=0}^{n-1} X_i Y_i^*$$

여기서, Y_i^* 는 Y_i 에 대한 켈레 복소수입니다.

Easy Text Report VI (Mac OS 와 Linux)

Easy Text Report VI 의 커브터 팬이 변경되었습니다. LabVIEW 8.x 에서, LabVIEW 7.x 또는 이전 버전에서 마지막으로 저장되었던 Easy Text Report VI 를 사용한 VI 를 열 때, SubVI 에서 마우스 오른쪽 버튼을 클릭한 후 바로 가기 메뉴에서 **Relink To SubVI** 를 선택해야 합니다.

문자열로 포맷 함수

LabVIEW 7.x 에서, %o, %b 또는 %x 포맷 지정자 구문 원소를 Format Into String 함수와 함께 사용하면 입력을 문자열로 변환하기 전에 부동소수 입력이 32 비트 정수로 반올림됩니다.

LabVIEW 8.x 에서, 이 포맷 지정자 구문 원소는 (문자열로 포맷) 함수가 입력을 문자열로 변환하기 전에 부동소수 입력을 64 비트 정수로 반올림하도록 합니다.

숫자 결합 함수

LabVIEW 7.x 및 이전 버전에서, Join Numbers 함수는 32 비트 정수 입력을 16 비트 정수로 강제 변환하여 하나의 32 비트 정수를 생성했습니다. LabVIEW 8.x 에서, (숫자 결합) 함수는 32 비트 정수 입력을 결합하여 하나의 64 비트 정수를 생성합니다.



노트 LabVIEW 7.x VI 를 LabVIEW 8.x 에서 여는 경우, LabVIEW 는 32 비트 정수 입력을 16 비트 정수로 강제 변환합니다.

수학 VI 와 행렬

LabVIEW 8.x 에서, 수학 VI 는 행렬 데이터 타입을 지원합니다. LabVIEW 8.x 에서 LabVIEW 7.x 로부터 VI 를 로드하며, 로드한 VI 에 행렬 데이터 타입을 사용할 수 있지만 2D 배열을 사용하는 함수에 연결된 수학 VI 가 있는 경우, 빨간색 7.x 문양이 함수에 나타납니다. 이 빨간색 문양은 LabVIEW 가 2D 배열을 행렬 데이터 타입으로 대체했음을 나타냅니다.

숫자를 문자열로 변환 함수

LabVIEW 7.x 에서, Number to Hexadecimal String, Number to Octal String, Number to Decimal String 함수는 입력을 문자열로 변환하기 전에 부동소수 입력을 32 비트 정수로 반올림합니다.

LabVIEW 8.x 에서, 이 함수는 입력을 문자열로 변환하기 전에 부동소수 입력을 64 비트 정수로 반올림합니다. 그러나 LabVIEW 7.x VI 를 LabVIEW 8.x 에서 여는 경우, LabVIEW 는 부동소수 입력을 32 비트 정수로 반올림하여 호환성과 기능을 유지합니다.

VI 참조 열기 함수

LabVIEW 7.x 에서, Open VI Reference 함수의 vi path 입력이 경로이고 메모리에 같은 이름의 VI 가 존재하는 경우, 메모리 상의 VI 경로가 지정한 경로와 일치하지 않더라도 LabVIEW 는 메모리 상의 VI 참조를 반환합니다.

LabVIEW 8.x 에서 (VI 참조 열기) 의 vi 경로 입력이 문자열이면, LabVIEW 는 vi 경로가 타겟 메모리 상에 있는 VI 의 유효한 VI 이름과 일치할 때만 VI 를 엽니다. vi 경로가 경로인 경우, LabVIEW 는 동일한 타겟에서 경로가 같은 VI 를 메모리에서 검색합니다. LabVIEW 가 경로가 일치하는 VI 를 찾지 못하는 경우, LabVIEW 는 디스크의 지정된 경로에서 VI 를 로드합니다. LabVIEW 8.5 및 이후 버전에서, LabVIEW 가 파일을 찾을 수 없거나 파일의 VI 이름이 해당 타겟의 메모리에 있는 다른 VI 의 유효한 VI 이름과 같으면 예러가 발생합니다.

빠른 스케일 VI

LabVIEW 7.1 및 이전 버전에서, Quick Scale 1D VI 또는 Quick Scale 2D VI 의 X 입력이 제로의 배열인 경우, 이 VI 는 $\max|X|$ 를 0 으로, $Y(i)=X(i)/\max|X|$ 또는 $Yij=Xij/\max|X|$ 를 NaN 의 배열로 반환합니다. LabVIEW 8.x 에서, (빠른 스케일) VI 의 X 입력이 제로의 배열인 경우, 이 VI 는 $\max|X|$ 를 0 으로, $Y(i)=X(i)/\max|X|$ 또는 $Yij=Xij/\max|X|$ 를 제로의 배열로 반환합니다.

키 읽기 VI

LabVIEW 7.x 및 이전 버전에서, Read Key VI의 문자열 인스턴스를 사용하여 Shift-JIS에서 인코딩된 멀티바이트 문자의 문자열을 읽습니다. **multibyte encoding** 입력에 1 또는 <Shift-JIS>를 연결해야 합니다. LabVIEW 8.x에서는, OS 로컬을 적절한 인코딩으로 설정해 놓으면 (키 읽기) VI의 문자열 인스턴스는 기본적으로 멀티바이트 문자의 인코딩된 문자열을 읽습니다.

스케일 VI

LabVIEW 7.1 및 이전 버전에서, Scale 1D VI 또는 Scale 2D VI의 **X** 입력이 제로의 배열인 경우, 이 VI는 **scale**을 0으로, **offset**을 0으로, $Y=(X-\text{offset})/\text{scale}$ 을 NaN의 배열로 반환합니다. LabVIEW 8.x에서, (스케일) VI의 **X** 입력이 제로의 배열인 경우, 이 VI는 **스케일**을 1로, **오프셋**을 0으로, $Y=(X-\text{오프셋})/\text{스케일}$ 을 제로의 배열로 반환합니다.

세마포어 VI

LabVIEW 7.x에서, Release Semaphore VI와 Acquire Semaphore VI는 error in 입력이 에러를 받을 때 실행을 시도하지 않습니다. LabVIEW 8.x의 경우, 이 VI는 에러 입력 입력이 에러를 받더라도 실행을 시도합니다. 그러나, LabVIEW 7.x VI를 LabVIEW 8.x에서 열 경우, LabVIEW 7.x의 특징을 유지하기 위해 LabVIEW는 VI를 업데이트합니다.

SMTP E-메일 VI

LabVIEW 7.x 및 이전 버전에서는, 값을 SMTP Email VI의 문자 세트 입력에 연결하여 문자 세트를 지정할 수 있습니다. LabVIEW 8.x에서는, (SMTP E-메일) VI는 메시지가 시스템 문자 세트에 있다고 가정합니다. 이 VI는 E-메일을 보내기 전에 메시지를 UTF-8 포맷으로 인코딩합니다. (SMTP E-메일) VI에는 더 이상 문자 세트 또는 문자 변환 파라미터가 없습니다.

복소수 정렬 VI

LabVIEW 7.x 및 이전 버전에서 Sort Complex Numbers VI의 method 입력을 **Magnitude**로 설정할 경우 LabVIEW는 동일한 크기를 가진 원소들의 순서를 바꾸지 않습니다. LabVIEW 8.x에서 메소드를 **크기**로 설정하면, LabVIEW는 동일한 크기를 가진 원소를 먼저 실수 부분에 대하여 정렬한 후, 허수 부분에 대하여 정렬합니다.

단위 벡터 VI

LabVIEW 7.x 및 이전 버전에서, Unit Vector VI는 다음 방정식을 사용하여 입력 벡터의 노름을 계산합니다:

$$\|X\| = \sqrt{x_0^2 + x_1^2 + \dots + x_{n-1}^2}$$

LabVIEW 8.x에서, (단위 벡터) VI는 다음 방정식을 사용하여 입력 벡터의 노름을 계산합니다:

$$\|X\| = \left(|x_0|^y + |x_1|^y + \dots + |x_{n-1}|^y \right)^{\frac{1}{y}}$$

이 때 X 는 입력 벡터이고 $\|X\|$ 는 노름, y 는 노름 타입입니다.

사용자 VI

labview\help, labview\project 또는 labview\wizard 디렉토리에 놓여지는 VI는 각각 **도움말**, **도구**, **파일** 메뉴에 나타납니다. LabVIEW 7.x 또는 이전 버전에서 이 디렉토리에 놓여지는 VI는 LabVIEW 8.x에서 예상대로 작동하지 않을 수 있습니다. 왜냐하면 LabVIEW 8.0 및 이후 버전은 이 VI를 프라이빗 어플리케이션 인스턴스에서 열기 때문입니다.

labview\vi.lib\Utility\allVIsInMemory.11b 의 VIMemory Get VIs in Memory VI 를 사용하여 모든 어플리케이션 인스턴스에서 메모리의 모든 사용자 VI 리스트를 생성합니다. labview\vi.lib\Utility\allVIsInMemory.11b의 Get User Application Reference VI 를 사용하여 현재 어플리케이션 인스턴스에 대한 참조를 생성합니다. 어플리케이션 인스턴스에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오.

삭제된 VI 와 함수

LabVIEW 8.x에서는 다음의 VI 와 함수를 지원하지 않습니다:

- LabVIEW 7.1 및 이후 버전은 Polynomial Real Zero Counter VI 를 설치하지 않습니다. 대신 (실수 다항식의 제로 카운터) VI 를 사용하십시오.
- **(Mac OS)** LabVIEW 7.1 및 이후 버전은 PPC VI 를 설치하지 않습니다. 대신 TCP VI 를 사용하십시오.
- LabVIEW 8.x는 (QR 인수분해) VI 를 지원하지 않습니다. 대신 (QR 분해) VI 를 사용하십시오.
- LabVIEW 8.x는 더이상 (레벤버그 - 마르카토) VI 또는 (비선형 레벤버그 - 마르카토 피팅) VI 를 지원하지 않습니다. 대신에 (비선형 커브 피팅) VI 를 사용하십시오.
- LabVIEW 8.x에서는 **함수** 팔레트에 VISA Status Description 함수가 없습니다. 대신 (단 순 에러 핸들러) 또는 (일반 에러 핸들러) VI 를 사용하십시오.
- LabVIEW 8.x는 (카이 제공 분포), (F 분포), (정규분포) 및 (T 분포) VI 를 지원하지 않습니다. 대신에 (연속적인 누적 분포 함수) VI 에서 카이제공, F, 정규, 스튜던트 t 인스턴스를 사용합니다.
- LabVIEW 8.x는 (역 카이제공 분포), (역 F 분포), (역 정규분포), (역 T 분포) VI 를 지원하지 않습니다. 대신에 (연속적인 역 누적 분포 함수) VI 에서 카이제공, F, 정규, 스튜던트 t 인스턴스를 사용합니다.
- LabVIEW 8.x에서 (1D 선형식 계산) VI 와 (2D 선형식 계산) VI 는 더 이상 **함수** 팔레트에 없습니다. 대신에 (선형 계산) VI 를 사용하십시오.
- LabVIEW 8.x에서 (1D 다항식 계산) VI 와 (2D 다항식 계산) VI 는 더 이상 **함수** 팔레트에 없습니다. 대신에 (다항식 계산) VI 를 사용하십시오.
- LabVIEW 8.x에서 (1D 직각좌표를 극좌표로) 와 (2D 직각좌표를 극좌표로) VI 는 더 이상 **함수** 팔레트에 없습니다. 대신에 (실수 / 허수를 극좌표로) 함수와 (극좌표를 실수 / 허수로) 함수를 사용하십시오.
- LabVIEW 8.x에서, (하모닉 분석기) VI 는 더 이상 **함수** 팔레트에 없습니다. **THD** 나 **구성 요소 레벨** 출력을 측정할 때는 (하모닉 왜곡 분석기) VI 를 사용하고, **SINAD** 나 **THD 더하기 노이즈** 출력을 측정할 때는 (SINAD 분석기) VI 를 사용하십시오.
- LabVIEW 8.x에서, (네트워크 함수 (평균)) VI 는 더 이상 **함수** 팔레트에 없습니다. 대신에 (주파수 응답 함수 (크기 - 위상)), (주파수 응답 함수 (실수 - 허수)), (크로스 스펙트럼 (크기 - 위상)), 또는 (크로스 스펙트럼 (실수 - 허수)) VI 를 사용하십시오.
- LabVIEW 8.x에서, (펄스 파라미터) VI 는 더 이상 **함수** 팔레트에 없습니다. 대신 (변이 측정) VI 를 사용하여 **슬루 속도**, **지속기간**, **오버슛** (상응하는 (변이 측정) VI 는 **변이 이후** 출력), 또는 **프리슛** (상응하는 (변이 측정) VI 는 **변이 이전** 출력) 출력을 측정하거나, (펄스 측정) VI 를 사용하여 **주기**, **펄스 지속기간**, 또는 **주기 점유율** 출력을 측정하거나, 또는 (진폭과 레벨) VI 를 사용하여 **진폭**, **높은 상태 레벨** 또는 **낮은 상태 레벨** 출력을 측정하십시오.
- LabVIEW 8.x에서 (전달 함수) VI 는 **함수** 팔레트에 없습니다. 대신에 (주파수 응답 함수 (크기 - 위상)) 또는 (주파수 응답 함수 (실수 - 허수)) VI 를 사용하십시오.
- LabVIEW 8.x에서, NI DIAdem Report Wizard Express VI 는 **함수** 팔레트에 없습니다. 대신 (DIAdem 리포트) 익스프레스 VI 를 사용하십시오.

- LabVIEW 8.x에서, VISA 리소스 이름 상수와 IVI 논리적 이름 상수는 더 이상 **함수** 팔레트에 없습니다. VISA 리소스 이름을 지정하려면, VISA VI의 VISA 리소스 이름을 사용하십시오. IVI 논리적 이름을 지정하려면, 인스트루먼트를 초기화하는 적절한 드라이버 VI의 적절한 입력을 사용하십시오.
- LabVIEW 8.x에서 에러 링 상수는 **함수** 팔레트에 없습니다. 원하는 에러 코드를 입력하려면 32 비트 부호있는 정수를 대신 사용하십시오.
- **(Windows 와 Linux)** LabVIEW 8.x에서, LabVIEW 7.x의 **Sound** 팔레트에 있던 Sound VI가 **함수** 팔레트에 없습니다. 대신에 LabVIEW 8.x에서 사운드 VI를 사용하십시오. LabVIEW 7.x에서 함께 출시된 예제는 LabVIEW 8.x에서는 함께 출시되지 않습니다.

