SIEMENS

SIMATIC HMI

WinCC V7.0 SmartTools

04/2008

Содержание

Обзор

1

Содержание

Smart Tools представляет собой набор программ, которые полезны при работе с WinCC.

В него входят следующие программы и файлы:

- Tag Simulator
- Tag Export/Import
- Dynamic Wizard Editor
- Documentation Viewer
- WinCC CrossReferenceAssistant
- OnlOff и OnlOn
- Communication Configurator
- WinCC Configuration Tool
- WinCC Archive ConfigurationTool

Примечание

SmartTools — это дополнительные инструменты. Помните, что они могут повлиять на работу системы WinCC, например, на работу среды исполнения и требования к памяти.

Критерии удобства для пользователя и функциональности, свойственные базовому программному обеспечению WinCC, необязательно применяются к этому набору инструментов.

Дополнительные источники информации

WinCC Communication Configurator (стр. 115)

Экспорт/импорт тегов (стр. 9)

Инструмент Tag simulator (Симулятор тегов) (стр. 19)

Редактор мастера динамики (стр. 25)

Средство просмотра документации WinCC (стр. 97)

WinCC CrossReferenceAssistant (стр. 103)

Файлы регистрации OnlOff.reg, OnlOn.reg (стр. 113)

Экспорт/импорт тегов 2

2.1 Экспорт/импорт тегов

Краткое описание

Программа экспортирует все соединения, все структуры данных и все теги из открытого проекта в соответствующие файлы ASCII. Эти данные можно затем импортировать во второй проект. Формат ASCII обеспечивает обработку данных с помощью программы для работы с электронными таблицами перед их повторным импортом.

2.2 Установка инструмента Export/Import Tags (Экспорт/импорт тегов)

Введение

Инструмент Tag Export/Import (Экспорт/импорт тегов) можно установить двумя способами.

Все функции этой программы можно использовать, только если установлена система WinCC.

Требования

В многопользовательских проектах WinCC интеллектуальный инструмент Tag Export/Import (Экспорт/импорт тегов) невозможно использовать для клиента без собственного проекта.

Процедура

1. Во время установки WinCC в диалоговом окне Programs (Программы) выберите пункт WinCC V7.0 complete (Полная установка WinCC V7.0).

   WinCC установится вместе с пакетами SmartTools, WinCC ConfigurationTool и WinCC Archive ConfigurationTool.

Для запуска приложения Tag Export/Import (Экспорт/импорт тегов) выберите SIMATIC > WinCC > Tools (Инструменты).

Альтернативная процедура

Приложение Tag Export/Import (Экспорт/импорт тегов) можно также установить с DVD-диска WinCC.

1. Перейдите в каталог на DVD-диске WinCC «InstData\WinCC\setup\Products\SC_SMARTTOOLS».
2. Дважды щелкните значок setup.exe.
3. В диалоговом окне Components (Компоненты) выберите VarExpImp.
4. Нажмите кнопку Continue (Продолжить). Следуйте инструкциям в диалоговых окнах.

2.3 Работа

ЭКСПОРТ

1. Сначала запустите систему WinCC и откройте проект, из которого необходимо экспортировать теги. Запустите файл VAR_EXIM.EXE.
2. Укажите путь к файлу и имя файла, в которой необходимо выполнить экспорт. Сначала потребуется указать только имя файла без расширения.
3. Выберите режим Export (Экспорт).
4. Нажмите кнопку Execute (Выполнить). Подтвердите сведения в окне сообщения.
5. Подождите, пока в строке состояния не отобразится сообщение End Export/Import (Экспорт/импорт завершен).
6. Созданные файлы можно просмотреть с помощью соответствующих кнопок tag (теги), con (соединения), dex (структуры) и diag (протокол).

Пустые группы не экспортируются.
Символ подчеркивания (_) зарезервирован для присвоения имен. По этой причине в именах файлов должен отсутствовать символ подчеркивания.

ИМПОРТ

1. Сначала запустите систему WinCC и откройте проект, в который необходимо импортировать теги.
2. Все драйверы каналов, в которые необходимо импортировать соединения, должны быть доступны в проекте. При необходимости добавьте отсутствующие драйверы в проекты.
3. Запустите файл VAR_EXIM.EXE.
4. Укажите путь к файлу и имя файла, из которого необходимо выполнить импорт. Сначала потребуется указать только имя файла без расширения. В диалоговом окне выбора щелкните один из трех экспортированных файлов.
5. Выберите режим Import (Импорт) или ImportOverwrite. В режиме ImportOverwrite теги, которые уже имеются в целевом проекте, перезаписываются импортированными тегами с таким же именем. В режиме Import (Импорт) сообщение вместо этого записывается в файл протокола и тег в целевом проекте остается без изменений.
6. Нажмите кнопку Execute (Выполнить). Подтвердите сведения в окне сообщения.

7. Подождите, пока в строке состояния не отобразится сообщение End Export/Import (Экспорт/импорт завершен).
8. Просмотрите созданные данные в системе управления тегами WinCC.

В обоих режимах импорт невозможен при запущенной среде исполнения WinCC.

В следующих разделах содержатся технические сведения о Tag Export/Import (Экспорт/импорт тегов). Тем не менее, эти сведения не требуются для стандартной ситуации, когда целевой компьютер имеет ту же конфигурацию системы, что и система WinCC, использовавшаяся при экспорте. Знания структуры переменных WinCC необходимы, только если необходимо создать новые или изменить существующие теги с помощью файлов ASCII.

2.4 Инструмент Tag Export/Import (Экспорт/импорт тегов)

Инструмент Tag Export/Import (Экспорт/импорт тегов) является независимым приложением на основе WinCC API. Инструмент может использоваться для экспорта всех тегов WinCC проекта в файлы ASCII, а также для импорта тегов, например во второй проект. При этом создаются три файла.

- [*имя*]_*cex.csv* для логических соединений
- [*имя*]_*dex.csv* для структурных описаний
- [*имя*]_*dex.csv* для структурных описаний

В файлах создается заголовок, содержащий сведения о дате создания. Во время импорта три файла считываются автоматически.

Файл [*имя*]_*cex.csv* импортируется первым, поскольку теги могут быть созданы только при наличии соответствующего соединения.
После этого импортируются структуры данных, определенные в файле [*имя*]_*dex.csv*. Эти типы пользовательских данных, которые должны быть известны до создания тега такого типа.
Потом считываются определения тегов из файла [*имя*]_*vex.csv*.

Группы тегов невозможно создать независимо от тега. Если группа не существует, она создается автоматически вместе с тегом. Поэтому при экспорте не создается файл группы. Если группы определены в проекте, который не содержит тегов, эти группы не экспортируются.

При создании тега, помимо прочего, настраивается адресация, определяющая физическое расположение тега в системе автоматизации. Этот адрес зависит от канала связи и подключенной системы автоматизации. При настройке с помощью проводника WinCC для пользователя отображается экран ввода данных, которые зависят от канала. При редактировании файла импорта инструмента Tag Export/Import (Экспорт/импорт тегов) эти характеристики необходимо учитывать.
В любом случае экспортированные теги можно импортировать во второй проект WinCC, только если эта система имеет ту же конфигурацию, что и системы, на которой осуществлялся экспорт данных. Во время импорта данных в проект теги, которые уже могут существовать, должны быть известны. При необходимости адреса потребуется исправить вручную.
С другой стороны в некоторых случаях возможен импорт тегов из систем с другой

конфигурацией канала. Возможность выполнения этой процедуры зависит от канала и AS!

Тем не менее, экспортированные структурные типы не зависят от оборудования. Все параметры канала или соединения, например Connection (Соединение), GroupName, информация об адресе, не учитываются. Эти параметры определяются только при создании структурного тега.

Примечания относительно экспорта строки адреса в szSpecific

Строка адреса проверяется программным обеспечением канала при создании тега. Эти канальные DLL предполагают наличие определенного синтаксиса, который не должен подвергаться изменениям для определенной страны. По этой причине во время экспорта строки адреса располагаются в дополнительных кавычках «», которые затем удаляются во время импорта. Это необходимо для того, чтобы средства, подобные Excel, не изменяли информацию об адресе (проблема с разделителем списков).

2.5 Соединения

Соединение можно импортировать, только если настроен соответствующий драйвер канала. Кроме того, параметры в файле [*имя*]_*cex.csv* должны соответствовать настроенному драйверу канала.

Если на целевом компьютере должен использоваться канал, отличный от системы, из которой экспортированы данные, то этот канал необходимо изменить в первую очередь в экспортированных данных.

Простая процедура для определения необходимых данных соединения:
настройте все соединения на целевом компьютере и начните процесс экспорта. Это позволяет получить параметры целевого компьютера из файла [*имя*]_*cex.csv*.

Пример логического соединения

#Conname	Блок	Общие сведения	Особенность	Флаг
VerbAS1	Industrial Ethernet	Общие сведения	""	0
VerbAS2	Profibus FMS	Общие сведения	""	0

Во второй строке отображается логическое соединение драйвера Industrial Ethernet из Simatic S7 Protocol Suite.
В третьей строке содержится логическое соединение драйвера Profibus FMS.

В столбце #Conname указано имя логического соединения. В экспортированном файле это имя логического соединения находится в данных тега. Логическое соединение используется для связи с системой автоматизации, чтобы получить доступ к внешним тегам в процессе.

2.6 Файловые структуры

Структура файла [*имя*]_*dex.csv*:

#Structure Name	Идентификатор типа	Идентификатор создателя	Путь к проекту	
Control_1	1001	2500	C:\Testdaten\Proj.mcp	
#Varname	C.Vartype	C.CreatorID	C.VarLength	и т. д.
NewTag1	...	...	...	...
NewTag2	...	...	...	...
NewTag3	...	...	...	...
и т. д.	...	...	...	...

#:	Комментарий
#Structure Name:	Следующая строка содержит имя файловой структуры с параметрами структуры. Указание пути к проекту служит только для документирования того, из какого проекта были экспортированы данные. Данные автоматически импортируются в текущий открытый проект.
#Varname:	В следующих строках содержатся элементы файловой структуры, пока не будет обнаружена новая строка #Structure Name или не будут определены дополнительные строки. В одной строке содержатся все параметры, необходимые для определения тега.

Примечание для пользователей WinCC API

Заголовок столбца первой строки содержит имена параметров соответствующих файловых структур вызовов API. В этом случае данные могут четко интерпретироваться в таблице Excel.
Указание буквы с точкой в имени облегчает назначение вызовам API.

Пример.

C.nnnnnn	содержится в подструктуре общих сведений
P.nnnnnn	содержится в подструктуре протокола
L.nnnnnn	содержится в подструктуре ограничений
S.nnnnnn	содержится в подструктуре масштабирования

2.7 Тег

Теги и соединения состоят из общей части и конкретной части. Конкретную часть всегда предоставляет канальный DLL. Несмотря на то, что эта часть может отсутствовать во время настройки (ее необходимо определить перед активацией), все объекты, в которых отсутствует конкретная часть, будут игнорированы во время импорта. Во время экспорта отсутствующие части будут заменены '*'.

Импорт тегов пользовательского типа.

Предварительно заданный тег WinCC определяется по типу данных со значением от 1 до 18.

Пользовательский тип данных получает в качестве типа данных значение TypID, которое было назначено во время создания структуры данных менеджером данных. Это значение TypID превышает 1000.

Структурный тип определяется по имени и типу данных.

Имя структуры данных одинаково на компьютерах, с которого структура была экспортирована, и на которой она была импортирована. Тем не менее, значение TypID не должно быть или не будет одинаковым.

Чтобы создать тег типа структурного типа, необходимо выполнить назначение из TypID в имя структуры.

Пример.

Структурные типы экспортируются в файл [*имя*]_*dex.csv*.

#Structure Name	Идентификатор типа	Идентификатор создателя	Путь к проекту		
ExternStr1	1046	0	G:\Testdaten\...		
#Varname	C.Vartype	C.CreatorID	C.VarLength	C.VarProperty	...
Tag1	1	0	2	...	...
Tag2	2	0	1	...	...
Tag3	3	0	1	...	...
Tag4	4	0	2	...	...

Теги экспортируются в файл [*имя*]_*vex.csv*.

#Varname	Conn	Группа	Spec	Флаг	CTyp
InstExStr1	VerbLp	GruLp	DB200,DBB10	0	1046
InstExStr1.Tag1	VerbLp	GruLp	DB200,D10.0	0	1
InstExStr1.Tag2	VerbLp	GruLp	DB200,DBB10	0	2
InstExStr1.Tag3	VerbLp	GruLp	DB200,DBB10	0	3
InstExStr1.Tag4	VerbLp	GruLp	DB200,DBB10	0	4

В файле [*имя*]_*dex.csv* определен структурный тип с именем ExternStr1 и TypID 1046.

В файле [*имя*]_*vex.csv* определен структурный тег типа ExternStr1 с именем InstExStr1. Назначение, представляющее структурный тип ExternStr1, основано на значении в столбце CTyp, который содержит значение TypID 1046 этого структурного типа.

Чтобы импортировать структурный тег, структурный тип должен также содержаться в файле [*имя*]_*dex.csv* и теги этого типа в файле [*имя*]_*vex.csv*.

Инструмент Tag Import/Export (Импорт/экспорт тегов) запоминает назначение Name/TypID (Тимя/TypID) для определения количества структурных тегов, даже если значение TypID отличается на целевом компьютере.

Если необходимо импортировать теги "пользовательского" типа без считывания структуры данных во время импорта (например, файл [*имя*]_*dex.csv* отсутствует), значение TypID целевого компьютера этой структуры данных необходимо отредактировать вручную в файле csv. В этом случае значение TypID будет определяться с помощью экспортированного файла [*имя*]_*dex.csv* целевого компьютера, как описано выше.

Свойства тегов отображаются в столбце CPro в виде десятичного значения в экспортированном файле [*имя*]_*vex.csv* . Существуют следующие свойства тегов.

Свойство тега	Десятичное значение	Шестнадцатеричное значение
Внутренний тег с обновлениями во всем проекте	2	2
Внутренний тег с обновлениями на определенном компьютере	8194	2002
Внешний тег	4	4

Например, если внутренние теги с обновлениями на определенном компьютере необходимо экспортировать из WinCC, но внутренние теги с обновлениями во всем проекте необходимо импортировать в WinCC, то значение 8194 свойства тега в столбце CPro можно изменить в файле экспорта на значение 2 для соответствующих тегов. Затем измененный файл экспорта сохраняется и импортируется в WinCC.

2.8 Приложение

```
Поля тегов
----------------
Varname: char szVarName[MAX_DM_VAR_NAME +1];
Conn: char szConnection[MAX_DM_CONNECTION_NAME +3];
Group: char szGroupName[MAX_DM_GROUP_NAME +1];
Spec: char szSpecific[MAX_DM_VAR_SPECIFIC +1];
Flag: DWORD dwFlags;

Общие сведения
Ctyp: DWORD dwVarType; // Тип переменной
CLen: DWORD dwVarLength; // Длина переменной
CPro: DWORD dwVarProperty; // Внутреннее/внешнее свойство переменной
CFor: DWORD dwFormat; // Преобразование формата

Протокол:
P1 : BOOL bTopLimitErr; // ошибка верхнего предела
P2 : BOOL bBottomLimitErr; // ошибка нижнего предела
P3 : BOOL bTransformationErr; // ошибка преобразования
P4 : BOOL bWriteErr; // ошибка чтения
P5 : BOOL bWriteErrApplication;
```

```
P6 : BOOL bWriteErrProcess;
```

Предельные значения

LF1: double dTopLimit;

LF2: double dBottomLimit;

LF3: double dStartValue;

LF4: double dSubstituteValue;

Флаги пределов

L1 : BOOL bTopLimit; // используйте подстановочное значение в верхнем пределе

L2 : BOOL bBottomLimit; // используйте подстановочное значение в нижнем пределе

L3 : BOOL bStartValue; // используйте подстановочное значение при запуске

L4 : BOOL bConnectionErr; // используйте подстановочное значение при ошибочном соединении

L5 : BOOL bTopLimitValid; // действительное значение верхнего предела

L6 : BOOL bBottomLimitValid; // действительное значение нижнего предела

L7 : BOOL bStartValueValid; // действительное значение запуска

L8 : BOOL bSubstValueValid; // действительное подстановочное значение

Значения для поля Ctyp

BIT 1

SBYTE 2

BYTE 3

SWORD 4

WORD 5

SDWORD 6

DWORD 7

FLOAT 8

DOUBLE 9

TEXT_8 10

TEXT_16 11

RAW 12

ARRAY 13

STRUCT 14

BITFIELD_8 15

BITFIELD_16 16

BITFIELD_32 17

TEXTREF 18

Описание полей соединения

Conname: char szConnection[MAX_DM_CONNECTION_NAME +3];

Unit: char szUnitName [MAX_DM_UNIT_NAME+1];

Common: char szCommon [MAX_DM_CON_COMMON +1]

Specific: char szSpecific [MAX_DM_CON_SPECIFIC +1] ;

Симулятор тегов 3

3.1 Инструмент Tag simulator (Симулятор тегов)

Краткое описание

Инструмент **tag simulator** (Симулятор тегов) используется для имитации внутренних тегов и тегов процесса.

Типичная область применения **Tag simulator** (Симулятор тегов) — тестирование конфигурации без подключенных периферийных устройств процесса или с подключенными периферийными устройствами процесса, но без текущего процесса.

При отсутствии подключенного процесса можно имитировать только внутренние теги.

При наличии подключенных периферийных устройств процесса теги процесса можно напрямую предоставлять со значениями с помощью **Tag simulator** (Симулятор тегов). Это позволит выполнить функциональный тест системы HMI с помощью оригинального оборудования.

Время обновления для значений тегов составляет одну секунду. Изменения вступают в силу только при активации функций или изменениях папки проекта.

Можно настроить не более 300 тегов.

Другое возможное применение **Tag simulator** (Симулятор тегов) — реализация проекта для демонстрации.
Часто системное соединение отсутствует для представления системы HMI. В этих случаях имитация управляет внутренними тегами.

Подробное описание симулятора тегов см. в соответствующей интерактивной справке.

⚠ ПРЕДУПРЕЖДЕНИЕ
Симулятор тегов записывает значения процесса в подключенную систему автоматизации. Это означает, что необходимо учитывать реакцию подключенных периферийных устройств процесса.

3.2 Использование Tag Simulator (Симулятор тегов)

Типичная область применения Tag simulator (Симулятор тегов) — тестирование конфигурации без подключенных периферийных устройств процесса или с подключенными периферийными устройствами процесса, но без текущего процесса.

Имитация тегов процесса без периферийных устройств процесса

При отсутствии подключенного процесса можно имитировать только внутренние теги. Чтобы имитировать процесс автономно, рекомендуется выполнить следующие действия.

1. Создайте резервную копию проекта, скопировав папку проекта и переименовав ее, например, в xxx_sim. Используйте эту резервную копию в качестве объекта для проверки. Откройте WinCC и этот скопированный проект.
2. Используйте функции Cut (Вырезать) и Paste (Вставить), чтобы добавить имитируемые теги во внутренние теги. **Не** используйте функции Copy (Копировать) и Paste (Вставить); в противном случае проводник WinCC автоматически создаст расширение для имени тега, чтобы обеспечить уникальность имен тегов в проекте. Следовательно, информация об адресах тегов, объявленных внутренними тегами, будет утеряна.
3. С помощью симулятора теперь в теги можно подставить значения.
4. По завершении проверки можно продолжить работу с оригинальным проектом.