파일 I/O VI 와 함수

LabVIEW 8.x에서, Read Characters From File VI는 **함수** 팔레트에 없습니다. 대신에 (텍스트 파일에서 읽기) 함수를 사용하십시오.

LabVIEW 8.x에서, Open/Create/Replace File VI는 **함수** 팔레트에 없습니다. 대신에 (파일 열기 / 생성 / 대체) 함수를 사용하십시오. 다음 함수는 LabVIEW 7.x 및 이전 버전의 Open/Create/Replace File VI의 일부 기능을 포함합니다:

- (파일 크기 얻기) 함수를 사용하여 파일의 크기를 결정합니다.
- (파일 대화 상자) 익스프레스 VI를 사용하여 파일 대화 상자에 대한 파일이나 디렉토리의 시작 경로, 파일 패턴, 기본 이름을 지정합니다.
- (참조 번호를 경로로) 함수를 사용하여 참조를 경로로 변환합니다.
- (2 진 파일에 쓰기) 함수를 사용하여 플랫폼에 독립적인 텍스트 파일이나 다른 타입의 2 진 파일을 생성하고, (2 진 파일에서 읽기) 함수를 사용하여 2 진 파일의 결과값을 읽습니다.

LabVIEW 8.x에서, Read File 와 (Write File 함수는 **함수** 팔레트에 없습니다. 대신에 (2 진 파일에서 읽기) 와 (2 진 파일에 쓰기) 함수를 사용하십시오.

LabVIEW 8.x에서, Write Characters To File VI는 **함수** 팔레트에 없습니다. 대신에 (텍스트 파일 쓰기) 함수를 사용하십시오.

LabVIEW 8.x에서, Access Rights 함수는 **함수** 팔레트에 없습니다. 대신에 (권한 얻기) 와 (권한 설정) 함수를 사용합니다.

LabVIEW 8.x에서, EOF 함수는 **함수** 팔레트에 없습니다. 대신에 (파일 크기 얻기) 와 (파일 크기 설정) 함수를 사용합니다.

LabVIEW 8.x에서, List Directory 함수는 **함수** 팔레트에 없습니다. 대신에 (폴더 열거) 함수를 사용하십시오.

LabVIEW 8.x에서, Lock Range 함수는 **함수** 팔레트에 없습니다. 대신에 (접근 거부) 함수를 사용하십시오.

LabVIEW 7.x에서 생성한 블록다이어그램에서 새 디렉토리 함수를 포함하는 VI를 열 경우, LabVIEW 8.x는 해당 함수를 (폴더 생성) 함수로 대체합니다. 경로 입력에 지정된 폴더가 없으면, (폴더 생성) 함수는 (새 디렉토리) 함수가 했던 것처럼 에러를 반환하기보다는 디렉토리를 생성합니다.

LabVIEW 8.x에서, Seek 함수는 **함수** 팔레트에 없습니다. 대신에 (파일 위치 얻기) 와 (파일 위치 설정) 함수를 사용하십시오.

LabVIEW 8.x에서, Type and Creator 함수는 **함수** 팔레트에 없습니다. 대신에 (타입과 생성자 얻기) 와 (타입과 생성자 설정) 함수를 사용하십시오.

LabVIEW 8.x에서, Volume Info 함수는 **함수** 팔레트에 없습니다. 대신에 (볼륨 정보 얻기) 함수를 사용합니다.

LabVIEW 8.x에서, Open File and New File 함수는 **함수** 팔레트에 없습니다. (파일로부터 라인 읽기) VI 는 **함수** 팔레트에 없지만 호환성을 위해 LabVIEW 와 함께 출시됩니다.

LabVIEW 8.x에서, (I16 파일로부터 읽기), (SGL 파일로부터 읽기), (I16 파일에 쓰기) 와 (SGL 파일에 쓰기) VI 는 **함수** 팔레트에 없습니다. 대신에 (2 진 파일에서 읽기) 와 (2 진 파일에 쓰기) VI 를 사용하십시오.

프로퍼티, 메소드, 이벤트 동작 변경 내역

LabVIEW 7.1 또는 8.0 에서 프로퍼티, 메소드, 이벤트의 동작이 다음과 같이 변경되었습니다.

어플리케이션 프로퍼티와 메소드

LabVIEW 8.x에서 어플리케이션 프로퍼티와 메소드의 동작은 소속되어 있는 어플리케이션 인스턴스에 따라 다릅니다. 예를 들어, 어플리케이션 : 메모리상의 모든 VI 프로퍼티의 동작은 사용하는 어플리케이션 인스턴스에 따라 달라집니다. 이 프로퍼티는 해당 프로퍼티와 같은 어플리케이션 인스턴스에서 메모리에 있는 모든 VI 리스트를 반환합니다. 그러나 어플리케이션의 동작 : 디렉토리 경로 프로퍼티는 사용하는 어플리케이션 인스턴스에 따라 달라지지 않습니다. 이 프로퍼티는 어플리케이션이 위치해 있는 디렉토리에 대한 절대 경로를 반환합니다. 이 정보는 각 어플리케이션 인스턴스에서 변하지 않습니다.

어플리케이션 인스턴스에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오.

프린트패널 : 열기 메소드

LabVIEW 7.0 Open FP 메소드가 LabVIEW 7.1 에서 Old Open FP 로 변경되었습니다.

LabVIEW 7.1 에는 프린트패널이 이미 열려있는 경우 에러를 반환하지 않는 다른 Open FP 메소드가 포함되어 있습니다. LabVIEW 7.1 Open FP 메소드가 LabVIEW 8.x에서 프린트 패널 : 열기로 변경되었습니다. LabVIEW 7.0 의 Old Open FP 메소드를 사용하는 VI 가 있는 경우, 프린트패널 : 열기 메소드로 대체하십시오.

VI 실행 메소드

LabVIEW 7.1 에서 Run VI 메소드의 **Auto Dispose Ref** 입력을 참으로 설정하면, VI 가 실행을 멈춘 후에도 LabVIEW 는 자동으로 참조를 삭제합니다. Run VI 메소드가 에러를 생성하면, 참조가 자동으로 닫히지 않습니다. LabVIEW 8.0 에서는, 메소드가 에러를 반환한 경우에도 LabVIEW 가 자동으로 참조를 삭제합니다. 블록 다이어그램의 일부가 이 참조에 의존할 경우 이 동작에 의해 실행 중인 VI 가 깨질 수도 있습니다. LabVIEW 8.2 및 이후 버전에서, 이 동작은 다시 7.1 버전의 동작으로 변경되었습니다.

키 다운과 키 반복 이벤트

키 다운, 키 다운?, 키 반복, 키 반복? VI 이벤트와 컨트롤의 V 키 데이터 필드는 현재 키보드의 문자와 숫자 섹션에서 <Return> 키와 숫자 키패드에서 <Enter> 키에 대해 별개의 값을 가지고 있습니다. LabVIEW 7.x 및 이전 버전에서, <Enter> 키 또는 <Return> 키가 이 이벤트 중 하나를 생성하면, LabVIEW 는 VKey 데이터 필드에 **<Enter>** 를 반환합니다. LabVIEW 8.x 에서, <Enter> 키 또는 <Return> 키가 이 이벤트 중 하나를 생성하면, LabVIEW 는 V 키 데이터 필드에 각각 **<Enter>** 또는 **<Return>** 을 반환합니다.

(Mac OS) LabVIEW 8.x는 바로 가기 메뉴에서 <Control>-클릭만 받으며 <Command>-클릭 키 조합은 받지 않습니다. 이벤트 구조를 사용하여 이 동작을 따라하려면, 새 동작에 맞게 VI 를 수정해야 합니다.

리스트박스 프로퍼티

LabVIEW 7.x 및 이전 버전에서, 리스트박스의 최상위 행 프로퍼티를 리스트박스의 제일 아래쪽 아이템 밑의 행으로 설정하면, LabVIEW 는 마지막으로 보이는 아이템으로 행을 고정합니

다. LabVIEW 8.x에서는, 리스트박스에 보이는 아이템의 숫자로 이 프로퍼티에 연결할 수 있는 행의 숫자가 제한되지 않습니다.

LabVIEW 8.x는 단일 열 리스트박스에서 더블 클릭 프로퍼티를 지원하지 않습니다. 대신 더블 클릭된 열 열기 메소드를 사용합니다.

객체소유 VI 프로퍼티

LabVIEW 7.x 및 이전 버전에서, 객체소유 VI 프로퍼티는 객체가 속한 VI 에 대한 참조를 반환합니다. 이 참조는 VI 를 메모리에 유지합니다. LabVIEW 8.x에서는 객체소유 VI 프로퍼티가 반환하는 참조는 메모리의 VI 를 유지하지 않습니다. 객체소유 VI 가 메모리에서 삭제될 경우, 이 참조는 유효하지 않게 됩니다. (VI 참조 열기) 함수를 사용하여 참조를 명확하게 닫기 전까지 메모리에 있는 VI 에 대한 참조를 얻습니다.

텍스트 프로퍼티

LabVIEW 7.x 및 이전 버전에서, Text 프로퍼티가 문자열을 일반 디스플레이로 반환합니다. LabVIEW 8.x에서, 텍스트 프로퍼티는 프런트패널 객체와 같은 텍스트 디스플레이로 문자열을 반환합니다. 예를 들어, 문자열 컨트롤을 암호 디스플레이로 디스플레이한 경우, 텍스트 프로퍼티는 문자열을 암호 디스플레이로 반환합니다.

트리 컨트롤 프로퍼티

LabVIEW 7.x 및 이전 버전에서, Active Cell Properties:Cell Size:Height 와 Active Cell Properties:Cell Size:Width 프로퍼티는 트리 컨트롤에서 각 라인에 대해 LabVIEW 8.x보다 하나 적은 픽셀을 반환합니다. 예를 들어, LabVIEW 8.x에서 트리 컨트롤의 높이와 폭으로 70 픽셀 x 16 픽셀을 반환하는 프로퍼티 노드가 있는 LabVIEW 7.xVI 를 로드하면, 높이와 폭을 위해 놓는 모든 새로운 프로퍼티 노드는 69 픽셀 x 15 픽셀을 반환합니다.

VI 문자열 메소드

VI 문자열 반출 메소드를 사용하여 이전 버전의 LabVIEW 에서 반출하는 문자열은 VI 문자열 : 반입 메소드를 사용할 때 LabVIEW 8.x에서 적절하게 반입하지 못할 수 있습니다.

삭제된 프로퍼티, 메소드, 이벤트

LabVIEW 8.x는 다음의 프로퍼티, 메소드, 이벤트를 지원하지 않습니다.

커서 프로퍼티

LabVIEW 8.x는 Cursor Lock Style 프로퍼티를 지원하지 않습니다. 대신에 커서 모드 프로퍼티를 사용하십시오.

리스트박스, 여러 열 리스트박스, 테이블, 디지털 테이블, 트리 컨트롤 프로퍼티와 이벤트

LabVIEW 8.x는 여러 열 리스트박스에서 셀 전경색 프로퍼티를 지원하지 않습니다. 대신에 활성 셀 : 셀 폰트 : 색 프로퍼티를 사용하십시오.

LabVIEW 8.x는 테이블 또는 디지털 테이블에서 셀 전경색 프로퍼티를 지원하지 않습니다. 대신에 테이블과 디지털 테이블에 활성 셀 : 셀 폰트 : 색 프로퍼티를 사용하십시오.

LabVIEW 8.x는 트리 컨트롤에서 활성 셀 : 전경색 프로퍼티를 지원하지 않습니다. 대신에 활성 셀 : 셀 폰트 : 색 프로퍼티를 사용하십시오.

LabVIEW 8.x는 트리 컨트롤 클래스에서 끌기, 끌기?, 놓기, 놓기? 이벤트를 지원하지 않습니다. 대신에 컨트롤 클래스에서 끌기 종료, 끌기 입력, 끌기 이탈, 끌기 통과, 끌기 소스 업데이트, 끌기 시작, 끌기 시작?, 끌기 이벤트를 사용하십시오.

이름있는 숫자형 프로퍼티

LabVIEW 8.x는 이름있는 숫자 객체의 이름있는 숫자 색, 이름있는 숫자 색 : 배경색, 이름있는 숫자 색 : 텍스트 색을 더 이상 지원하지 않습니다. 텍스트 색, 텍스트 색 : 배경색, 텍스트 색 : 텍스트 색 프로퍼티를 각각 대신해서 사용하십시오.

패널 프로퍼티

LabVIEW 8.x는 구획 클래스의 색 프로퍼티를 지원하지 않습니다. LabVIEW 8.x에서 이 프로퍼티를 사용하면, 프로퍼티는 왼쪽 상단의 구획에만 적용됩니다. 대신에 구획 클래스의 구획 색 프로퍼티를 사용하십시오.

서브패널 프로퍼티

LabVIEW 8.x에서 서브패널의 SubVI 팬을 사용하여 서브패널 컨트롤 스크롤 막대의 보이기를 설정하고 서브패널 컨트롤에서 프런트패널을 스케일합니다.

LabVIEW 8.x는 서브패널 컨트롤의 X 스크롤 막대 보이기 프로퍼티를 지원하지 않습니다. 대신에 팬에 수평 스크롤 막대 보이기 프로퍼티를 사용하십시오.

LabVIEW 8.x는 서브패널 컨트롤의 Y 스크롤 막대 보이기 프로퍼티를 지원하지 않습니다. 대신에 팬에 수직 스크롤 막대 보이기 프로퍼티를 사용하십시오.

LabVIEW 8.x는 서브패널 컨트롤의 패널 스케일 프로퍼티를 지원하지 않습니다. 대신에 팬에 스케일링 모드 설정 메소드를 사용하십시오.

VI 프로퍼티, 메소드, 이벤트

LabVIEW 8.x는 프런트패널 윈도우: 자동 중심 설정 프로퍼티를 지원하지 않습니다. 대신 프런트패널: 중심 메소드를 사용하십시오.

LabVIEW 8.x는 프런트패널 윈도우: 스크린에 맞춤 프로퍼티를 지원하지 않습니다. 대신 프런트패널 윈도우: 상태 프로퍼티를 사용하십시오.

LabVIEW 8.x는 VI 클래스에서 프런트패널 윈도우: 원점 프로퍼티를 지원하지 않습니다. LabVIEW 8.x에서 이 프로퍼티를 사용하면, 프로퍼티는 왼쪽 상단의 구획에만 적용됩니다. 대신에 구획 클래스의 원형 프로퍼티를 사용하십시오.

LabVIEW 8.x는 VI 클래스에서 프런트패널 윈도우: 스크롤 막대 표시 프로퍼티를 지원하지 않습니다. LabVIEW 8.x에서 이 프로퍼티를 사용하면, 프로퍼티는 왼쪽 상단의 구획에만 적용됩니다. 대신에 구획 클래스에서 수평 스크롤 막대 보이기와 수직 스크롤 막대 프로퍼티를 사용하십시오.

LabVIEW 8.x는 VI 클래스에서 프런트패널 스케일링 모드 얻기나 프런트패널 스케일링 모드 설정 메소드를 지원하지 않습니다. LabVIEW 8.x에서 이 메소드를 사용하면, 메소드는 단지 왼쪽 상단의 구획에 적용됩니다. 대신에 구획 클래스에서 스케일링 모드 얻기와 스케일링 모드 설정 메소드를 사용하십시오.

LabVIEW 8.x에서는 **이벤트 편집** 대화 상자에서 VI 클래스의 마우스 다운, 마우스 다운?, 마우스 이동 또는 마우스 업 이벤트를 선택할 수 없습니다. 대신에 구획 클래스에서 각각 마우스 다운, 마우스 다운?, 마우스 이동, 마우스 업 이벤트를 사용하십시오.

어플리케이션 빌더의 이전 버전에서 변경된 사항

어플리케이션 빌더는 **프로젝트 탐색기** 윈도우에 통합됩니다. LabVIEW Base Package 또는 Full Development System 을 사용하는 경우, National Instruments 웹 사이트 ni.com/info 를 방문하고 정보 코드 rd1v21 을 입력하여 어플리케이션 빌더를 별도로 구매할 수 있습니다.

프로젝트 탐색기 윈도우의 **빌드 스펙**을 사용하여 독립 어플리케이션 (EXE), 공유 라이브러리 (DLL), zip (압축) 파일을 만들고 이에 대한 빌드 스펙을 생성합니다. **(Windows) 또한, 빌드 스펙**을 사용하여 설치 프로그램을 만들고 이에 대한 빌드 스펙을 만들 수 있습니다. 빌드 스펙은 어플리케이션 빌더 이전 버전의 .b1d 파일과 같으나, 이제 개별 파일이 아닌 LabVIEW 프로젝트의 일부입니다.



노트 프로젝트 내에서는 어플리케이션 빌더 도구를 사용해야 합니다 .

.bld 파일을 변환하여 새 프로젝트의 빌드 스펙으로 만들 수 있습니다 . **도구»빌드 스크립트 변환**을 선택하여 변환할 .bld 파일을 탐색하고 선택합니다 . LabVIEW 는 이 파일을 사용하여 소스 파일과 빌드 스펙을 포함하는 새 프로젝트를 생성합니다 .

어플리케이션 아이템 태그

다음 리스트는 더 이상 지원되지 않거나 다른 기능과 결합되어 LabVIEW 에서 제거된 어플리케이션 아이템 태그를 포함합니다 :

- APP_SAVE_WITH_OPTIONS
- APP_UPDATE_VXI
- APP_DSC_TOOLBAR
- APP_DSC_TAGEDITOR
- APP_DSC_TAGMONITOR
- APP_DSC_HTV
- APP_DSC_ENGINE
- APP_DSC_SECURITY
- APP_DSC_LOGOUT
- APP_DSC_CPWD
- APP_DSC_USERINFO
- APP_DSC_USEREDITOR
- APP_DSC_ADVANCED
- APP_DSC_STARTUP
- APP_DSC_SRVBRW
- APP_DSC_IST
- APP_DSC_IMAGENAV
- APP_DSC_OPTIONS
- APP_SRC_CODE_CTRL
- APP_BUILD_STANDALONE_APP
- APP_EDIT_VI_LIBRARY
- APP_DN_ASSEMBLY_REFS
- APP_SHOW_CLIPBOARD
- APP_VIEW_PRINTED_MANUALS
- APP_RT_ENGINE_INFO
- APP_SWITCH_EXEC_TARGET
- APP_REALTIME

이전 버전의 LabVIEW 에서 저장한 런타임 메뉴 (.rtm) 파일을 사용하고 사용하는 파일에 삭제된 태그가 있는 경우 , **메뉴 편집기** 대화 상자에서 파일을 저장할 때 , LabVIEW 8.x는 자동으로 .rtm 파일에서 해당 태그를 삭제합니다 . 삭제된 어플리케이션 아이템 태그는 LabVIEW 에 의해 지정되므로 사용자 태그로 사용될 수 없습니다 .

HiQ 지원

National Instruments 는 8.x에서 HiQ 기능을 지원하지 않습니다 . 어플리케이션이 HiQ VI 를 사용하는 경우 , 이를 수학과 신호 처리 VI 로 대체하는 것을 고려하십시오 .

에러 리스트 윈도우

LabVIEW 7.x 및 이전 버전에서, **Error list** 윈도우의 **VI List** 섹션은 메모리상의 모든 VI의 에러를 보여줍니다. LabVIEW 8.x에서, **에러 리스트** 윈도우의 **에러가 있는 아이템** 섹션은 VI 및 라이브러리와 같은 메모리 상의 모든 아이템의 에러를 보여줍니다. 이름이 같은 아이템이 두 개 이상인 경우, 이 섹션은 각 모호한 아이템에 대한 특정 어플리케이션 인스턴스를 보여줍니다. 어플리케이션 인스턴스에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오.

VI 문자열 파일 구분

LabVIEW 8.x는 **도구 > 고급 > 문자열 반입**을 선택하거나 VI 문자열; 반입 메소드를 사용하여 VI 문자열 파일을 반입할 때, 새 태그의 세트인 <GROUPER></GROUPER>를 검색합니다. 이 태그 세트는 함께 그룹화된 프론트패널 객체를 나타냅니다. 따라서 LabVIEW 8.x에서, 이전 버전의 LabVIEW에서 저장한 VI 문자열 파일을 반입할 수 없습니다.

LabVIEW 7.1 및 이전 버전은 리스트박스 문자열을 프라이빗 데이터의 <ITEMS> 섹션에 나열합니다. LabVIEW 8.x 리스트박스 문자열을 프라이빗 데이터의 <STRINGS> 섹션에 나열합니다. 또한, LabVIEW 7.1 및 이전 버전에서 리스트박스는 한가지 폰트만을 사용할 수 있습니다. LabVIEW는 이를 프라이빗 데이터의 <LBLABEL> 섹션에 나열합니다. LabVIEW 8.x에서, 리스트박스는 여러가지 폰트를 사용할 수 있습니다. LabVIEW는 이를 프라이빗 데이터의 <CELL_FONTS> 섹션에 나열합니다.

LabVIEW 7.1 및 이전 버전은 여러 열 리스트박스 문자열을 기본 데이터에 나열합니다. 그러나, 여러 열 리스트박스의 기본 데이터는 정수 또는 정수의 배열입니다. LabVIEW 8.x는 여러 열 리스트박스 문자열을 프라이빗 데이터에 보여줍니다.

LabVIEW 7.1 및 이전 버전은 트리 컨트롤에 문자열이나 폰트를 반출하지 않습니다. LabVIEW 8.x는 트리 컨트롤 문자열과 폰트 모두를 반출할 수 있으며, 리스트박스 및 여러 열 리스트박스와 동일한 포맷으로 문자열과 폰트를 반출합니다.

LabVIEW 8.x에서 반출 파일의 각 라인은 적어도 프라이빗 혹은 기본 데이터에 해당하는 두 개의 태그를 포함합니다. 또한, LabVIEW 8.x는 각 레벨에서 아이템을 한단계 아래로 이동시킵니다.

다음 단계를 따라 VI 문자열 파일을 LabVIEW 8.x 포맷으로 변환합니다.

1. 이전 버전의 LabVIEW에서 VI 문자열 파일을 반입합니다.
2. VI를 저장합니다.
3. LabVIEW 8.x에서 VI를 로드합니다.
4. **도구 > 고급 > 문자열 반출**을 선택하여 VI 문자열 파일을 LabVIEW 8.x 포맷으로 저장합니다.

LabVIEW 7.x로 (혹은 로부터) 타입 설명자 데이터 변환하기

LabVIEW 8.x에서 LabVIEW가 타입 설명자를 저장하는 포맷이 변경되었습니다. LabVIEW 7.x에서는 16비트의 패턴 형태로 타입 설명자를 저장합니다. LabVIEW 8.x에서는 32비트의 패턴 형태로 타입 설명자를 저장합니다. 따라서 타입 설명자 크기가 64KB로 제한되지 않습니다.

LabVIEW 8.x는 LabVIEW 7.x에서 작성한 타입 설명자를 읽고 LabVIEW 7.x가 읽을 수 있는 타입 설명자를 쓰는 메커니즘을 제공합니다. (패턴화된 문자열로) 함수에는 **7.x 데이터 변환** 바로 가기 메뉴 아이템이 있습니다. 이 함수에서 마우스 오른쪽 버튼을 클릭하고 이 메뉴 항목을 선택할 경우, 함수는 입력 데이터를 LabVIEW 7.x에서 쓰여진 것과 같이 처리합니다. **7.x 데이터 변환** 바로 가기 메뉴 아이템을 선택하고 데이터 문자열 출력이 연결되면, LabVIEW 8.x는 함수에 붉은색 7.x 문양을 표시하여 데이터가 LabVIEW 7.x 포맷으로 변환되거나 LabVIEW 7.x 포맷으로부터 변환된 것을 나타냅니다. 데이터가 변환되지 않게 하려면, 바로 가기 메뉴에서 **7.x 데이터 변환**을 다시 선택하여 확인 표시를 제거하십시오.

LabVIEW 8.x에서 LabVIEW 7.x 또는 이전 버전에서 마지막으로 저장한 VI를 로드하면, LabVIEW 8.x는 자동으로 (패턴화된 문자열로) 함수에 **7.x 데이터 변환** 속성을 설정합니다. 이 함수는 계속해서 LabVIEW 7.x 및 이전 버전에서처럼 동작합니다. VI가 LabVIEW 8.x 타입 설명자 포맷을 사용하게 하려면, (패턴화된 문자열로) 함수에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **7.x 데이터 변환**을 선택하여 확인 표시를 제거하십시오. VI가 LabVIEW 7.x 또는 이전 버전에서 쓰여진 데이터가 있는 파일을 수정할 필요가 없거나, LabVIEW 7.x 또는 이전 버전에서 실행되는 VI로 데이터를 제공하거나 받지 않는 경우, LabVIEW 8.x의 타입 설명자 포맷을 사용해야 합니다. 이후 버전의 LabVIEW에서는 이전 타입의 설명자 포맷을 지원하지 않을 수 있습니다.

LabVIEW 내장 소스 컨트롤 제공자에서 이동하기

LabVIEW 7.x 및 이전 버전의 내장 소스 컨트롤 제공자는 LabVIEW 8.x에서 사용할 수 없습니다. LabVIEW에서 소스 컨트롤을 사용하려면 반드시 타사 소스 컨트롤 제공자를 선택해야 합니다. 이전 버전에서 내장되어 있는 제공자를 사용한 경우, LabVIEW에서 소스 컨트롤을 사용하려면 파일을 다른 제공자로 이동해야 합니다. LabVIEW에서 지원하는 타사 소스 컨트롤 제공자의 최신 리스트를 찾으려면 National Instruments 웹 사이트 ni.com/info에서 정보 코드 `exgucn`을 입력하십시오.

파일을 새 소스 컨트롤 제공자로 이동하면, 내장되어 있는 제공자에서 저장한 개정 히스토리를 잃게 됩니다. 이전 버전의 파일을 새 제공자로 이동할 수 없습니다.

다음 단계를 따라 내장 소스 컨트롤 제공자에서 타사 소스 컨트롤 제공자로 파일을 이동하십시오.

1. 이전 버전의 LabVIEW에서, 모든 사용자가 LabVIEW 내장 소스 컨트롤 제공자에 포함된 파일을 체크 인했는지 확인합니다.
2. 파일을 새 소스 컨트롤 제공자에 추가하려는 컴퓨터에서, 내장되어 있는 제공자를 사용하여 모든 파일의 최신 버전을 얻습니다.
3. 내장되어 있는 제공자를 사용하여 소스 컨트롤에서 파일을 체크 아웃합니다.
4. 타사 소스 컨트롤 제공자에서, 새 소스 컨트롤 프로젝트에 사용하려는 셋팅을 설정합니다.
5. LabVIEW가 타사 소스 컨트롤 제공자와 함께 작동하도록 설정합니다.

LabVIEW가 타사 소스 컨트롤 제공자와 함께 작동하도록 설정하는 것에 대한 정보는 *LabVIEW 도움말*의 **목차** 탭에서 **기본 > 프로젝트 구성하기와 관리하기 > 사용법 > LabVIEW에서 소스 컨트롤 사용하기** 모음을 참조하십시오.

6. LabVIEW 프로젝트를 생성합니다. 내장되어 있는 제공자에 포함된 파일을 프로젝트에 추가합니다. LabVIEW가 입력 요청을 하면 파일을 소스 컨트롤에 추가합니다. 또한, 타사 제공자로부터 파일을 직접 추가할 수 있습니다.

LabVIEW 프로젝트 생성에 대한 정보는 *LabVIEW 도움말*의 **목차** 탭에서 **기본 > 프로젝트 구성하기와 관리하기 > 사용법 > LabVIEW 프로젝트 생성하기** 모음을 참조하십시오.