Имитация тегов процесса с подключенными периферийными устройствами процесса

При наличии подключенных периферийных устройств процесса теги процесса можно напрямую предоставлять со значениями с помощью Tag simulator (Симулятор тегов). Это позволит выполнить функциональный тест системы HMI с помощью оригинального оборудования, например:

- Проверка уровней предельных значений, вывод сообщений.
- Проверка связанности аварийных сигналов, предупреждений, сообщений об ошибках и проверка отображения состояний.
- Предварительная установка, считывание и установка цифровых и аналоговых вводов и выводов.
- Имитация аварийного сигнала.

3.3 Функции симулятора

Введение

Симулятор предоставляет конфигуратору шесть различных функций. С помощью этих функций можно подставлять в настроенные объекты реальные значения.

Для проверки различных случаев симулятор содержит 6 функций. Каждый тег можно назначить одной из перечисленных ниже 6 функций.

Sine (Синус)

В качестве периодической нелинейной функции.

Oscillation (Колебание)

Для имитации переходов переменной ссылочного типа.

Random numbers (Случайные числа)

Функция Randon numbers (Случайные числа) позволяет пользователю применять случайно созданные значения.

Increment (Приращение)

Суммирующий счетчик, который снова начинает отсчет с минимального значения после достижения максимального значения.

Decrement (Декремент)

Обратный счетчик, который снова начинает отсчет с максимального значения после достижения минимального значения.

Slider (Ползунок)

С помощью ползунка пользователь может установить фиксированное значение.

3.4 Установка симулятора

Tag simulator (Симулятор тегов) можно установить двумя способами.

Процедура

1. Во время установки WinCC в диалоговом окне Programs (Программы) выберите пункт WinCC V7.0 complete (Полная установка WinCC V7.0).

   WinCC установится вместе с пакетами SmartTools, WinCC ConfigurationTool и WinCC Archive ConfigurationTool.

Запустите Tag simulator (Симулятор тегов), выбрав SIMATIC > WinCC > Tools (Инструменты).

Альтернативная процедура

Приложение Tag simulator (Симулятор тегов) можно также установить с DVD-диска WinCC.

1. Перейдите в каталог на DVD-диске WinCC "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Дважды щелкните значок setup.exe.
3. Выберите в диалоговом окне Components (Компоненты) пункт WinCC Tag Simulator (Симулятор тегов WinCC).
4. Нажмите кнопку Continue (Продолжить). Следуйте инструкциям в диалоговых окнах.

Запуск симулятора

Симулятор Simulation.exe можно запустить с помощью Проводника Windows или этот файл можно внести в список загрузки проводника WinCC, согласно которому он будет автоматически загружаться при открытии проекта.

Для правильной работы симулятора в проводнике WinCC должен быть открыт проект. Если симулятор добавлен в список загрузки проекта, это условие выполняется автоматически.

3.5 Добавление/удаление тегов

Добавление новых тегов

С помощью команд меню Edit/New Tag (Правка/Создать тег) можно добавлять теги в симулятор. Для этого будет вызвано диалоговое окно выбора тегов проводника WinCC, в котором можно выбрать необходимые теги из активного проекта. Если планируется создать новые теги, это можно также выполнить в диалоговом окне выбора. При подтверждении выбора с помощью кнопки OK ранее выбранный тег вводится на вкладке Properties... (Свойства) симулятора. На этой вкладке можно указать способ изменения значения.

Для окончательного приема в симулятор перед добавлением следующего тега необходимо щелкнуть элемент управления вкладки тегов.

Настроенную имитацию тега можно сохранить в файле конфигурации с расширением файла sim.

Удаление тегов

Если тег необходимо удалить из списка симулятора, его необходимо выбрать и удалить, выбрав пункт меню Edit/Delete Tag (Правка/Удалить тег). Выбранный тег будет удален из списка имитируемых тегов без открытия диалогового окна подтверждения.

3.6 Назначение параметров функций

Параметры функций можно установить отдельно для каждого тега.

Sine wave (Синусоида)

Для функции синуса диапазон значений можно установить с помощью параметра **Amplitude** (Амплитуда).

Нулевую точку для диапазона значений можно определить с помощью параметра **Offset** (Смещение).

Период устанавливается с помощью параметра **Period of oscillation** (Период колебания) (уставка * время цикла).

Oscillation (Колебание)

Параметр **Setpoint** (Уставка) используется для определения значения, которое сохраняется после кратковременного ответного действия.

Параметр **Overshoot** (Отклонение) определяет объем отклонения значений от уставки, если для ослабления установлено нулевое значение.

Параметр **Period of oscillation** (Период колебания) определяет интервал времени. По истечении интервала времени колебание начинается заново.

Random numbers (Случайные числа)

Параметры **Lower limit** (Нижняя уставка) и **Upper limit** (Верхняя уставка) определяют интервал для случайных чисел.

Increment (Приращение)

Параметры **Start value** (Начальное значение) и **Stop value** (Окончательное значение) указывают интервал для суммирующего счетчика.

Decrement (Декремент)

Параметры **Start value** (Начальное значение) и **Stop value** (Окончательное значение) указывают интервал для обратного счетчика.

Slider (Ползунок)

Параметры **Start value** (Начальное значение) и **Stop value** (Окончательное значение) определяют диапазон регулировки ползунка.

3.7 Включение/отключение тегов

Для плавного перехода от автономной к интерактивной настройке можно отдельно включать и отключать теги с помощью специального флажка.

Если тег включен, симулятор вычисляет значения и переносит их в проводник WinCC.

Если флажок не установлен, симулятор не передает значения в проводник WinCC.

3.8 Отображение тегов

На этапе конфигурации управление тегами упрощается благодаря отображению следующей информации на вкладке Tags (Теги):

- текущий проект WinCC;
- имя тега;
- назначенная функция;
- состояние (включен/отключен);
- текущее значение.

При выборе имени тега остальные параметры вводятся на вкладке Properties (Свойства).

3.9 Загрузка/сохранение данных имитации

Данные имитации можно сохранить, чтобы они были доступны при повторном запуске симулятора. Для этого выберите пункты меню File/Save (Файл/Сохранить) или File/Save as... (Файл/Сохранить как).

Сохраненная конфигурация симулятора загружается с помощью команд меню File/Open (Файл/Открыть).

Во время запуска симулятора последняя использовавшаяся конфигурация, относящаяся к проекту WinCC, загружается автоматически.

3.10 Часто задаваемые вопросы

Ошибка DM-API, файл DLL не найден

Ошибки, возникающие при вызове симулятора и связанные с DLL, происходят из-за отсутствия указания пути в файле AUTOEXE.BAT. Проверьте этот файл на наличие следующей записи в указании пути:

SET PATH = ...;<WinCC drive>:\<WinCC directory>\bin

например: SET PATH=C:\SIEMENS\WINCC\BIN;

Редактор мастера динамики 4

4.1 Редактор мастера динамики

Краткое описание

Dynamic Wizard Editor (Редактор мастера динамики) представляет собой инструмент для создания собственных мастеров динамики. Мастера динамики позволяют автоматизировать часто повторяющиеся процессы конфигурации.

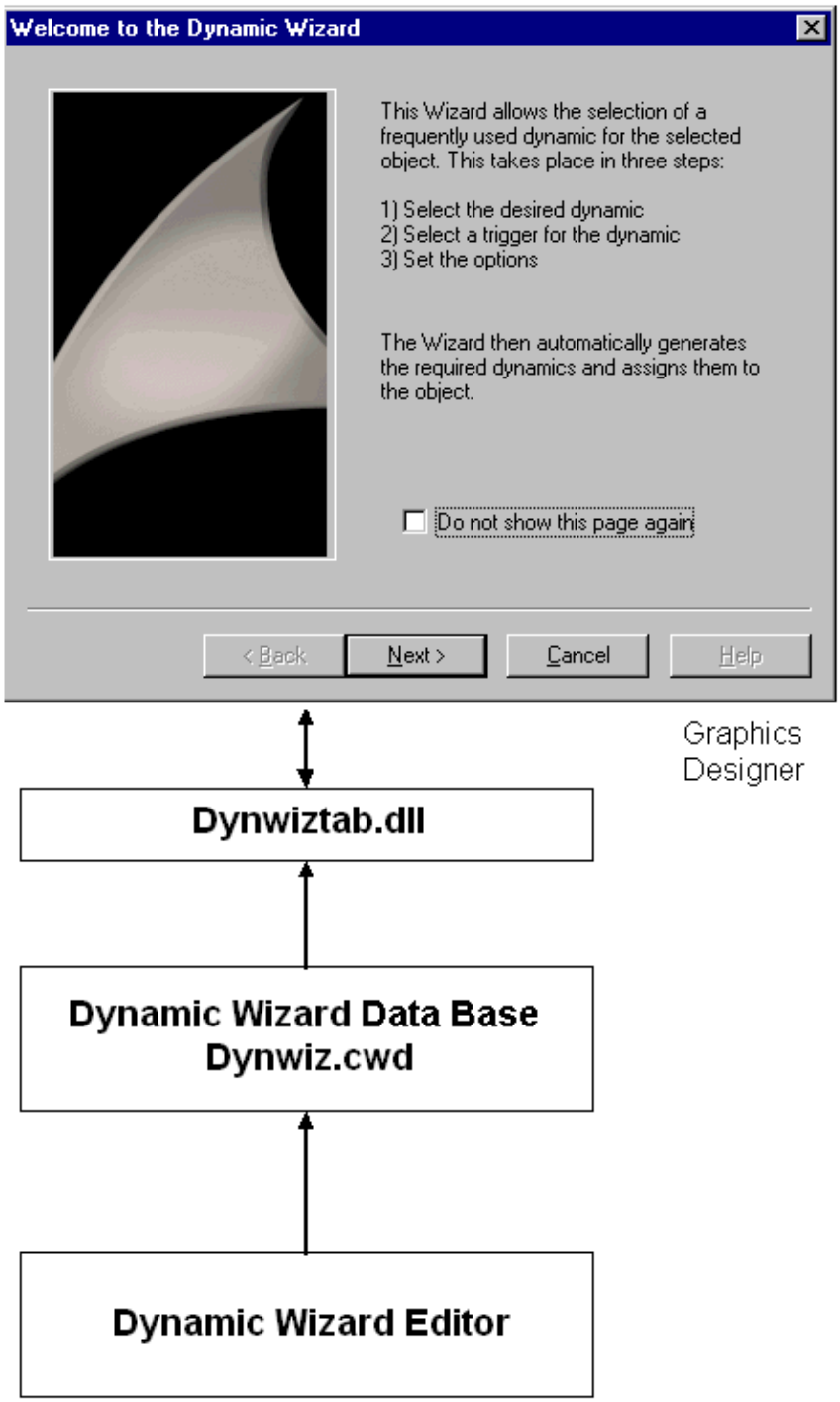
4.2 Обзор

Введение

Dynamic Wizard (Мастер динамики) обеспечивает дополнительные возможности графического дизайнера. Он помогает пользователю обрабатывать часто повторяющиеся последовательности конфигурации. Это облегчает конфигурацию и исключает вероятность ошибок конфигурации.

Dynamic Wizard (Мастер динамики) состоит из множества функций. Большинство функций Dynamic Wizard (Мастер динамики) уже включены в систему. Их можно также дополнить собственными функциями.

Для создания собственных функций Dynamic Wizard (Мастер динамики) предусмотрен редактор. Этот редактор является программой dynwizedit.exe.



Все функции Dynamic Wizard (Мастер динамики) хранятся в базе данных (...\WinCC\wscript\Dynwiz.cwd) на жестком диске. Dynamic Wizard (Мастер динамики) имеет стандартизированное отображение и интерфейс пользователя для выбора и определения параметров функций мастера динамики. После выбора функции мастера динамики она будет загружена в память и выполнена.

Взаимодействие мастера динамики и функции мастера динамики

Связь между Dynamic Wizard (Мастер динамики) и функцией мастера динамики осуществляется по интерфейсу системы в функции мастера динамики. Структура этого интерфейса определена. Это интерфейс содержит информацию, которую может оценить Dynamic Wizard (Мастер динамики).
Основное содержимое интерфейса.

Ссылка на функцию процесса

Функция процесса — назначенная основная функция мастера динамики. Она содержит службу, которую предоставляет пользователю функция мастера динамики, например создание действия для графического объекта.

Список параметров определяет параметры, которые необходимы для функции процесса. Кроме того, он определяет способ их указания с помощью интерфейса пользователя диалогового окна.

Список триггеров List (Список) определяет триггеры, которые будут связаны с создаваемым объектом. Кроме того, он определяет способ их указания с помощью интерфейса пользователя диалогового окна.

Дополнительные источники информации

Список триггеров (стр. 42)

Список параметров (стр. 40)

4.3 Установка редактора мастера динамики

Dynamic Wizard Editor (Редактор мастера динамики) можно установить двумя способами.

Процедура

1. Во время установки WinCC в диалоговом окне Programs (Программы) выберите пункт WinCC V7.0 complete (Полная установка WinCC V7.0).

   WinCC установится вместе с пакетами SmartTools, WinCC ConfigurationTool и WinCC Archive ConfigurationTool.

Запустите Dynamic Wizard Editor (Редактор мастера динамики), выбрав SIMATIC > WinCC > Tools (Инструменты).

Альтернативная процедура

Приложение Dynamic Wizard Editor (Редактор мастера динамики) можно также установить с DVD-диска WinCC.

1. Перейдите в каталог на DVD-диске WinCC «InstData\WinCC\setup\Products\SC_SMARTTOOLS».
2. Дважды щелкните значок setup.exe.
3. Выберите Dynamic Wizard Editor (Редактор мастера динамики) в диалоговом окне Components (Компоненты).

4. Нажмите кнопку Next (Далее). Следуйте инструкциям в диалоговых окнах.

4.4 Структура

4.4.1 Структура

Dynamic Wizard Editor (Редактор мастера динамики) состоит из следующих элементов:

Строка меню

В строке меню содержатся функции Dynamic Wizard Editor (Редактор мастера динамики). Строка меню отображается всегда.

Панель инструментов

Панель инструментов можно отобразить по мере необходимости и перетащить в любое место на экране с помощью мыши.

Окно редактора

Окно редактора отображается, только если открыта функция мастера динамики для редактирования или создана новая функция. Каждая функция открывается в собственном окне редактирования. Можно открыть одновременно несколько окон редактирования.

Окно вывода

Окно вывода можно открыть по мере необходимости. В нем отображается результат функции Create CWD (Создание CWD), Read Wizard Script (Считывание сценария мастера) и Compile Script (Компиляция сценария).

Строка состояния

Строку состояния можно отобразить по мере необходимости. В ней содержится информация о настройке клавиатуры и положении курсора в окне редактирования.

Мастер динамики

В Dynamic Wizard (Мастер динамики) можно динамизировать объект с помощью C-макросов. При использовании мастеров предварительно настроенные C-макросы и инициирующие события определяются и передаются в свойства объекта.

Дополнительные источники информации

Окно вывода (стр. 33)

Окно редактора (стр. 31)

Панель инструментов (стр. 29)

4.4.2 Панель инструментов

Введение

Панель инструментов позволяет быстрее выполнять действия. Благодаря панели инструментов нужную функцию больше не нужно выбирать через целый ряд меню.

Значки

Значок	Описание
	Создание новой функции Dynamic Wizard (Мастер динамики).
	Открытие существующей функции Dynamic Wizard (Мастер динамики) (*.wnf).
	Сохранение функции Dynamic Wizard (Мастер динамики).
	Вырезание выбранного текста и его копирование в буфер обмена.
	Копирование выбранного текста в буфер обмена.
	Вставка содержимого буфера обмена в местоположении курсора.
	Печать содержимого текущего окна редактирования.
	Отображение дополнительной информации по Dynamic Wizard Editor (Редактор мастера динамики).
	Создание файла данных Dynamic Wizard Data (CWD). Эта функция считывает все существующие сценарии мастеров на выбранном в настоящее время языке и подготавливает их к обработке в Dynamic Wizard (Мастер динамики). Созданный файл находится в папке установки WinCC (Папка_установки\wscripts\dynwiz.cwd).
	Считывание сценариев мастеров и предоставление доступа к ним в Dynamic Wizard (Мастер динамики).

Значок	Описание
	Установка языка, для которого настроен сценарий мастера. В этом списке можно выбрать известные в WinCC языки независимо от установленных языков. Изменение языка мастера не влияет на общий интерфейс системы и конфигурации.
	Изменение объекта. Dynamic Wizard (Мастер динамики), который также доступен в редакторе для тестирования, зависит от различных свойств объекта в графическом дизайнере. Эта функция позволяет переключаться на существующий объект в имеющемся кадре таким образом, что новый или существующий сценарий мастера можно проверить в редакторе. Новый заданный объект влечет за собой отображение в Dynamic Wizard (Мастер динамики) только тех сценариев мастеров, которые подходят для этого объекта.
	Отображение всех сценариев Dynamic Wizard (Мастер динамики) для выбранного языка. Кроме того, сценарии мастеров в диалоговом окне можно удалить из списка.
	Открытие редактора справки.
	Компилирование сценария.

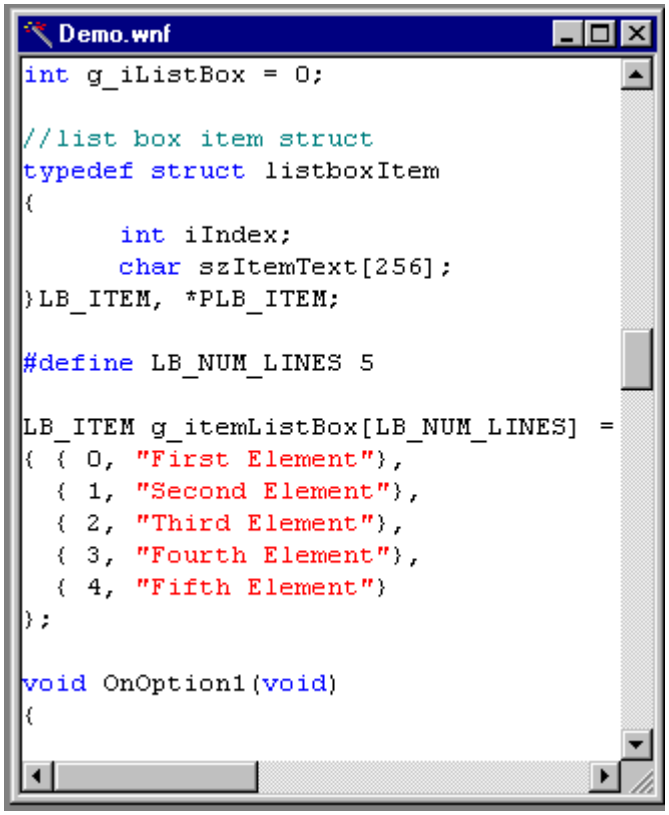
Дополнительные источники информации

Редактор справки (стр. 32)

4.4.3 Окно редактора

Введение

Окно редактора служит для создания и редактирования функций Dynamic Wizard (Мастер динамики).