NaN 문자열을 정수 타입으로 변환하기 (Windows)

LabVIEW 7.x에서 명백하게 또는 함축적으로 NaN을 정수로 변환할 때, 값은 해당 정수 데이터 타입의 가장 작은 값이 됩니다. 예를 들어, NaN을 16비트 부호있는 정수로 변환하면 16비트 부호있는 정수의 가장 작은 값인 -32,768이 생성됩니다.

LabVIEW 8.x에서 명백하게 또는 함축적으로 NaN을 정수로 변환할 때, 값은 해당 정수 데이터 타입의 가장 큰 값이 됩니다. 예를 들어, NaN을 16비트 부호있는 정수로 변환하면 16비트 부호있는 정수의 가장 큰 값인 32,767이 생성됩니다.

케이스 구조에 연결된 상수

LabVIEW 7.x 및 이전 버전에서, 상수를 케이스 구조에 연결하고, 실행되지 않는 케이스에 SubVI 를 놓아 메모리에 SubVI 를 보존할 수 있습니다. 예를 들어, 참 상수를 케이스 구조에 연결하고, SubVI 를 케이스 구조의 거짓 케이스에 놓은 경우, LabVIEW 는 VI 호출하면서 SubVI 를 로드합니다. LabVIEW 8.x 는 실행되지 않은 모든 코드를 제거합니다. 따라서 이전 버전의 LabVIEW 에서 저장한 VI 를 케이스 구조에 연결된 상수와 함께 LabVIEW 8.x 에서 로드하는 경우, LabVIEW 는 상수를 숨겨진 컨트롤로 변경하여 이전 버전의 LabVIEW 동작을 유지합니다.

OS 메시지 지연

LabVIEW 7.x 에서, LabVIEW 는 .NET 과 ActiveX 이벤트를 처리하기 위해 콜백 VI 를 실행하는 동안 OS 메시지를 처리합니다. LabVIEW 8.x 에서 LabVIEW 는 콜백 VI 가 실행을 멈추거나 모달 대화 상자를 로드할 때까지 OS 메시지의 처리를 지연시킵니다. 이같은 지연으로 콜백 VI 가 간섭없이 실행되고, LabVIEW 가 다른 이벤트 내에서 이벤트를 실행시켜 교착 상태에 빠지는 것을 방지할 수 있습니다.

콜백 VI 로부터 모달이 아닌 대화 상자에 대해 동기적으로 호출할 수 없습니다. 모달이 아닌 대화 상자에서 VI 실행 메소드를 실행하고 이 메소드의 완료까지 기다림 입력에 거짓 볼리언 상수를 연결하여 콜백 VI 로부터 비동기적으로 모달이 아닌 대화 상자를 호출해야 합니다.

LabVIEW 7.x 에서, LabVIEW 는 DLL 이나 공유 라이브러리 함수를 실행하는 동안 OS 메시지를 처리합니다. LabVIEW 8.x 에서 LabVIEW 는 DLL 함수에 대한 호출이 끝날 때까지 또는 DLL 로부터 모달 대화 상자를 로드할 때까지 OS 메시지의 처리를 지연시킵니다. 이같은 지연으로 DLL 함수가 간섭없이 실행되고, LabVIEW 가 DLL 함수가 실행되고 있는 동안 DLL 을 호출하여 교착 상태에 빠지는 것을 방지할 수 있습니다. 공유 라이브러리 함수가 실행되는 동안 같은 공유 라이브러리 파일의 호출을 방지하기 위해 키보드 메시지와 같은 OS 메시지를 지연하는 것이 유용합니다. 예를 들어, 사용자가 버튼을 누르는데 대한 응답으로 공유 라이브러리 함수가 호출되는 경우, 공유 라이브러리 함수가 완료될 때까지는 사용자가 버튼을 누를 수 없도록 해야 합니다.

이 기본 동작을 사용하면 DLL 이 실행되는 동안 모달이 아닌 대화 상자에 대해 동기적으로 호출할 수 없습니다. 모달이 아닌 대화 상자에서 VI 실행 메소드를 실행하고 이 메소드의 완료까지 기다림 입력에 거짓 볼리언 상수를 연결하여 DLL 로부터 비동기적으로 모달이 아닌 대화 상자를 호출해야 합니다.

생성한 DLL 에서 OS 메시지를 지연시킬지를 선택할 수 있습니다. **프로젝트 탐색기** 윈도우의 DLL 에서 마우스 오른쪽 버튼을 클릭하고, 바로 가기 메뉴에서 **프로퍼티**를 선택하고, **항목** 리스트에서 **고급**을 선택하고, **공유 라이브러리에서 OS 메시지 지연** 확인란에서 확인 표시를 제거하여 DLL 함수가 실행되고 있는 동안 OS 메시지를 처리합니다.

리소스 관리자 (Mac OS)

LabVIEW 7.x 및 이전 버전은 Macintosh 리소스 파일을 읽고 쓸 수 있는 문서화되지 않은 기능을 제공합니다. LabVIEW 8.x 에서는 이 메소드가 더 이상 존재하지 않습니다. 이 문서화되지 않은 기능을 사용하는 유틸리티가 더 이상 작동하지 않으므로 VI 에서 Macintosh 리소스 파일을 읽거나 쓸 수 없습니다.

단일 및 두 개 버튼 대화 상자

LabVIEW 7.x 및 이전 버전에서, 단일 버튼 대화 상자 또는 두 개 버튼 대화 상자를 디스플레이하는 VI 를 프로그램적으로 강제 종료할 수 없습니다. LabVIEW 8.x 에서, 강제 종료 VI 메소드를 사용하여 이러한 대화 상자를 디스플레이하는 VI 를 프로그램적으로 강제 종료할 수 있습니다.

프로퍼티와 인보크 노드

LabVIEW 7.x의 커서 범례로부터 내부적으로 링크된 프로퍼티 노드나 인보크 노드를 생성하면, LabVIEW 8.x에서 VI를 열 때 LabVIEW는 노드를 삭제합니다.

공유 라이브러리 업데이트하기

labview.lib에 링크되어 있는 LabVIEW 7.x 또는 이전 버전에서 공유 라이브러리 (DLL)를 만들면, LabVIEW 8.x에서는 그 공유 라이브러리를 labviewv.lib에 연결합니다. 공유 라이브러리를 labviewv.lib에 링크하는 것에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오.

인쇄를 위한 여백 값

LabVIEW 7.x 및 이전 버전에서 **Options** 대화 상자의 **Printing** 페이지에서 지정한 **Margins** 옵션은 여백 값에 대해 센티미터를 사용합니다. LabVIEW 8.x에서 **여백** 옵션은 여백 값에 대해 밀리미터를 사용합니다.

LabVIEW 6.x에서 업그레이드하기

LabVIEW 6.x에서 LabVIEW 8.6으로 업그레이드할 때 다음과 같은 호환성 문제가 발생할 수 있습니다. 업그레이드할 때 발생할 수 있는 유의사항에 대한 추가적인 정보는 이 문서의 *LabVIEW 7.x에서 업그레이드하기*, *LabVIEW 8.0에서 업그레이드하기*, *LabVIEW 8.2에서 업그레이드하기*, *LabVIEW 8.5에서 업그레이드하기* 섹션을 참조하십시오.

각 버전의 새로운 기능 및 변경사항에 대한 추가적인 정보는 ni.com/manuals에서 LabVIEW 버전 6.x부터 8.6까지 각각의 버전에 대한 *LabVIEW 업그레이드 노트*를 참조하십시오.

웨이브폼 데이터 타입의 변경

LabVIEW 7.0에서, 웨이브폼 데이터 타입은 **to** 구성요소에 대해 배정도 부동소수가 아닌 타임스탬프 데이터 타입을 사용합니다. LabVIEW 6.x에서, 데이터 형태에 대한 정보를 포함하지 않고 웨이브폼 데이터 타입의 데이터를 파일로 저장했다면, LabVIEW 7.x 및 이후 버전에서, 데이터를 가져올 때 에러가 발생할 수 있습니다.

LabVIEW 7.x 및 이후 버전에서, **Read Waveform from File VI**는 파일의 예전 웨이브폼 데이터 타입 포맷을 새로운 웨이브폼 데이터 타입 포맷으로 변환합니다. 이 VI는 변환을 수락할 것인지에 대해 입력 요청하는 대화 상자를 디스플레이합니다. LabVIEW 런타임 엔진에서 **Read Waveform from File VI**는 변환을 수행하지 못하고 대신 에러를 반환합니다. LabVIEW 6.x에서 LabVIEW 7.x 및 이후 버전으로 웨이브폼 데이터 이동에 대한 추가적인 정보는 National Instruments 웹 페이지 ni.com/info에서 정보 코드 `exd9zq`를 입력하여 참조하십시오.

시리얼 호환 VI

LabVIEW 7.x 및 이후 버전에서, 시리얼 호환 VI는 **함수** 팔레트에 없습니다. 시리얼 장비와 통신하는 VI를 작성하려면 **VISA VI**와 **함수**를 사용하십시오.

LabVIEW 7.x 및 이후 버전에서 LabVIEW는 OS의 시리얼 드라이버와 통신하기 위해 `serpdrv` 드라이버를 사용하지 않습니다. LabVIEW는 **VISA**를 기반으로 하는 호환가능한 VI를 포함합니다. 새 어플리케이션인 경우 **VISA**, 시리얼 VI와 **함수**를 사용하여 시리얼 장비를 컨트롤하십시오. **Serial VI**를 포함한 이전 버전의 LabVIEW에서 작성한 모든 VI는 LabVIEW 7.1 및 이후 버전에서도 동작합니다.

포트 번호의 매핑을 포트로 다시 설정한 경우, 매핑을 반드시 이 포트로 지정해 주어야 합니다. 시리얼 포트 매핑을 지정하려면 `labview\vi.lib\Instr_sersup.lib`의 **Set Serial Alias Ports VI**를 사용하십시오. 이 VI의 **VISA 가명** 입력에 문자열 배열을 연결하고, 입력 배열에 사용할 포트의 이름을 입력합니다. 배열의 각 원소는 하나의 포트에 대응해야 합니다. 예를 들어, 포트 0이 **VISA 가명**인 `MySerialPort`에 매핑하도록 설정한 경우, **VISA 가명** 입력 배열의

첫 원소에 MySerialPort 를 입력합니다. 이런 경우, (VISA 시리얼 포트 설정) VI 를 호출하기 전에 반드시 set serial alias ports VI 를 먼저 호출해야 합니다.

VISA VI 및 함수를 사용하여 시리얼 인스트루먼트를 컨트롤하는 예제는 labview\examples\instr\smp1ser1.llb 를 참고하십시오.

루프의 기본 데이터

LabVIEW 6.0 및 이전 버전에서, For 루프가 실행되지 않을 경우 정의되지 않은 데이터를 발생시켰습니다. LabVIEW 6.1 및 이후 버전에서, For 루프는 카운터 터미널에 0 을 연결하거나, 빈 배열을 입력으로 For 루프에 연결하여 오토인덱싱을 활성화하면 기본값을 만듭니다. 루프는 실행되지 않고 오토인덱싱이 비활성화된 모든 출력 터미널에는 이 터미널 데이터 타입에 대한 기본값이 있습니다.

리모트 프론트패널 라이선스

LabVIEW Full Development System 과 어플리케이션 빌더는 한 명의 클라이언트가 원격으로 프론트패널을 모니터링하고 컨트롤할 수 있는 리모트 프론트패널 라이선스를 포함합니다.

LabVIEW Professional Development System 은 다섯 명의 클라이언트가 원격으로 프론트패널을 보고 컨트롤할 수 있도록 하는 리모트 프론트패널 라이선스를 포함합니다.

더 많은 수의 클라이언트가 원격으로 리모트 프론트패널을 사용할 수 있도록 업그레이드할 수 있습니다.

멀티스레드 할당

LabVIEW 7.1 및 이후 버전은 VI 를 실행하는데 있어서 LabVIEW 7.1 이전 버전보다 더 많은 스레드를 할당합니다. 이러한 변경 때문에, 호출한 DLL 이 실제 재호출이 아닌 경우 라이브러리 함수 호출 노드를 재호출로 잘못 지정하면 멀티스레드 에러가 발생합니다. 라이브러리 함수 호출 노드와 재호출에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오.

LabVIEW 가 스레드를 할당하는 방법을 변경하려면 labview\vi.lib\Utility\sysinfo.llb 의 threadconfig VI 를 사용하십시오. 또한, **파일** >> **VI 프로퍼티**를 선택하고, **항목** 풀다운 메뉴에서 **실행**을 선택한 후, **재호출 실행** 확인란에서 확인 표시를 제거하여 VI 에 대한 재호출을 비활성화시킬 수 있습니다.

스레드 할당에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오.

인스트루먼트 드라이버

LabVIEW 7.x 또는 이후 버전의 LabVIEW 패키지에는 인스트루먼트 드라이버가 포함된 LabVIEW 인스트루먼트 드라이버 라이브러리 CD 가 더 이상 포함되지 않습니다. National Instruments Instrument Driver Network 인 ni.com/idnet 에서 인스트루먼트 드라이버를 다운로드하십시오. National Instruments 디바이스 드라이버 CD 는 NI-DAQ, NI-VISA, 기타 National Instruments 드라이버를 포함합니다.

단위와 변환율

LabVIEW 7.x 및 이후 버전에서, Compound Arithmetic 함수를 사용하게 되면 여분의 단위를 제거하기 위해 Convert Unit 함수를 더 이상 사용할 필요가 없습니다.

LabVIEW 7.1 및 이후 버전에서, 변환율은 National Institute for Standards and Technology (NIST) 가 *국제단위계 (SI) 사용법에 대한 가이드*에서 제시한 가이드 라인을 더욱 정확하게 준수합니다. 또한, **칼로리** 단위는 이제 **칼로리 (열)** 로, **마력**은 **마력 (전기)** 으로 사용됩니다. 이 단위의 약자는 변경되지 않았습니다. 다음 테이블은 LabVIEW 6.1 과 7.x 및 이후 버전 사이에서 변경된 단위 변환율을 설명합니다.