Цветовая маркировка

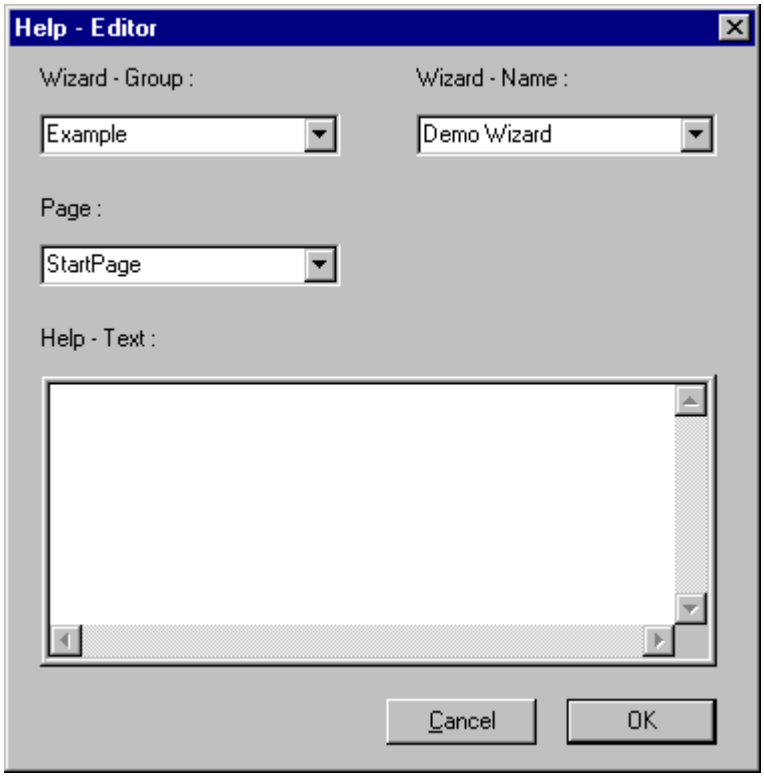
Код C code отображается с использованием следующих цветов.

Цвет	Значение	Пример
синий	Ключевые слова	#define, void
зеленый	Комментарии	// это комментарий
красный	Строки	First Element
черный	остальной код C	OnOption1

4.4.4 Редактор справки

Введение

В этом окне можно ввести текст справки для каждой страницы, созданной с помощью сценария мастера. Можно ввести только уже созданные тексты справки для мастеров динамики.



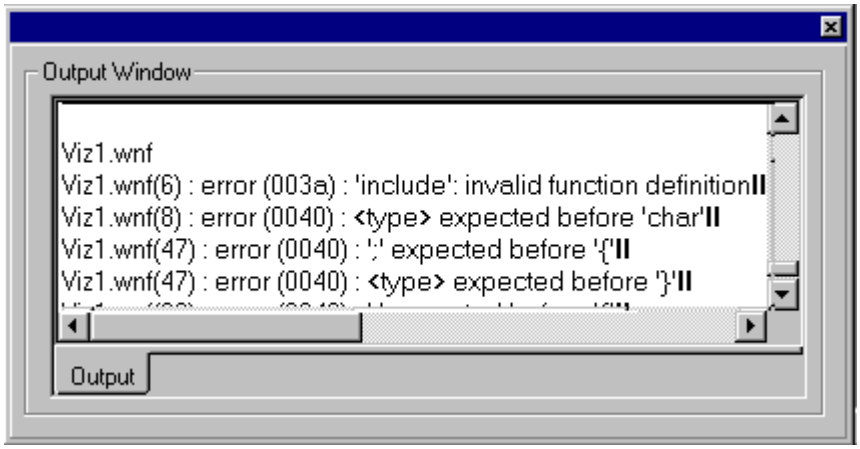
Элементы редактора справки

Элемент	Описание
Wizard group (Группа мастера)	В этом поле указывается группа (=вкладка), содержащая Dynamic Wizard (Мастер динамики).
Wizard name (Имя мастера)	Это поле используется для выбора Dynamic Wizard (Мастер динамики), для которого создается текст справки.
Page (Страница)	Это поле используется для выбора страницы диалогового окна, для которой создается текст справки.
Help text «Текст справки»	В это поле вводится текст справки.

4.4.5 Окно вывода

Введение

В окне вывода отображается результат функций Create CWD (Создание CWD), Read Wizard Script (Считывание сценария мастера) и Compile Script (Компиляция сценария).



Окно вывода облегчает поиск ошибок в сценариях.

При наличии ошибки в сценарии отображается следующее сообщение:

...\WinCC\wscripts\wscripts.deu\DemoWiz1.wnf(6):error(003a): 'include': invalid function definition (...\WinCC\wscripts\wscripts.deu\DemoWiz1.wnf(6):ошибка(003a): 'include': неверное определение функции)

	Описание
...\WinCC\wscripts\wscripts.deu\	Каталог, в котором находится файл wnf.
DemoWiz1.wnf(6)	Имя файла и номер строки, в которой происходит ошибка
error(003a): 'include': invalid function definition	Номер ошибки и ее описание.

4.5 Структура функции мастера динамики

4.5.1 Структура функции мастера динамики

Введение

Функция Dynamic Wizard (Мастер динамики) должна иметь определенную заданную структуру. Она должна соответствовать необходимым компонентам.

1. Интеграция заголовочных файлов и DLL

2. Определения, зависящие от установленного языка
3. Флаги мастера
4. Список свойств
5. Интерфейс системы
6. Глобальные переменные
7. Список параметров
8. Список триггеров
9. Отображение назначения параметров

Дополнительные источники информации

Отображение назначения параметров (стр. 45)
Список триггеров (стр. 42)
Список параметров (стр. 40)
Глобальные переменные (стр. 39)
Интерфейс системы (стр. 37)
Список свойств (стр. 37)
Флаги мастера (стр. 36)
Определения, зависящие от установленного языка (стр. 35)
Интеграция заголовочных файлов и DLL (стр. 34)

4.5.2 Диалоговое окно Dynamic Wizard (Мастер динамики)

Введение

Каждый параметр Dynamic Wizard (Мастер динамики) имеет собственную особую функцию. Однако из-за предварительно определенной структуры все функции имеют похожие последовательность и интерфейс диалогового окна. Диалоговое окно Dynamic Wizard (Мастер динамики) состоит из нескольких страниц диалогового окна.

- Диалоговое окно Welcome to the Dynamic Wizard (Приветствие мастера динамики)
- Диалоговое окно Select trigger (Выбор триггера)
- Диалоговое окно Set options (Настройка параметров)
- Диалоговое окно Finished ! (Готово!)

4.5.3 Интеграция заголовочных файлов и DLL

Введение

Заголовочный файл содержит объявления констант, типов данных, тегов и функций.

Заголовочные файлы интегрируются в функцию с помощью команды #include. Крайне важно интегрировать файл dynamic.h, в котором, помимо прочего, объявлены функции для проектирования интерфейса Dynamic Wizard (Мастер динамики).

```
//*****************************************************
//**        Integration of Header-Files               **
//*****************************************************
            #include "dynamic.h"
```

Файлы DLL (Dynamic Link Library — динамически подключаемая библиотека) представляют собой исполняемые программы, которые могут загружаться программой по мере необходимости.

Для использования файлов DLL они интегрируются в функцию с помощью команды #pragma.

```
//*************************************************
//**        Integration of Dlls                   **
//*************************************************
            #pragma code("pdlcsapi.dll")
            #include "pdlcsapi.h"
            #pragma code()
```

В Dynamic Wizard Editor (Редактор мастера динамики) определены следующие пути:

Заголовочные файлы WinCC: ...\WinCC\aplib\

Файлы DLL WinCC: ...\WinCC\bin\

Если файлы необходимо сохранить в другом каталоге, в командах `#include-` и `#pragma` необходимо указать полный путь.

4.5.4 Определения, зависящие от установленного языка

Введение

Стандартные функции Dynamic Wizard (Мастер динамики) доступны на трех языках: немецкий, английский и французский. При изменении языка в проводнике WinCC соответствующая языковая версия также выбирается для функций Dynamic Wizard (Мастер динамики).

В указанных ниже путях

..\WinCC\wscripts\wscripts.deu

..\WinCC\wscripts\wscripts.enu

..WinCC\wscripts\wscripts.fra

должен существовать файл WNF для каждой функции мастера.

При создании все определения, зависящие от установленного языка, необходимо упорядочить в этом разделе. Это упростит создание других языковых версий.

```
//**************************************************************************//
//                  Language-Dependent Definitions                          //
//**************************************************************************//
//                         German                                           //
//--------------------------------------------------------------------------//
include  "defdeu.h"

char* DynWizGroupName     = "WinCC C-Kurs";
char* DynWizDynamicName   = "Motor dynamisieren";
char* DynWizToDoOption1   = "Wählen Sie die gewünschte Strukturvariable:";
//--------------------------------------------------------------------------//
//                         Englisch                                         //
//--------------------------------------------------------------------------//
#include  "defenu.h"

char* DynWizGroupName     = "WinCC C-Course";
char* DynWizDynamicName   = "Make a Motor Dynamic";
char* DynWizToDoOption1   = "Select the desired Structure Tag:";
//--------------------------------------------------------------------------//
//                        French                                            //
//--------------------------------------------------------------------------//
#include  "deffra.h"

char* DynWizGroupName     = "Cours de C WinCC";
char* DynWizDynamicName   = "Dynamiser moteur";
char* DynWizToDoOption1   = "Sélectionnez la variable de structure:";
```

4.5.5 Флаги мастера

Введение

Эти флаги используются для определения типа конфигурации, к которой применяется функция Dynamic Wizard (Мастер динамики).

```
WIZARD_FLAGS(WIZARD_FLAG_OCX | WIZARD_FLAG_ALL_PROJECT_TYPES)

BEGIN_PROPERTY_SCHEME
END_PROPERTY_SCHEME
```

Флаги

ФЛАГ	
WIZARD_FLAG_OCX	Для всех файлов OCX
WIZARD_FLAG_ALL_PROJECT_TYPES	Для всех проектов
WIZARD_FLAG_SINGLEUSER_PROJECT	Только для однопользовательских проектов
WIZARD_FLAG_MULTICLIENT_PROJECT	Для проектов клиента
WIZARD_FLAG_MULTIUSER_PROJECT	Только для клиентов без данных проекта

4.5.6 Список свойств

Введение

Список свойств определяет типы объектов, для которых можно использовать функцию Dynamic Wizard (Мастер динамики). Для этого укажите список свойств объектов. Если объект обладает, по меньшей мере, одним из указанных свойств, к нему будет применена функция Dynamic Wizard (Мастер динамики).

```
//************************************************
//** Objektauswahl mittels Objekteigenschaften  **
//************************************************

BEGIN_PROPERTY_SCHEME
 {"BackColor", VT_I4},
END_PROPERTY_SCHEME
```

Каждая запись в списке свойств состоит из двух параметров:

- Имя свойства, например Backcolor в английском варианте.
- Тип данных WinCC

Если используется пустой список свойств, функцию Dynamic Wizard (Мастер динамики) можно применить ко всем типам объектов. В любом случае должен существовать список свойств, даже если он пуст.

4.5.7 Интерфейс системы

Введение

Интерфейс системы используется для определения свойств новой функции Dynamic Wizard (Мастер динамики).

```
BEGIN_DYNAMICS
      {
      DynWizGroupName,          // 1. Parameter
      DynWizDynamicName,        // 2. Parameter
      NULL,                     // 3. Parameter
      "logo16.bmp",             // 4. Parameter
      DynWizHelpText,           // 5. Parameter
      {                         // 6. Parameter
      // "OnOption1",
      // "OnOption2",
      NULL
      },
      "OnGenerate",             // 7. Parameter
      "OnShowGenerateInfo",     // 8. Parameter
      {                         // 9. Parameter
      // PREDEFINED_MACRO,
      // {DynWizTrigger1Text, OnTrigger1},
      {NULL,NULL}
      },
      },
END_DYNAMICS
```

Описание параметра

1. Первый параметр определяет вкладку, на которой должна отображаться функция Dynamic Wizard (Мастер динамики).
2. Второй параметр определяет имя, под которым должна отображаться функция Dynamic Wizard (Мастер динамики).
3. Значение третьего параметра — всегда ZERO.
4. Четвертый параметр назначает имя значка, используемого для функции Dynamic Wizard (Мастер динамики).
5. Пятый параметр представляет собой текст справки, содержащий более подробное описание работы функции Dynamic Wizard (Мастер динамики).
6. Шестой параметр представляет собой список с названиями функций, созданных для отдельных страниц параметров. Данный список заканчивается записью ZERO. Можно создать не более пяти страниц параметров. Дополнительную информацию по этой теме см. в разделе «Список параметров».
7. Седьмой параметр представляет собой название функции процесса, вызванной после нажатия кнопки Complete (Готово). Функция процесса — назначенная основная функция мастера динамики. Она содержит службу, которую предоставляет пользователю функция мастера динамики, например создание действия для графического объекта.
8. Восьмой параметр представляет собой название функции, которая предоставляет сводку настроек на страницах параметров и отображает их для пользователя, прежде чем будет нажата кнопка Complete (Готово). Дополнительную информацию по этой теме см. в разделе «Отображение назначения параметров».
9. Девятый параметр представляет собой список триггеров, отображаемых на странице триггеров. Для большинства применений этот список триггеров можно заполнить макросами. Дополнительную информацию по этой теме см. в разделе «Список триггеров».

4.5.8 Глобальные переменные

Введение

Для каждого параметра, устанавливаемого на страницах параметров, необходимо определить глобальную переменную. Это необходимо для того, чтобы заданные параметры были известны во всех созданных функциях, и их можно было использовать.

Передача данных между функциями системы возможна только с помощью глобальных переменных. Это всегда необходимо, когда требуется передать параметры триггеров и/или вариантов в функцию процесса.

```
//****************************************
//        Definition of Global Tags
//****************************************

char g_Demo_Typ ="Demo"
```

4.5.9 Список параметров

Введение

Параметры необходимы для работы функции Dynamic Wizard (Мастер динамики). Для параметров не требуется триггер.

Параметры определяются в списке параметров интерфейса системы. Для каждого параметра в списке указано имя назначенной функции, например OnOption1.

```
BEGIN_DYNAMICS
{
DynWizGroupName,
DynWizDynamicName,
NULL,
"logo16.bmp",
DynWizHelpText,
//*******************************
//          Optionenliste
//*******************************
{
"OnOption1",
"OnOption2",
NULL
},
"OnGenerate",
"OnShowGenerateInfo",
{ // Triggerliste
{ NULL, NULL }
},
},
END_DYNAMICS
```

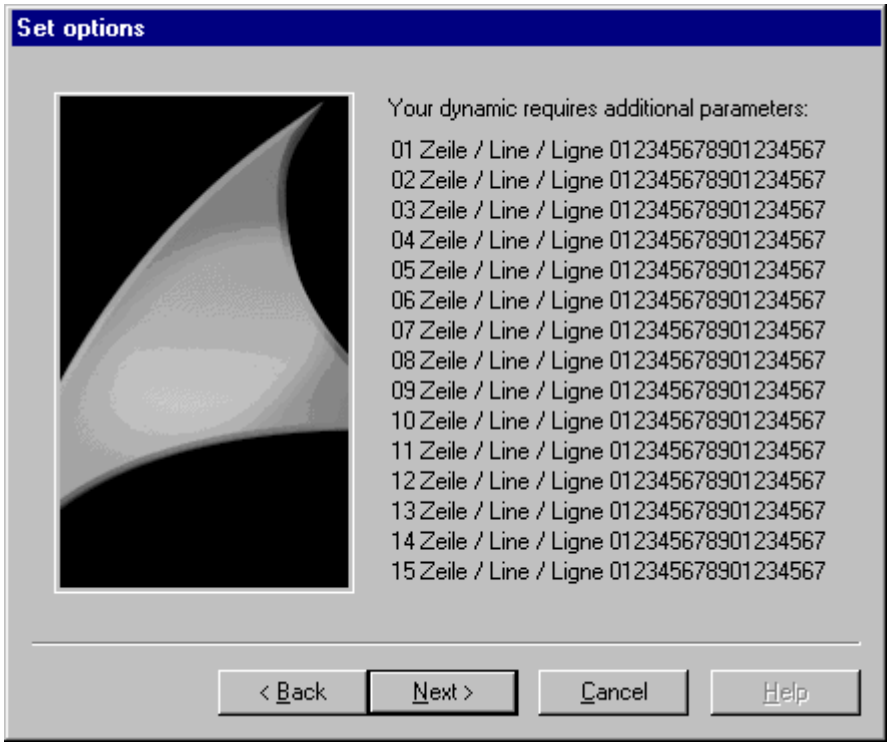
Список параметров заканчивается указателем ZERO. В списке можно определить не более пяти параметров.

Функции параметров

Dynamic Wizard (Мастер динамики) последовательно вызывает функции параметров согласно их порядку в списке параметров. Для каждой функции параметра отображается диалоговое окно Set options (Настройка параметров), в котором функция программирует определенную запись.

Для программирования записи используются функции системы мастера. Дополнительную информацию по этой теме см. в разделе «Функции системы мастера».

В диалоговом окне Set options (Настройка параметров) доступна заданная область для упорядочивания статических текстов, полей ввода и других окон ввода.



В диалоговом окне Set options (Настройка параметров) вся область заполнена строками 1-15.

Ниже указана соответствующая функция параметра:

```
//---------------------------------------------------------------------
//     Option-Funktion OnOption1
//---------------------------------------------------------------------

void OnOption1(void)
{
CreateStatic(0, 0,"01 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0, 15,"02 Zeile / Line / Ligne 012345678901234567");
CeateStatic(0, 30,"03 Zeile / Line / Ligne 012345678901234567");
CeateStatic(0, 45,"04 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0, 60,"05 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0, 75,"06 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0, 90,"07 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,105,"08 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,120,"09 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,135,"10 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,150,"11 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,165,"12 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,180,"13 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,195,"14 Zeile / Line / Ligne 012345678901234567");
CreateStatic(0,210,"15 Zeile / Line / Ligne 012345678901234567");
}
```

4.5.10 Список триггеров

Введение

Триггеры необходимы только для действий в графическом объекте.

Триггеры определяются в списке триггеров интерфейса системы. В списке триггеров содержится запись для каждого триггера.

```
BEGIN_DYNAMICS
{
DynWizGroupName,
DynWizDynamicName,
NULL,
"logo16.bmp",
DynWizHelpText,
"OnOption1",
"OnOption2",
NULL
},
"OnGenerate",
"OnShowGenerateInfo",
{
//*********************************
//          Trigger  list
//*********************************
{ "Mouse click" , "OnTriggerMC" },
{ "Pressing left mouse key" , "OnTriggerLMDown" },
{ "Releasing left mouse key" , "OnTriggerLMUp" },
{ "Pressing right mouse key" , "OnTriggerRMDown" },
{ "Releasing right mouse key" , "OnTriggerRMUp" },
{ NULL, NULL }
      },
      },
END_DYNAMICS
```

Запись состоит из двух параметров. Первый параметр — обозначение триггера, которое отображается в интерфейсе, например Click left mouse button (Щелчок левой кнопкой мыши). Второй параметр определяет имя назначенной функции триггера.