단위	6.1 의 정의	7.x 및 이후 정의
천문학 단위 (AU)	149,498,845,000 m	149,597,900,000 m
영국 온도 단위 (평균)	1055.79 J	1055.87 J
전자 볼트 (eV)	1.602e-19 J	1.60217642e-19 J
푸트캔들 (조명도)	10.764 lx	10.7639 lx
마력 vs. 마력 (전기)	745.7 W	746 W. 새 변환값이 정확합니다 .
영국 갤런	4.54596 l	4.54609 l
광년	9.4605 Pm	9.46073 Pm
파운드	4.448 N	4.448222 N
로드 (길이)	16.5 ft	5.029210 m
슬러그 (질량)	32.174 lb	14.59390 kg
합쳐진 원자 질량 (u)	1.66057e-27 kg	1.66053873e-27 kg

패널 업데이트 연기 프로퍼티

LabVIEW 6.1 및 이전 버전에서, 변경이 적용되지 않은 프런트패널 객체를 다시 그리려면, LabVIEW 는 패널 업데이트 연기 프로퍼티가 거짓으로 되기까지 기다렸습니다. LabVIEW 7.0 및 이후 버전에서, 이 프로퍼티를 참으로 설정하면, LabVIEW 는 변경이 적용되지 않은 모든 프런트패널 객체를 다시 그린 후, 새로운 프런트패널 업데이트 요청을 모두 연기합니다. 일부 경우에, 이같은 변경으로 LabVIEW 가 프런트패널에서 수정된 객체를 필요 이상으로 다시 그릴 수 있습니다.

숫자 컨트롤의 데이터 범위

LabVIEW 6.1 및 이전 버전에서, 특정 숫자 컨트롤은 기본 최소값이 0.00, 최대값이 0.00, 증가값이 0.00, 그리고 범위 이탈 작업이 **ignore** 입니다. LabVIEW 7.x 및 이후 버전에서, 숫자 컨트롤은 데이터 타입의 기본 데이터 범위값을 이용합니다.

강제 변환점과 타입 정의

LabVIEW 6.1 및 이후 버전에서, 와이어는 타입 정의에 대한 정보를 포함하므로, 블록다이어그램에서 더 많은 강제 변환점을 확인할 수 있습니다. 타입 정의된 요소를 타입이 정의된 터미널이 아닌 VI 또는 함수 터미널에 연결할 경우, 강제 변환점이 나타납니다. 또한, 타입 정의된 출력 터미널을 타입 정의되지 않은 인디케이터에 연결해도 강제 변환점이 나타납니다. 이러한 강제 변환점은 VI 에서 일관되게 타입 정의를 사용하지 않은 부분을 나타냅니다. 이 경우에 강제 변환점은 런타임 성능에 영향을 주지 않습니다.

타입 정의 패턴화에 (패턴화된 문자열로) 함수를 사용하는 데 대한 정보는 *LabVIEW 도움말*을 참조하십시오.

파일 대화 상자의 버튼 라벨

LabVIEW 6.1 및 이전 버전에서, File Dialog 함수가 디스플레이하는 파일 대화 상자에는 사용자가 새 파일 이름을 입력할 수 있을 때 **저장** 버튼 라벨이 나타납니다. 그렇지 않으면, 버튼 라벨은 **열기**입니다. LabVIEW 8.x에서, (파일 대화 상자) 익스프레스 VI 가 디스플레이하는 파일 대화 상자의 버튼 라벨은 라벨을 변경한 경우를 제외하고는 모든 경우에서 **확인**입니다. (파일 대화 상자) 익스프레스 VI 의 **버튼 라벨** 입력을 사용하여 버튼의 라벨을 변경합니다. 기존의 VI 에서 (파일 대화 상자) 익스프레스 VI 를 사용할 경우, VI 의 동작을 다시 검토하여 **확인**으로 표시되는 기본 라벨이 VI 의 기능에 적절한지를 확인하십시오.

온라인 도움말 컨트롤 함수

(온라인 도움말 컨트롤) 함수의 **도움말 파일의 경로** 입력은 이제 필수입니다 . 컴파일된 도움말 파일 이름 (.chm 또는 .hlp) 또는 컴파일된 도움말 파일의 전체 경로를 입력에 연결할 수 있습니다 . 컴파일된 도움말 파일 이름만 연결할 경우 , LabVIEW 는 labview\help 디렉토리에서 이 파일을 검색합니다 .

로드시 프론트패널 디스플레이

LabVIEW 7.x 및 이후 버전에서 , LabVIEW 가 VI 를 로드할 때 VI 의 프론트패널이 디스플레이 되도록 설정하고 VI 서버를 사용하여 VI 를 로드하면 , LabVIEW 는 프론트패널을 디스플레이하지 않습니다 . 프로그램적으로 프론트패널을 디스플레이하려면 프론트패널 : 열기 메소드를 반드시 사용해야 합니다 .

VI 참조 열기 함수

LabVIEW 6.1 및 이전 버전에서 , Open VI Reference 함수의 options 파라미터에 값을 연결하지 않은 경우 , 템플릿이 메모리에 없는 경우 , LabVIEW 는 템플릿에서 VI 를 예로 들어 설명합니다 . 템플릿이 메모리에 있는 경우 , LabVIEW 는 템플릿의 참조를 엽니다 . LabVIEW 7.0 및 7.1 에서 , Open VI Reference 함수를 사용하여 이미 메모리에 있는 템플릿의 참조를 생성하는 경우 , options 파라미터에 0x02 를 지정해 주지 않으면 이 함수는 에러를 반환합니다 . LabVIEW 8.0 및 이후 버전에서 , (VI 참조 열기) 함수를 사용하여 템플릿의 참조를 생성하는 경우 , LabVIEW 는 템플릿이 이미 메모리에 있더라도 템플릿에서 VI 를 예로 들어 설명합니다 .

지수형

LabVIEW 6.0 및 이전 버전에서 , ^ 연산자는 수식 노드에서 지수를 나타냅니다 . LabVIEW 6.1 및 이후 버전에서 , 지수 연산자는 ** 입니다 (예를 들어 , x**y) . ^ 연산자는 배타적 비트 또는 (XOR) 연산을 나타냅니다 .

IVI 설정 저장 파일

현재 IVI 설정 저장 파일 포맷에서는 모든 이름은 대소문자 구분입니다 . 어플리케이션에서 논리 이름 , 드라이버 세션 이름 , 또는 버추얼 이름을 사용한 경우 , 사용한 이름이 IVI 설정 저장 파일에서 정의한 이름과 대소문자까지 정확하게 일치하는지 확인하십시오 .

LabVIEW 5.x 또는 이전 버전에서 업그레이드하기

LabVIEW 5.x 또는 이전 버전에서 LabVIEW 8.6 으로 업그레이드하는 것에 대한 추가적인 정보는 National Instruments 웹 사이트 ni.com/info 에서 정보 코드 ext8h9 를 입력하십시오 . 이 사이트에서는 이전 버전의 *LabVIEW 업그레이드 노트*와 LabVIEW 8.6 에서 동작하도록 VI 를 변환하는 것에 대한 추가적인 정보에 대해 설명합니다 .

LabVIEW 8.6 의 특징과 변경 내역

프로그래밍 개념과 단계별 해설 , 참조 정보 등 LabVIEW 8.6 기능에 대한 추가적인 정보는 *LabVIEW 도움말*을 참조하십시오 . **도움말** > **LabVIEW 도움말 검색**을 선택하여 *LabVIEW 도움말*에 접근합니다 .

유의사항 , 수정된 일부 버그 리스트 , 추가적인 호환성 문제 , LabVIEW 8.6 의 최신 추가 기능에 대한 자세한 내용은 labview 디렉토리의 readme.html 을 참조하십시오 .

LabVIEW 설치하기

(Windows) LabVIEW 8.6 에서는 LabVIEW 플랫폼 DVD 를 통해 LabVIEW 를 설치하고 모듈과 툴킷을 선택할 수 있습니다 . 추가적인 정보는 이 문서의 *모듈 , 툴킷 , 인스트루먼트 드라이버 업그레이드하기* 섹션 또는 *LabVIEW 릴리즈 노트*의 *LabVIEW 8.6 설치하기* 섹션을 참조하십시오 .

LabVIEW 문서

LabVIEW 도움말 카드는 키보드 바로 가기 키, 온라인 리소스, 문서 사용법을 포함하도록 재구성되었습니다.

새 예제 VI

LabVIEW 8.x에 추가된 예제 VI들의 설명을 보고 실행하기 위해서는 NI 예제 탐색기의 **탐색** 탭에서 **LabVIEW 8.x**의 새로운 예제 폴더를 참고하십시오.

블록다이어그램 개선사항

LabVIEW 8.6의 블록다이어그램과 관련 기능에서 다음과 같이 개선되었습니다.

익스프레스 VI 생성하고 편집하기

도구 > **고급** > **익스프레스 VI 생성 또는 편집**을 선택하여 **익스프레스 VI 생성 또는 편집** 대화 상자를 사용하여 익스프레스 VI를 생성하거나 편집합니다. 기존 VI, 다른 익스프레스 VI, 또는 빈 VI로부터 익스프레스 VI를 생성할 수 있습니다. 이전 버전의 LabVIEW에서는 LabVIEW Express VI Development Toolkit이 설치된 경우에만 이 기능을 사용할 수 있었습니다.

LabVIEW에서 익스프레스 VI 생성하기에 대한 추가적인 정보는 *LabVIEW* 도움말의 **목차**에서 **기본** > **블록다이어그램 만들기** > **개념** > **익스프레스 VI** 모음을 참조하십시오.

브레이크포인트 관리자 윈도우를 사용하여 브레이크포인트 관리하기

블록다이어그램에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **브레이크포인트** > **브레이크포인트 설정**을 선택하여 브레이크포인트를 생성합니다. 브레이크포인트를 물리적으로 제거하지 않은 채로 브레이크포인트를 비활성화하고 브레이크포인트에서 VI가 정지하는 것을 막으려면, 브레이크포인트가 있는 객체에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **브레이크포인트** > **브레이크포인트 비활성화**를 선택합니다. 비활성화된 브레이크포인트가 있는 객체에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **브레이크포인트** > **브레이크포인트 활성화**를 선택하여 비활성화된 브레이크포인트를 활성화할 수 있습니다. 브레이크포인트를 제거하려면, 브레이크포인트가 있는 객체에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **브레이크포인트** > **브레이크포인트 지우기**를 선택합니다.

브레이크포인트 관리자 윈도우를 사용하여 VI에 있는 브레이크포인트를 관리하십시오. **브레이크포인트 관리자** 윈도우를 사용하여 한번에 전체적으로 또는 개별적으로 VI 계층구조에 있는 브레이크포인트를 비활성화, 활성화, 제거할 수 있습니다. **브레이크포인트 관리자** 윈도우에 있는 브레이크포인트에서 더블 클릭하여 특정 브레이크포인트의 블록다이어그램 상에서의 위치를 찾을 수 있습니다.

보기 > **브레이크포인트 관리자**를 선택하여 **브레이크포인트 관리자** 윈도우를 디스플레이하거나, 객체에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **브레이크포인트** > **브레이크포인트 관리자**를 선택합니다. 개별적으로 브레이크포인트를 선택하거나 **모두 선택** 버튼을 클릭하여 모든 브레이크포인트를 선택합니다. 동시에 선택한 브레이크포인트를 활성화 또는 비활성화하려면 **활성화** 또는 **비활성화** 버튼을 클릭합니다.

Timed 구조에서 공유된 복제 재호출 VI의 미리 할당 허용하기

Timed 루프 또는 Timed 시퀀스 구조 내에서 공유된 복제 재호출 VI의 복제본을 미리 할당하거나 공유하도록 LabVIEW를 설정할 수 있습니다. 예를 들어, 공유된 복제 재호출 VI를 Timed 루프 또는 Timed 시퀀스 구조 안에 놓으면, 구조 안에서 호출하는 재호출 VI의 인스턴스에 대한 복제본을 미리 할당하도록 구조를 설정할 수 있습니다. 구조 밖에서 호출하는 재호출 VI의 인스턴스는 계속해서 복제본을 공유합니다. 기본적으로, LabVIEW는 Timed 구조에 대한 복제 VI를 미리 할당합니다. 공유된 복제 재호출 VI에서 메모리 사용을 절약하려면, Timed 구조를 설정하여 필요한 대로 복제 VI를 할당합니다. Timed 루프 또는 Timed 시퀀스 구조 내에서

의 VI 할당을 설정하려면, 구조에서 마우스 오른쪽 버튼을 클릭한 후 바로 가기 메뉴에서 **공유 복제 할당**을 선택하고 다음 옵션 중 하나를 선택합니다 :

- **자동**—LabVIEW 가 구조를 **미리 할당** 또는 **필요한 만큼 할당**으로 설정할지를 해당 구조의 내용에 따라 설정합니다 . LabVIEW 는 바로 가기 메뉴에 텍스트를 추가하여 LabVIEW 가 선택하는 셋팅을 표시합니다 . 예를 들어 , LabVIEW 가 **미리 할당**을 선택하는 경우 , 메뉴 아이템은 **자동 (미리 할당)** 이 됩니다 .
- **미리 할당**—LabVIEW 가 구조 내에 있는 재호출 VI 에 대한 각각의 호출을 위한 복제 VI 를 생성합니다 . Timed 구조 또는 Timed 시퀀스 구조는 해당 아이콘을 디스플레이하여 LabVIEW 가 구조 내에 있는 모든 재호출 VI 에 대한 복제본을 미리 할당하고 있음을 표시합니다 .
- **필요한 만큼 할당**—사용자가 구조 내에 있는 재호출 VI 를 호출하기 전까지는 LabVIEW 가 복제 VI 를 생성하지 않습니다 . Timed 구조 또는 Timed 시퀀스 구조는 해당 아이콘을 디스플레이하여 LabVIEW 가 구조 내에 있는 모든 재호출 VI 에 대한 복제본을 공유하고 있음을 표시합니다 .

여러 케이스가 있는 구조에서 자동으로 터널 연결하기

LabVIEW 는 여러 케이스가 있는 구조에서 자동으로 입력 및 출력 터널을 연결할 수 있습니다 . 출력 터널에서 마우스 오른쪽 버튼을 클릭하고 **링크된 입력 터널** 바로 가기 메뉴에서 다음 옵션 중에서 하나를 선택하여 , 입력 및 출력 터널에 대한 자동 와이어링을 설정합니다 :

- **와이어 연결 없는 케이스 생성 & 연결**—LabVIEW 는 와이어로 연결하지 않은 기존 케이스 모두에 터널을 연결합니다 . LabVIEW 는 또한 사용자가 생성하는 새 케이스에도 모두 터널을 연결합니다 .
- **생성**—LabVIEW 는 사용자가 생성하는 모든 새 케이스에 자동으로 터널을 연결합니다 . 기존 케이스는 연결이 안된 채로 놔둡니다 .

앞에 있는 옵션 중에서 하나를 선택하는 경우 , LabVIEW 는 출력 터널과 입력 터널을 연결하거나 이 사이에 링크를 생성합니다 . 두 터널에 흰색 삼각 모양이 나타나서 LabVIEW 가 터널을 링크했다는 것을 표시합니다 . 링크된 출력 터널에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **링크된 입력 터널 >> 찾기** 또는 **링크된 입력 터널 >> 삭제**를 선택하여 , 링크된 입력 터널을 찾거나 터널 사이의 링크를 제거합니다 .



노트 다층 시퀀스 구조에서는 터널에 대해 자동 와이어링을 설정할 수 없습니다 .

기타 블록다이어그램의 개선사항

LabVIEW 8.6 의 블록다이어그램에서 다음과 같이 부수적으로 개선되었습니다 :

- LabVIEW 8.6 에서 모든 비교 함수는 LabVIEW 클래스와 같이 동작합니다 .
- LabVIEW 8.6 은 공유 라이브러리에 빌드한 VI 의 커넥터 팬에서 불리언 컨트롤과 인디케이터에 대한 원형 정보를 생성합니다 . LabVIEW 에서 공유 라이브러리를 호출하면 , LabVIEW 는 불리언을 부호없는 8 비트 정수로 처리합니다 . 컨트롤의 라벨이 **불리언**이면 , LabVIEW 는 에러를 반환합니다 . **불리언**은 대소문자를 구분하지 않습니다 .

프런트패널 개선사항

LabVIEW 8.6 의 프런트패널과 관련 기능에서 다음과 같이 개선되었습니다 .

3D 그래프 개선사항

(Windows) LabVIEW 8.5 및 이전 버전에 포함된 3D 표면 그래프 , 3D 파라미터형 그래프 , 3D 커브 그래프는 각각 ActiveX 3D 표면 그래프 , ActiveX 3D 파라미터형 그래프 , ActiveX 3D 커브 그래프로 이름이 변경되었습니다 . ActiveX 3D 그래프는 ActiveX 기술과 3D 형을 다루는 VI 를 사용합니다 .

LabVIEW 8.6은 모든 플랫폼에서 사용가능한 3D 그래프의 X 컨트롤 버전을 포함합니다. 3D 표면, 3D 파라미터형, 3D 라인 그래프를 사용하여 3D 플롯에 3D 데이터를 디스플레이합니다. **3D 그래프 프로퍼티** 대화 상자를 사용하여 3D 그래프 X 컨트롤의 모양을 사용자 정의합니다. 일반적인 그래프 모양을 설정하고, x, y, z 플롯을 포맷하고, 그래프 축을 포맷하고, 값 인덱스를 추가하고, 커서를 추가할 수 있습니다.

기타 프런트패널 개선사항

프라이빗 데이터 컨트롤은 X 컨트롤 또는 엄격한 타입의 X 컨트롤 참조 번호를 포함할 수 없습니다.

환경 개선사항

LabVIEW 8.6에서 LabVIEW 환경의 다음과 같은 사항이 개선되었습니다.

에러 리스트 윈도우 개선사항

LabVIEW 8.6에서 LabVIEW 클래스 에러 리포트 기능이 개선되어서 **에러 리스트** 윈도우에 깨지지 않은 클래스 아이템이 나타나는 경우가 줄었습니다.

성능 향상

VI 파일 크기가 감소했습니다.

VI 계층구조 윈도우 개선사항

LabVIEW 8.6에서 **VI 계층구조** 윈도우가 다음과 같이 개선되었습니다:

- **VI 계층구조** 윈도우는 MathScript 노드, 익스프레스 VI, 정적 VI 참조에서 참조하는 LabVIEW 클래스 및 다이내믹 멤버 VI, X 컨트롤, 공유 라이브러리, .m 파일을 디스플레이합니다.
- VI의 호출 체인을 하이라이트하려면, **VI 계층구조** 윈도우 상에서, 정지된 VI의 VI 아이콘에서 마우스 오른쪽 버튼을 클릭한 후 바로 가기 메뉴에서 **호출 체인 보이기**를 선택합니다. 호출 체인은 현재 VI에서부터 최상위 VI까지의 호출자의 체인입니다. 계층구조 연결을 하이라이트하면, 굵은 빨간색 선으로 연결이 표시됩니다.

VI 계층구조 윈도우에 대한 추가적인 정보는 *LabVIEW 도움말*의 **목차** 탭에서 **기본** >> **VI와 SubVI 생성하기** >> **사용법** >> **SubVI 생성하기** >> **VI 계층구조** 모음을 참조하십시오.

대화 상자 개선사항

LabVIEW 8.6의 대화 상자에서 다음과 같이 개선되었습니다.

경고 대화 상자 개선사항

경고 대화 상자가 **로드 경고 요약** 및 **이전 버전으로 저장 경고 요약** 대화 상자가 되었습니다. 요약 대화 상자에서 **세부사항 보이기** 버튼을 클릭하여 **로드 및 저장 경고 리스트** 대화 상자를 열고 각 경고 항목의 세부사항을 봅니다. 또한 **보기** 메뉴에서 **로드 및 저장 경고 리스트**를 선택하여 **로드 및 저장 경고 리스트** 대화 상자를 엽니다.

로드 경고 요약 대화 상자는 사용자가 로드하려는 최상위 레벨 아이템에 대한 경고 항목 리스트를 리포트합니다.

이전 버전으로 저장 경고 요약 대화 상자는 사용자가 저장하려는 최상위 레벨 아이템에 대한 경고 항목 리스트를 리포트합니다.

어플리케이션 빌더 대화 상자 변경사항

LabVIEW 8.6 에서 LabVIEW 어플리케이션 빌더가 다음과 같이 개선되었습니다 :

- **설치 프로그램 프로퍼티** 대화 상자의 **추가적인 설치 프로그램** 페이지에서 **설치 프로그램 소스 위치** 텍스트 박스는 소스 위치만을 디스플레이합니다 . 더이상 사용자가 이 대화 상자에서 설치 프로그램 소스 위치를 변경할 수 없습니다 . LabVIEW 가 자동으로 설치 프로그램의 소스를 찾을 수 없으면 , LabVIEW 는 이후 빌드 과정에서 직접 소스의 위치를 찾도록 입력 요청합니다 .
- 추가적인 설치 프로그램이나 구성요소를 포함하는 설치 프로그램을 만들 때 , 선택한 설치 프로그램 또는 설치 프로그램의 의존성을 LabVIEW 가 찾지 못하는 경우에는 **필요한 배포 파일** 대화 상자가 나타날 수 있습니다 .
- **(Windows)** 웹 서비스 빌드 스펙을 사용하면 LabVIEW 프로젝트에 웹 서비스로 VI 를 만들고 배포할 수 있습니다 . **프로젝트 탐색기** 윈도우의 **빌드 스펙**에서 마우스 오른쪽 버튼을 클릭한 후 바로 가기 메뉴에서 **새로 만들기** » **웹 서비스 (RESTful)** 를 선택하여 **웹 서비스 프로퍼티** 대화 상자를 디스플레이합니다 . LabVIEW 웹 서비스는 LabVIEW Full Development System 과 LabVIEW Professional Development System 에서만 사용 가능합니다 .

공유 변수 프로퍼티 대화 상자 개선사항

LabVIEW 8.6 은 **공유 변수 프로퍼티** 대화 상자에서 다음과 같이 개선되었습니다 :

- **공유 변수 프로퍼티** 대화 상자의 **변수** 페이지에서 **소스에 연결이 가명 활성화**로 변경되었습니다 . **가명 활성화** 확인란에 확인 표시를 하면 , 공유 변수의 접근 타입을 지정할 수 있습니다 .
- **변수** 페이지의 **버퍼링 사용**과 **단일 발신자** 옵션이 **공유 변수 프로퍼티** 대화 상자의 **네트워크**로 이동되었습니다 .
- **공유 변수 프로퍼티** 대화 상자의 **설명** 페이지가 DSC Module 이 설치되지 않은채로 나타납니다 .

옵션 대화 상자 개선사항

LabVIEW 8.6 은 **옵션** 대화 상자의 **웹 서버 : 설정**에서 다음과 같이 개선되었습니다 :

- LabVIEW 웹 서버에 대한 하나 이상의 인터페이스를 설정할 수 있습니다 . **서버 라벨** 옵션을 사용하여 서버 인터페이스의 이름을 사용자 정의할 수 있고 , **서버 라벨 (포트)** 리스트 박스는 웹 서버에서 현재 설정되어 있는 모든 인터페이스를 디스플레이합니다 .
- **리스너의 IP 주소**는 선택한 웹 서버 인터페이스의 IP 주소를 지정합니다 . 폴다운 메뉴는 로컬 컴퓨터에서 사용할 수 있는 모든 주소를 포함합니다 . 필드는 LabVIEW Real-Time Module 타겟을 사용할 때 자동으로 채워지지 않습니다 .
- **접근 허용** 옵션이 선택한 웹 서버 인터페이스에서 사용할 수 있는 기능을 설정합니다 . **접근 허용** 섹션에서 , 리모트 프런트패널 , 스크립트 , 웹 서비스 , 정적 내용 및 스냅샷을 활성화시킬 수 있습니다 .

LabVIEW 8.6 은 **옵션** 대화 상자에서 새 **웹 서비스 : 보안** 페이지를 포함합니다 . 이 페이지를 사용하면 LabVIEW 웹 서비스에 대한 보안을 설정할 수 있습니다 .

기타 대화 상자 개선사항

LabVIEW 8.6 에서 대화 상자가 다음과 같이 부수적으로 개선되었습니다 :

- **총돌 해결책** 대화 상자는 배포 과정에서 LabVIEW 가 감지하는 문제를 해결할 수 있는 옵션을 포함합니다 . 예를 들어 , 이 대화 상자는 꺼져 있는 타겟에 VI 를 배치할 경우에 나타납니다 . 사용자가 직접 문제를 해결할 수 있으면 , **총돌 해결책** 열에서 폴다운 메뉴가 나타납니다 . 리스트에서 옵션을 선택하여 관련 문제를 해결합니다 . 이 대화 상자에는 **이전** 및

다음 버튼이 있어서 **총돌** 리스트에 있는 문제가 있는 이전 또는 다음 아이템을 하이라이트 합니다 .

- **숫자형 프로퍼티** 대화 상자 및 **숫자형 상수 프로퍼티** 대화 상자의 **데이터 타입** 페이지와 **숫자형 노드 프로퍼티** 대화 상자의 **출력 설정** 페이지에 LabVIEW 가 고정 소수점 수에서 오버플로우 상태를 포함할지의 여부를 설정하는 **오버플로우 상태 포함** 확인란을 포함시켰습니다 .
- **잃어버린 아이템 찾기** 대화 상자는 LabVIEW 가 찾을 수 없는 디스크 상의 아이템을 참조하는 LabVIEW 프로젝트의 모든 아이템을 찾습니다 . **프로젝트 탐색기** 윈도우에 있는 프로젝트 루트에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **잃어버린 아이템 찾기** 를 선택하여 이 대화 상자를 디스플레이합니다 .

팔레트 개선사항

LabVIEW 8.6 에서 팔레트가 다음과 같이 개선되었습니다 :

프로그램적으로 컨트롤 및 함수 팔레트 세트 편집하기

팔레트 편집 VI 를 사용하여 팔레트를 프로그램적으로 생성하고 편집할 수 있습니다 . 여러 개의 팔레트를 편집하거나 , 새 팔레트를 생성하거나 , 편집한 팔레트의 모양을 나중에 점검하려는 경우에 팔레트 편집 VI 를 사용합니다 .

labview\examples\Palette API 디렉토리에서 프로그램적으로 **컨트롤**과 **함수** 팔레트 세트를 편집하는 예를 참조합니다 .

프로그램적으로 팔레트를 편집하는 방법에 대한 추가적인 정보는 *LabVIEW 도움말의 목차* 탭에 있는 **VI 와 함수 참조** » **프로그래밍 VI 와 함수** » **어플리케이션 컨트롤 VI 와 함수** » **팔레트 편집 VI** 모음을 참조하십시오 .

기타 팔레트 개선사항

LabVIEW 8.6 에서 팔레트가 다음과 같이 부수적으로 개선되었습니다 :

- **XML** 팔레트에 있었던 VI 와 함수를 이제 **LabVIEW 스키마** 팔레트에서 사용할 수 있습니다 . **LabVIEW 스키마** 팔레트는 **XML** 팔레트의 서브팔레트입니다 .
- 업그레이드하면 , LabVIEW 는 이전에 설치된 LabVIEW 버전에서 팔레트 포맷을 복구합니다 .

기타 환경 개선사항

LabVIEW 8.6 에서 환경이 다음과 같이 부수적으로 개선되었습니다 :

- 명령 라인 윈도우에서 다음의 명령을 입력하여 LabVIEW 로 만든 설치 프로그램이 설치한 파일과 LabVIEW 로 만든 설치 프로그램을 내부적으로 설치 제거할 수 있습니다 .

```
uninst.exe /qb /x 제품 이름 여기서 제품 이름은 설치 제거하려는 제품의 이름입니다 .
```

입력한 제품 이름이 제품의 이름과 일치하지 않으면 , 설치해제가 실패에 대한 통지 없이 실패합니다 .uninst.exe 파일은 C:\Program Files\National Instruments\Shared\NIUninstaller 디렉토리에 있습니다 .C:\Program Files\National Instruments\Shared\NIUninstaller 디렉토리가 작업 디렉토리가 아니면 , 디스크에서 uninst.exe 파일에 대한 전체 경로를 입력하십시오 .