Список триггеров заканчивается парой указателей ZERO. В списке можно определить не более 50 триггеров.

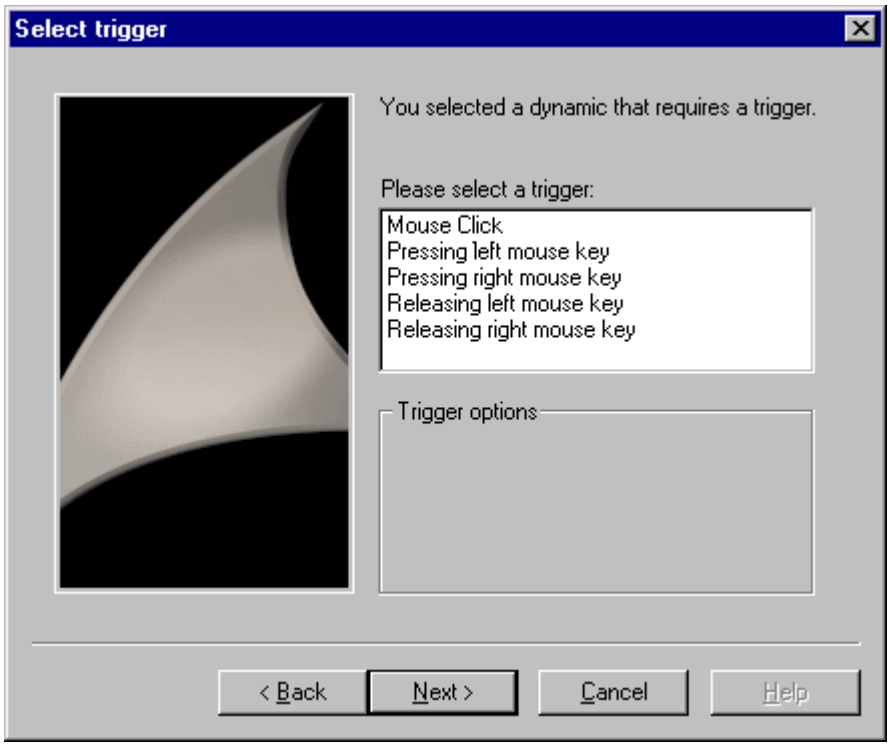
Для наиболее часто используемых триггеров доступны предварительно заданные макросы.

Макрос	
JCR_TRIGGERS	События триггера DECLARE_JCR_TRIGGERS Mouse click (При нажатии кнопки мыши), Left mouse button (При нажатии левой кнопки мыши), Right mouse button (При нажатии правой кнопки мыши)
JCR_ZYCL_TRIGGERS	Цикличные триггеры DECLARE_JCR_ZYKL_TRIGGERS Picture cycle (Цикл кадра), Window cycle (Цикл окна), Upon change (При изменении), 250 ms (250 мс), 500 ms (500 мс), 1 second (1 секунда), 2 seconds (2 секунды), 5 seconds (5 секунд), 10 seconds (10 секунд), 1 minute (1 минута), 5 minutes (5 минут), User cycle 1 (Пользовательский цикл 1), User cycle 2 (Пользовательский цикл 2), User cycle 3 (Пользовательский цикл 3), User cycle 4 (Пользовательский цикл 4), User cycle 5 (Пользовательский цикл 5)
JCR_ACTION_TRIGGERS	Триггеры макросов DECLARE_JCR_ACTION_TRIGGERS

```
...
...
BEGIN_PROPERTY_SCHEME
END_PROPERTY_SCHEME
BEGIN_DYNAMICS
      {
       "System Functions",
       "Exit WinCC Runtime",
       NULL,
       "logo16.bmp",
       "Exits WinCC Runtime and switches to \r\nthe DESIGN Mode.",
       { NULL, NULL, },
       "OnGenerate",
       "OnShowGenerateInfo",
       {
            JCR_TRIGGERS,
       },
END_DYNAMICS

DECLARE_JCR_TRIGGERS
...
...
```

Диалоговое окно Select trigger (Выбор триггера) создается из списка триггеров. Все обозначения триггеров отображаются в списке для выбора.



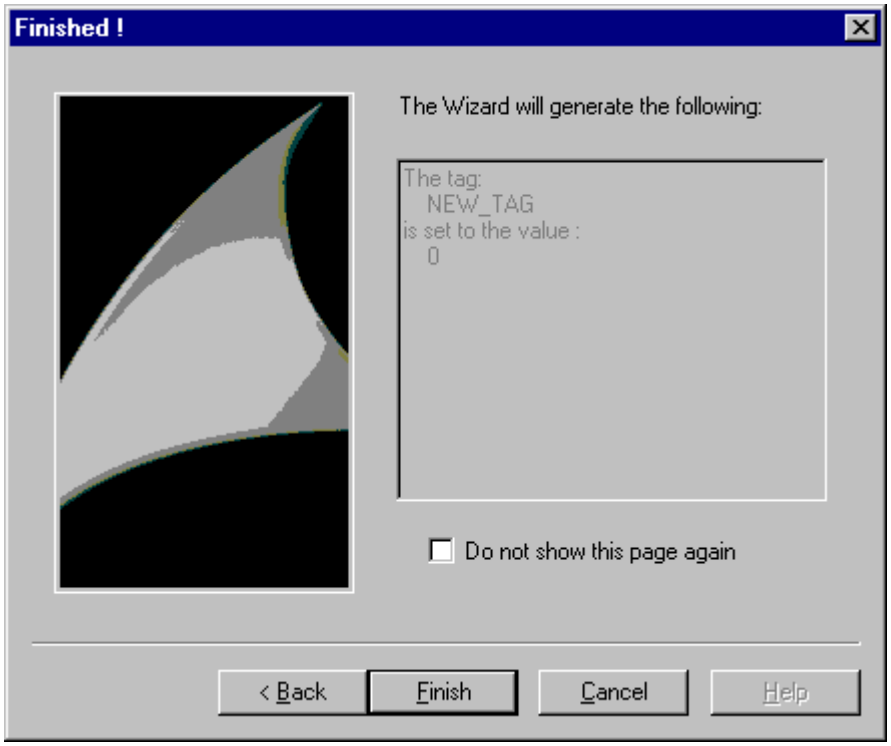
После выбора триггера Dynamic Wizard (Мастер динамики) вызывает назначенную функцию триггера.

4.5.11 Отображение назначения параметров

Введение

Параметры триггера и другие параметры отображаются в диалоговом окне Finished ! (Готово!). В этом окне пользователь может повторно проверить настройку параметров и изменить ее в случае необходимости.

В поле отображения страницы Finished ! (Готово!) можно вывести тест с помощью функций Windows — SetWindowText. Высота поля отображения — 12 строк.



4.5.12 Функции мастера для ввода параметров

4.5.12.1 CreateStatic

Введение

В диалоговом окне Set options (Настройка параметров) отображается статический текст для координат x, y.

Синтаксис

HWND CreateStatic (int x, int y, char* "Text")

Параметры

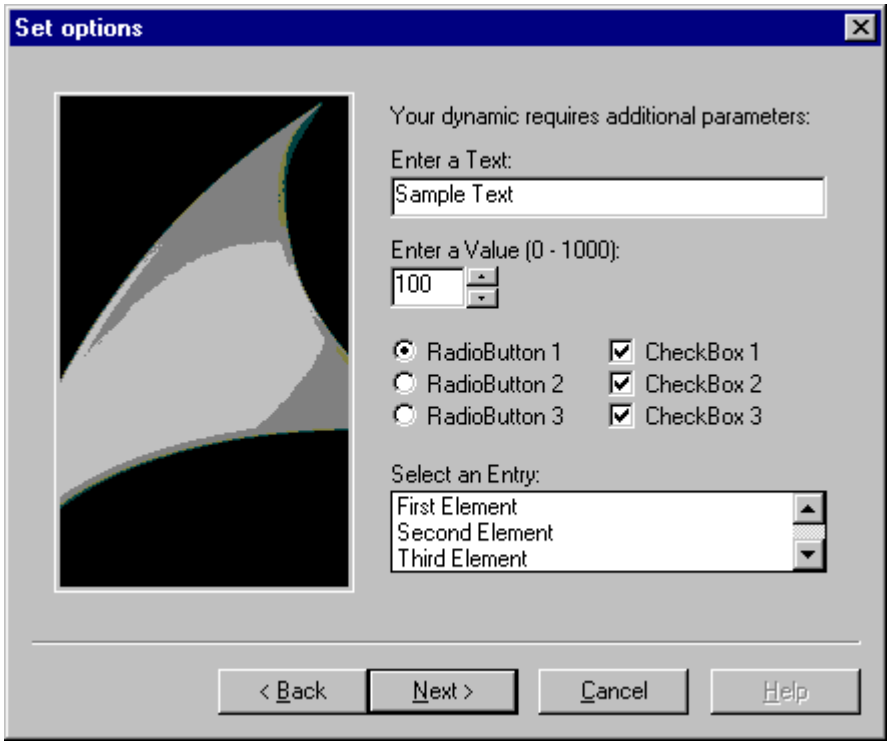
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
char* Text	Отображает выводимый текст.

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции.




```
char* DynWizEditStatic = "Введите текст:";
...
..
void OnOption1(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
.....
```

```
if (bFirst == TRUE)
{
strcpy(g_szEdit,DynWizEdit);
bFirst = FALSE;
}
//Статический текст
CreateStatic(0,5,DynWizEditStatic);
.....
......
}
```

4.5.12.2 CreateEdit

Введение

В диалоговом окне Set options (Настройка параметров) отображается поле ввода для координат x, y. Текст можно ввести в это поле ввода.