- labview\vi.lib\Platform 디렉토리의 fileVersionInfo.llb 를 사용하여 독립 어플리케이션이나 공유 라이브러리의 버전 정보를 프로그램적으로 가져올 수 있습니다 . 이 VI 는 **함수** 팔레트에 없습니다 .
- **수행** 메뉴에는 더이상 **브레이크포인트** 옵션이 없습니다 . 대신에 **보기** » **브레이크포인트 관리자** 를 선택하여 , VI 계층구조에 있는 브레이크포인트를 관리합니다 .

- 폴더 내에 프로젝트 라이브러리가 있는 경우, 수행하는 소스 컨트롤 작업은 프로젝트 라이브러리 내의 아이템까지 확장됩니다.
- `_goopsup.llb` 가 `labview\vi.lib\Utility\` 디렉토리로 옮겨졌습니다.

새로운 또는 변경된 VI, 함수, 노드 개선사항

LabVIEW 8.6의 VI와 함수에서 다음과 같이 추가되고 변경되었습니다. VI, 함수, 노드에 대한 추가적인 정보는 *LabVIEW 도움말의 목차* 탭에서 **VI와 함수 참조** 모음을 참조하십시오.

새로운 VI와 함수

LabVIEW 8.6의 VI와 함수에서 다음과 같이 추가되었습니다.

고급 알림자 기다림 VI 및 함수

고급 알림자 기다림 팔레트는 다음과 같은 새 함수를 포함합니다:

- 알림자 히스토리가 있는 여러 알림 기다림
- 알림자 히스토리가 있는 알림 기다림

어플리케이션 컨트롤 VI와 함수

어플리케이션 컨트롤 팔레트는 다음과 같은 새 VI가 있는 새 **팔레트 편집** 팔레트를 포함합니다:

- 팔레트 읽기
- 팔레트 새로 고침
- 팔레트 쓰기

비교 함수

비교 팔레트는 다음과 같은 새 함수를 포함합니다:

- 고정 소수점 오버플로우?

연결 VI와 함수 (Windows)

연결 팔레트는 다음과 같은 새 VI가 있는 새 **웹 서비스** 팔레트를 포함합니다:



노트 다음의 VI는 LabVIEW Full 과 Professional Development Systems 에서만 제공됩니다.

- 세션 존재하는지 확인
- 세션 생성
- 세션 변수 삭제
- 세션 삭제
- 출력 비우기
- 세션 ID 쿠키 얻기
- 모든 Form Data 읽기
- 모든 요청 변수 읽기
- 모든 세션 변수 읽기
- Form Data 읽기
- Postdata 읽기
- 요청 변수 읽기
- 세션 변수 읽기
- ESP 템플릿 제공
- ESP 변수 설정

- HTTP 헤더 설정
- HTTP 다시 지정 설정
- HTTP 응답 코드 설정
- HTTP 응답 MIME 타입 설정
- 응답 쓰기
- 세션 변수 쓰기

웹 서비스 팔레트는 다음과 같은 새 VI 가 있는 새 **유틸리티** 팔레트를 포함합니다 :

- HTTP URL 인코딩
- HTTP URL 디코딩

고정 소수점 함수

고정 소수점 팔레트는 다음과 같은 새 함수를 포함합니다 :

- 고정 소수점 오버플로우 상태 지우기
- 고정 소수점 오버플로우 상태 포함
- 고정 소수점 오버플로우 상태 제거

고급 그림 기능 VI

고급 그림 기능 팔레트는 LabVIEW Full 과 Professional Development Systems 에서 센서 맵핑 익스프레스 VI 를 포함합니다 . 센서 맵핑 익스프레스 VI 를 사용하여 3D 모델로 현실 세계의 데이터를 맵핑합니다 .

큐 동작 함수

큐 동작 팔레트는 다음과 같은 새 함수를 포함합니다 :

- 손실있는 원소 큐에 추가

세마포어 VI

세마포어 팔레트는 다음과 같은 새 VI 를 포함합니다 :

- 세마포어 참조 얻기
- 세마포어 참조 해제

XML VI 와 함수

XML 팔레트는 다음의 VI 와 함수가 있는 새 **XML 파서** 팔레트를 포함합니다 .

- 닫기
- 첫번째 텍스트 아닌 하위 아이템 얻기
- 다음의 텍스트 아닌 등위 아이템 얻기
- 노드 텍스트 내용 얻기
- 인보크 노드 (XML)
- 로드
- 새로 만들기
- 프로퍼티 노드 (XML)
- 저장

VISA 고급 VI 와 함수

VISA 고급 팔레트는 다음과 같은 새 VI 를 포함합니다 :

- VISA 프로퍼티 노드

변경된 VI, 함수, 노드

LabVIEW 8.6 에서 다음과 같은 VI, 함수, 노드가 변경되었습니다 .

고급 파일 VI 및 함수

고급 파일 팔레트는 다음과 같이 변경된 함수를 포함합니다 .

볼륨 정보 얻기 함수는 **섹터 크기 (바이트)** 출력을 포함하여 디스크의 가장 작은 물리적 스토리지 단위의 크기를 표시합니다 . 버퍼링 없이 파일을 여는 경우 , 파일 데이터의 크기를 주어진 섹터 크기의 배수로 만들어야 합니다 . (파일 열기 / 생성 / 대체) 함수의 **버퍼링 비활성화** 입력을 참으로 설정하여 , 버퍼링을 비활성화할 수 있습니다 .

비교 함수

비교 팔레트는 다음과 같은 변경된 함수를 포함합니다 :

- **범위내 확인과 강제변환**— 고정 소수점 데이터 타입을 지원합니다 .
- **최대 & 최소**— 고정 소수점 데이터 타입을 지원합니다 .

파일 I/O VI 와 함수

파일 I/O 팔레트는 다음과 같이 변경된 함수를 포함합니다 .

파일 열기 / 생성 / 대체는 **버퍼링 비활성화** 입력이 포함되어 , 버퍼링 없이 LabVIEW 가 파일을 열고 , 생성하고 , 대체할지의 여부를 지정합니다 .

특정 상황에서 데이터 전송 속도를 높이려면 버퍼링을 비활성화합니다 . 예를 들어 , RAID (Redundant Array of Independent Disks) 에 대용량의 데이터를 스트리밍하거나 데이터를 저장하려는 경우 , 버퍼링을 비활성화하여 LabVIEW 가 직접 디스크에 데이터를 쓰게합니다 .

LabVIEW 스키마

LabVIEW 스키마 팔레트는 다음과 같이 변경된 함수를 포함합니다 :

- **패턴화된 XML 로**— LabVIEW 클래스와 고정 소수점 데이터 타입을 지원합니다 .
- **XML 로부터 패턴화 해제**— LabVIEW 클래스와 고정 소수점 데이터 타입을 지원합니다 .

숫자형 함수

숫자형 팔레트는 다음과 같은 변경된 함수를 포함합니다 :

- **나누기**— 고정 소수점 데이터 타입을 지원합니다 .
- **역수**— 고정 소수점 데이터 타입을 지원합니다 .
- **제곱근**— 고정 소수점 데이터 타입을 지원합니다 .

리포트 생성 VI

리포트 생성 팔레트는 LabVIEW Report Generation Toolkit 특유의 입력과 출력을 가진 VI 를 포함합니다 . Report Generation Toolkit 을 설치하지 않으면 VI 는 관련 입력과 출력을 무시합니다 . **리포트 생성** 팔레트는 다음과 같이 추가적으로 변경된 VI 도 포함합니다 :

- **수평 라인을 리포트에 추가**— 폭 입력의 기본값이 픽셀에서 퍼센트로 변경되었습니다 .
- **리포트 텍스트 추가**— 포맷 문자열 입력을 포함하여 LabVIEW 가 숫자를 문자로 변환할 때 사용하는 숫자 포맷을 지정합니다 .
- **VI 블록 다이어그램을 리포트에 추가**— 모든 인스턴스는 **정렬** 입력을 포함하여 리포트에서의 관련 이미지 정렬을 설정합니다 .
- **VI 계층구조를 리포트에 추가**— 모든 인스턴스는 **정렬** 입력을 포함하여 리포트에서의 관련 이미지 정렬을 설정합니다 .
- **VI 아이콘을 리포트에 추가**— 모든 인스턴스는 **정렬** 입력을 포함하여 리포트에서의 관련 이미지 정렬을 설정합니다 .

- **VI의 컨트롤 리스트를 리포트에 추가**—모든 인스턴스는 **정렬** 입력을 포함하여 리포트에서의 관련 이미지 정렬을 설정합니다.
- **VI의 SubVI 리스트를 리포트에 추가**—모든 인스턴스는 **정렬** 입력을 포함하여 리포트에서의 관련 이미지 정렬을 설정합니다.
- **새 리포트 라인**—**라인 개수** 입력을 포함하여 VI가 리포트에 추가하는 라인의 수를 지정합니다.
- **리포트를 파일에 저장**—**대체 입력요청?** 입력을 포함하여 사용자가 기존 파일 이름을 지정할 경우 VI가 대화 상자를 디스플레이할지의 여부를 결정합니다.

스펙트럼 분석 VI (Base Package 에 포함 안됨)

스펙트럼 분석 팔레트는 다음과 같이 변경된 VI를 포함합니다.

STFT 스펙트로그램 VI는 **시간 - 주파수 설정** 입력을 포함하여 **주파수 계급**의 설정을 지정하고 **STFT 스펙트로그램 {X}**의 열 개수를 결정합니다. 이 VI는 또한 **에너지 보존?** 입력을 포함하여, 이를 통해 조인트 시간 - 주파수 영역의 에너지가 시간 영역의 에너지와 같도록 **STFT 스펙트로그램 {X}**을 스케일할지의 여부를 지정합니다.

라이브러리 함수 호출 노드

라이브러리 함수 호출 대화 상자에서 다음과 같은 사항이 변경되었습니다:

- **파라미터 탭의 데이터 타입** 폴더는 **부호있는 포인터 크기 정수** 및 **부호없는 포인터 크기 정수** 숫자형 데이터 타입을 포함합니다. 이 두가지 데이터 타입을 사용하는 경우, 라이브러리 함수 호출 노드는 실행되고 있는 특정 OS에 맞춰지고 적절한 크기의 데이터를 라이브러리 함수에게 반환합니다. 64 비트 플랫폼 LabVIEW는 이러한 숫자형 데이터 타입을 64 비트 정수 타입으로 변경합니다. 32 비트 플랫폼 LabVIEW는 이러한 숫자형 데이터 타입을 32 비트 정수 타입으로 변경합니다.
- **라이브러리 함수 호출 노드 호출**은 * 와일드카드의 사용을 지원하여 32 비트 및 64 비트 윈도우 사이에서 **라이브러리 이름 또는 경로** 참조 플랫폼을 독립적으로 만듭니다. 확장자 왼쪽에 하나의 *가 있는 경우, 32 비트 윈도우 플랫폼에서는 32로 해석되고 64 비트 윈도우 플랫폼에서는 64로 해석됩니다. 예를 들어, myLibrary*.dll는 myLibrary32.dll 나 myLibrary64.dll로 인식될 것입니다. 두 개의 *가 있는 경우는 이전에는 라이브러리 이름이 플랫폼을 참조하지 않았으나, 새 라이브러리는 참조하는 상황에서 쓰입니다. 예를 들어, myLibrary**.dll는 32 비트 플랫폼에서는 myLibrary.dll로 해석되고 64 비트 플랫폼에서는 myLibrary_64.dll로 해석됩니다.

공유 변수 노드

공유 변수 노드 노드가 다음과 같이 변경되었습니다:

- 데이터를 읽는 것으로 설정된 네트워크 공유 변수 노드에 대한 **ms 타임아웃 컨트롤과 타임아웃?** 인디케이터를 활성화하는 옵션을 포함합니다.
- 블록 다이어그램의 많은 공유 변수와 함께 작업시 공유 변수가 LabVIEW 8.6에서 성능 향상을 보여줍니다.
- LabVIEW Real-Time Module를 설치할 때 지원되는 타겟에서 사용할 수 있는 타임 트리거 변수가 표준 공유 변수와 별개로 나타납니다. 지원되는 타겟에서 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **새로 만들기** > **타임 트리거 변수**를 선택하여 타임 트리거 변수를 생성합니다.

기타 VI, 함수, 노드 변경 내역

LabVIEW 8.6의 기타 사항이 다음과 같이 부수적으로 변경되었습니다.

(복소수 정렬) VI의 **메소드** 입력의 값 이름이 변경되었습니다. 크기의 이름이 **크기, 실수, 허수**로 변경되었습니다. **방법** 입력이 새 값인 **크기, 위상 각**을 포함합니다. **크기, 실수, 허수**는 우선 크기에 따라 원소를 정렬합니다. 크기가 같은 원소의 경우, VI는 실수 부분, 그리고 허

수 부분에 따라 원소를 정렬합니다. 크기, 위상 각은 우선 크기에 따라 원소를 정렬합니다. 크기가 같은 원소의 경우 π 는 위상 각에 따라 원소를 정렬합니다. 위상 각도는 π 에서 π 까지의 범위 내에 있습니다.

새로운 클래스, 프로퍼티, 메소드, 이벤트

LabVIEW 8.6은 새로운 VI 서버 클래스, 프로퍼티, 메소드, 이벤트를 포함합니다. 새 클래스, 프로퍼티, 메소드, 이벤트의 리스트는 *LabVIEW 도움말의 목차* 탭에서 **LabVIEW 8.6의 특징과 변경 내역** > 새 VI 서버 객체 모음을 참조하십시오.

XML 파서 프로퍼티와 메소드

XML 파서 VI 및 함수와 함께 사용할 수 있는 새 프로퍼티와 메소드의 리스트에 대한 정보는 *LabVIEW 도움말의 목차* 탭에 있는 **프로퍼티와 메소드 참조** > XML 파서 모음을 참조합니다.

3DPC_SurfacePlot 프로퍼티

3D 그래프와 같이 사용할 수 있는 새 프로퍼티 목록은 *LabVIEW 도움말의 목차* 탭에서 **프로퍼티와 메소드 참조** > 3DPC_SurfacePlot 모음을 참조하십시오.

블록다이어그램 자동 정리

편집 > **다이어그램 정리**를 선택하여, 자동으로 모든 기존 와이어를 다시 연결하고 블록다이어그램의 객체를 재배치하여 더 깔끔한 레이아웃을 만듭니다. 정리 옵션을 설정하려면, **도구** > **옵션**을 선택하여 **옵션 대화 상자**를 디스플레이하고 **항목** 리스트에서 **블록다이어그램 : 정리**를 선택합니다. 자동으로 컨트롤을 다이어그램 왼쪽으로 옮기고, 인디케이터를 다이어그램 오른쪽으로 옮기고, 블록다이어그램 객체와 와이어 사이에 지정된 픽셀 수를 놓고, 블록다이어그램 레이아웃을 간결하게 만들도록 LabVIEW를 설정할 수 있습니다.

빠른 탐색을 사용하여 객체 놓기

빠른 탐색 대화 상자를 사용하여 블록다이어그램 또는 프런트패널 객체를 이름으로 찾고, **컨트롤**이나 **함수** 팔레트를 탐색하지 않은 채로 객체를 블록다이어그램 또는 프런트패널에 놓을 수 있습니다. (Windows 및 Linux) <Ctrl-Space> 키를 누르거나 **보기** > **빠른 탐색**을 선택하여 **빠른 탐색** 대화 상자를 디스플레이합니다. 중국어 키보드에서는, <Ctrl-Shift-Space> 키를 누릅니다. (Mac OS) <Command-Shift-Space> 키를 눌러 이 대화 상자를 디스플레이합니다.

빠른 탐색 대화 상자를 사용하는 것에 대한 보다 자세한 정보에 대해서는 *LabVIEW 도움말의 Contents* 탭에 있는 **기본** > **LabVIEW 환경** > **사용법** > **객체 찾기** 모음을 참조하십시오.

여러 객체의 프로퍼티 편집

프런트패널 또는 블록다이어그램에서 여러 객체를 선택하고, 해당 객체들이 공유하는 프로퍼티를 모두 편집할 수 있습니다. 여러 객체를 선택하려면, 위치 도구를 사용하여 선택 사각형으로 편집하려는 모든 객체를 둘러싸거나 <Shift> 키를 누른 채로 각각의 객체를 클릭합니다. 선택한 객체 중 하나에서 마우스 오른쪽 버튼을 클릭한 후 바로 가기 메뉴에서 **프로퍼티**를 선택하여 **프로퍼티** 대화 상자를 디스플레이합니다. **프로퍼티** 대화 상자는 선택된 객체가 서로 공유하는 탭과 프로퍼티만을 디스플레이합니다. 비슷한 객체를 선택하여 더 많은 탭과 프로퍼티를 디스플레이합니다. 같은 프로퍼티를 공유하지 않는 여러 객체를 선택하는 경우, **프로퍼티** 대화 상자는 탭이나 프로퍼티를 디스플레이하지 않습니다.

LabVIEW 웹 서비스 (Windows, Base Package 에 포함 안됨)

LabVIEW 8.6을 사용하여 VI를 웹 서비스로 공개할 수 있습니다. 웹 서비스는 서버가 HTTP 클라이언트가 접근가능한 어플리케이션을 배포하는 표준화된 방법입니다. LabVIEW 웹 서비스는 대부분의 주요 플랫폼과 프로그래밍 언어에서 클라이언트를 지원하며, 이를 통해 LabVIEW를 사용하여 네트워크 상에 웹 어플리케이션을 쉽게 구현하고 배포할 수 있습니다.

옵션 대화 상자의 웹 서버 : 설정 페이지를 통해 웹 서비스 기능을 활성화할 수 있습니다. 웹 서비스를 만들어 배포하려면, **프로젝트 탐색기** 윈도우의 **빌드 스펙**에서 마우스 오른쪽 버튼을 클릭한 후 바로 가기 메뉴에서 **새로 만들기** > **웹 서비스 (RESTful)** 를 선택하여 **웹 서비스 프로퍼티** 대화 상자를 디스플레이합니다. LabVIEW 웹 서비스는 LabVIEW Full Development System 과 LabVIEW Professional Development System 에서만 사용 가능합니다.

LabVIEW 에서 웹 서비스를 사용하는 것에 대한 추가적인 정보는 *LabVIEW 도움말의 목차* 탭에서 **기본** > **LabVIEW 웹 서비스** 모음을 참조하십시오.

NI 배포된 시스템 관리자

(Windows) LabVIEW 8.6 은 변수 관리자를 공유 변수를 관리하는 통합 도구로 대체하는 NI 배포된 시스템 관리자를 포함합니다. 시스템 관리자를 사용하여 변수, 프로세스, I/O 서버, 웹 서비스를 생성하고 모니터링합니다. 또한 시스템 관리자를 사용하여 공유 변수 엔진과 상호작용하고 보안과 가명을 관리합니다.

배포된 시스템 관리자에 대한 추가적인 정보는 *LabVIEW 도움말의 목차* 탭에 있는 **기본** > **LabVIEW 를 이용한 네트워킹** > **개념** > **공유 변수를 사용하여 실시간 데이터 공유하기** 항목을 참조하십시오.

LabVIEW MathScript 개선사항 (Base Package 에 포함 안됨)

LabVIEW 8.6 에서 MathScript 가 다음과 같이 개선 및 변경되었습니다.



노트 도구 > **MathScript 윈도우**를 선택하여 **LabVIEW MathScript 윈도우**를 디스플레이합니다.

새 MathScript 함수

LabVIEW 8.6 에서 다음의 MathScript 함수가 추가되었습니다. 이러한 함수를 **LabVIEW MathScript 윈도우**나 MathScript 노트에서 사용할 수 있습니다.

MathScript 클래스	함수
고급	sphbesselh, sphbesselj, sphbessely
근접	csaps, lsqcurvefit
오디오	auread
명령	keyboard
필터 디자인	gaussfir, yulewalk
필터 구현	cconv
적분	quad
ode	odeset, radau5
플롯	imagesc, xlim, ylim, zlim
스펙트럼 분석	buffer, cpsd, mscohere, pburg, pcov, peig, periodogram, pmcov, pmusic, pwelch, pyulear, rooteig, rootmusic, specgram, spectrogram, tfestimate
pde	pdeelliptic
지원	warning, xlsread, xlswrite

MathScript 클래스	함수
웨이브폼 생성	stepfun
윈도우	taylorwin

Mac OS 및 Linux 에서의 LabVIEW MathScript

LabVIEW MathScript 윈도우 및 MathScript 노드를 Mac OS 와 Linux 에서 사용할 수 있습니다. 하지만, 일부 함수는 관련 OS 에서 사용할 수 없습니다. 다음 테이블은 Mac OS 또는 Linux 에서 사용할 수 없는 MathScript 함수를 보여줍니다. **x** 는 함수를 해당 OS 에서 사용할 수 없다는 것을 표시합니다.



노트 getfileproperty 함수, load 함수, save 함수, setfileproperty 함수는 Mac OS 와 Linux 에서 사용 가능합니다. 하지만, 이 함수를 2 진 측정 파일 (.tdm 또는 .tdms) 과 함께 사용할 수 없습니다.

MathScript 함수	Mac OS	Linux
aich	x	
aiwf	x	
aoch	x	
aowf	x	
calllib	x	x
dioread	x	
diowrite	x	
dos	x	x
ginput	x	x
gtext	x	x
libfunctionsview	x	x
libisloaded	x	x
loadlibrary	x	x
pause	x	x
system	x	x
unloadlibrary	x	x
waitforbuttonpress	x	x

MathScript 노드의 디버깅 개선사항

LabVIEW 8.6 에서 MathScript 노드 디버깅이 다음과 같이 개선되었습니다 :

- MathScript 노드의 스크립트내에서 실행 하이라이트 , 단계별 실행 , 브레이크포인트를 사용할 수 있습니다 .
- MathScript 노드 왼쪽의 회색 영역에 디스플레이되는 것 :
 - 에러를 포함하는 스크립트 라인 옆의 빨간 에러 문양
 - 경고 문양
 - 브레이크포인트
- **에러 리스트** 윈도우에서 MathScript 노드 에러를 선택하고 **에러 보기**기 버튼을 클릭하면 , LabVIEW 는 에러를 포함하는 스크립트 라인을 하이라이트합니다 .

MathScript 노드에서의 스크립트 하이라이트하기

스크립트 하이라이트는 색을 사용하여 MathScript 노드 스크립트의 서로 다른 부분을 구분합니다 . 색을 사용하여 스크립트를 더 쉽게 알아볼 수 있고 , 에러를 포함하고 있거나 기대하지 않은 데이터를 반환하는 스크립트를 디버깅하는데 도움이 됩니다 . 예를 들어 , 스크립트 하이라이트를 사용하면 사용자 정의 함수나 변수가 내장된 MathScript 함수를 덮어쓸 때 이를 볼 수 있습니다 . 기본적으로 스크립트 하이라이트는 활성화되어 있는데 , 이전 버전의 LabVIEW 에서 마지막으로 저장되었고 사용자 색을 사용하는 MathScript 노드의 경우는 예외입니다 .

특정 MathScript 노드에 대해 스크립트 하이라이트를 활성화 또는 비활성화하려면 , MathScript 노드에서 마우스 오른쪽 버튼을 클릭한 후 바로 가기 메뉴에서 **스크립트 하이라이트 활성화** 또는 **스크립트 하이라이트 비활성화**를 선택합니다 . 모든 MathScript 노드에 대해 스크립트 하이라이트를 활성화 또는 비활성화하고 스크립트 하이라이트가 활성화된 MathScript 노드의 스크립트 하이라이트 색을 사용자 정의하려면 , **옵션** 대화 상자의 **MathScript: 스크립트 하이라이트** 페이지를 사용합니다 .

부수적인 MathScript 개선사항과 변경 내역

LabVIEW 8.6 의 MathScript 에서 다음과 같은 부수적인 사항이 변경되었습니다 :

- **VI 계층구조** 윈도우는 MathScript 에서 참조하는 .m 파일을 디스플레이합니다 .
- `delete` 함수는 `obj` 입력을 포함하는데 , 이를 통해 플롯 객체를 삭제하고 **LabVIEW MathScript 윈도우의 변수 리스트**에서 객체를 제거할 수 있습니다 .
- `filter` 함수는 `'direct'` 입력을 포함하는데 , 이를 통해 LabVIEW 가 푸리에 변환 대신 직접 합성곱을 사용하도록 만들 수 있습니다 .
- LabVIEW MathScripts 의 성능이 개선되어 **LabVIEW MathScript 윈도우**와 MathScript 노드 모두에서 컴파일 작업이 더욱 빠르게 수행됩니다 .
- VI 에 MathScript 노드가 포함되어 있으면 , LabVIEW 는 이 VI 를 저장하거나 실행할 때 모든 실행 취소 정보를 제거합니다 .

LabVIEW MathScript 에 대한 추가적인 정보는 *LabVIEW 도움말의 목차* 탭에서 **기본**»**수식과 방정식** 모음을 참조하십시오 .

고정 소수점 수에 대해 오버플로우 관리하기

오버플로우 조건은 고정 소수점 수에서 산술 연산을 수행하거나 (고정 소수점으로) 함수를 사용하여 숫자형 데이터를 고정 소수점 데이터로 변환할 때 발생할 수 있습니다 . 오버플로우의 발생 여부를 구분하기 위해 , 오버플로우 상태를 포함하도록 고정 소수점 수를 설정할 수 있습니다 . 고정 소수점 수에 오버플로우 상태를 포함하면 , LabVIEW 는 추가 저장 공간을 할당하여 고정 소수점 수가 오버플로우된 작업의 결과인지 확인할 수 있습니다 .

오버플로우 상태를 포함하도록 고정 소수점 수를 설정한 이후, 고정 소수점 컨트롤, 상수, 인디케이터에 오버플로우 상태 LED 를 디스플레이할 수 있습니다. 고정 소수점 수의 오버플로우 상태가 참일 때 이 LED 가 켜집니다. 또한 고정 소수점 오버플로우 함수를 사용하여 고정 소수점 수의 오버플로우 상태를 구분할 수 있습니다. 고정 소수점 함수를 사용하여 고정 소수점 수의 오버플로우 상태를 조절할 수 있습니다.

고정 소수점 수에 대한 추가적인 정보는 *LabVIEW 도움말의 목차*에서 **기본»블록다이어그램 만들기»개념»숫자형 데이터** 항목을 참조하십시오.

LLB 병합하기 (Base Package 에 포함 안됨)

LabVIEW Professional Development System 에서, **병합할 LLB 선택** 대화 상자를 사용하여 원본 LLB 와 두 개정본 사이의 차이를 병합하고 해결합니다.

도구»병합» LLB 병합을 선택하여 **병합할 LLB 선택** 대화 상자를 디스플레이합니다. **기본 LLB** 필드에서 원본 LLB 를 지정합니다. **상대편 LLB** 및 **내 LLB** 필드에 병합할 두 LLB 를 지정합니다. **병합** 버튼을 클릭하여 선택한 LLB 를 병합하고 **LLB 병합** 대화 상자를 엽니다. LLB 개정본 사이의 차이를 해결하고 **닫기**와 **저장**을 클릭하여 병합된 LLB 를 저장합니다.

LLB 병합에 대한 추가적인 정보는 *LabVIEW 도움말의 목차* 탭에서 **기본» Development Guidelines »사용법» VI 및 LLB 병합하기» LLB 병합하기** 항목을 참조하십시오.

National Instruments, NI, ni.com 과 LabVIEW 는 National Instruments Corporation 의 상표들입니다. National Instruments 의 상표들에 관한 더 많은 정보를 원하신다면 ni.com/legal 에서 *Terms of Use* 란을 참조하십시오. 이 문서에서 언급된 다른 제품과 회사의 이름들은 각각 해당 회사들의 상표이거나 상호들입니다. MATLAB® 은 The MathWorks, Inc. 의 등록된 상표입니다. 적절한 위치에서 내쇼날인스트루먼트의 특허권을 참조할 수 있습니다 : 소프트웨어의 **Help>Patents**, CD 의 `patents.txt` 파일, 또는 ni.com/patents.