Синтаксис

HWND CreateEdit (int x, int y, char* pText)

Параметры

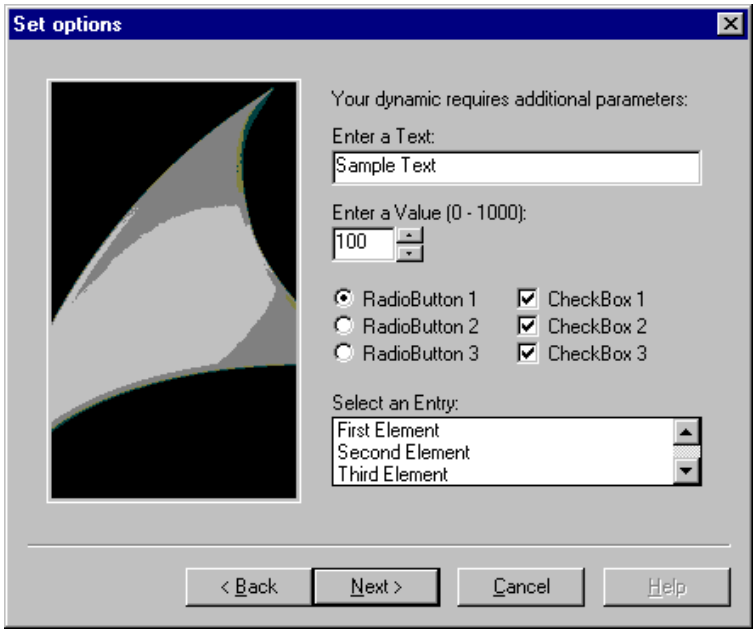
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
char* pText	Указатель на буфер ввода. Буфер ввода может иметь предварительно введенное значение. Оно отображается в поле ввода.

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
pText	Буфер ввода содержит введенный текст.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции.
Поле ввода отображается в диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации).




```
char* DynWizEditStatic = "Введите текст:";
char* DynWizEdit = "Пример текста";
...
..
char g_szEdit[256];
void OnOption1(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
.....
if (bFirst == TRUE)
{
strcpy(g_szEdit,DynWizEdit);
bFirst = FALSE;
}
```

//Статический текст для поля ввода

```
CreateStatic(0,5,DynWizEditStatic);
```

//Поле ввода

```
hWnd = CreateEdit(0,20,g_szEdit)
GetWindowRect(GetParent(hWnd), &rect);
MoveWindow(hWnd,0,20,(rect.right-rect.left),21,TRUE);
.....
......
}
```

4.5.12.3 CreateSpinEdit

Введение

В диалоговом окне Set options (Настройка параметров) отображается поле ввода с элементами управления для координат x, y.

Это поле ввода используется для ввода целого значения в переменную записи.

Синтаксис

HWND CreateSpinEdit (int x, int y, int* pValue, int Min, int Max, int Base)

Параметры

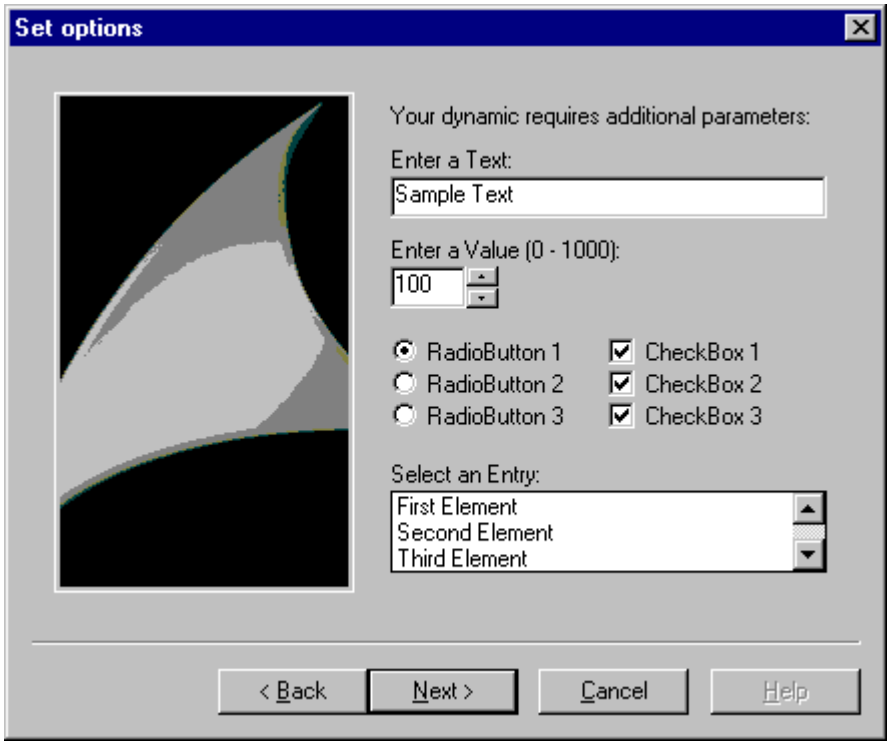
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
int* pValue	Указатель на переменную записи целого значения. Переменная записи может иметь предварительно заданное значение по умолчанию.
int Min	Нижняя уставка для входного значения
int Max	Верхняя уставка для входного значения
int Base	Формат числа для ввода: 10 = ввод десятичного значения 16 = ввод шестнадцатеричного значения

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
pValue	Переменная ввода содержит введенное значение.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции.
Поле ввода с элементами управления отображается в диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации). Можно выбрать значение от 0 до 1000.




```
char* DynWizSpinStatic= "Введите значение (0-1000):";
char* DynWizEdit = "Пример текста";
...
...
char g_szEdit[256];
void OnOption1(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
.....
if (bFirst == TRUE)
{
strcpy(g_szEdit,DynWizEdit);
bFirst = FALSE;
}
...
...
```

```
//Статический текст для поля ввода с элементами управления
CreateStatic(0,50,DynWizSpinStatic);
...
//Поле ввода с элементами управления
hWnd = CreateSpinEdit(0,65,&g_iSpinEdit,0,1000,10);
MoveWindow(hWnd,0,65,(rect.right-rect.left)/4,21,TRUE);

...
...
}
```

4.5.12.4 CreateListBox

Введение

В диалоговом окне Set options (Настройка параметров) отображается поле для выбора для координат x, y. В поле для выбора может содержаться несколько записей. Щелчком мыши можно выбрать одну запись.

Синтаксис

HWND CreateListbox (int X, int Y, char* Headline, int NumLines, int* pSelect)

Параметры

Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
char* Headline	Заголовок поля для выбора
int NumLines	Число строк в поле для выбора. Необходимо указать следующие данные: NumLines = число строк + 1 (1 =< NumLines = <16)
int* pSelect	Указатель на переменную результата

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
pSelect	Номер выбранной записи. Номер представляет собой индекс в списке (начиная с 0).

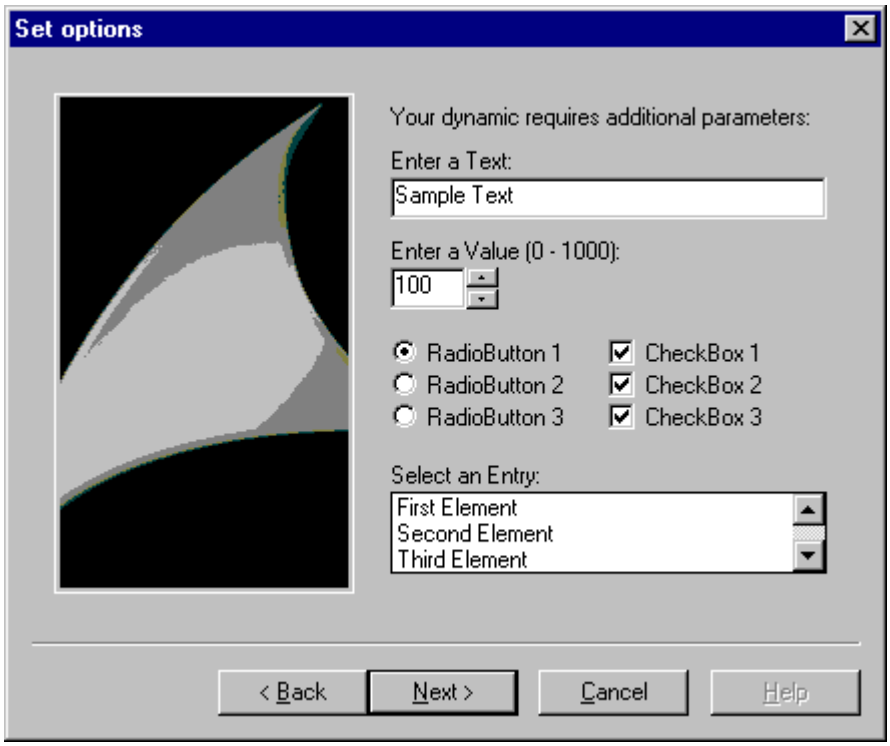
Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображается поле для выбора. Возможный размер поля для

выбора — три строки. При наличии более трех записей отображается строка состояния.

Примечание

С помощью функции CreateListbox создается только поле для выбора. Содержимое строки необходимо ввести с помощью функции SendMessage.




```
char* DynWizListStatic= "Выберите запись:";
...
int g_iListBox = 0;
```

//Введите определение элементов в поле для выбора

```
typedef struct listboxItem
{
int iIndex;
char szItemText[256];
}LB_ITEM, *PLB_ITEM;

#define LB_NUM_LINES 5

LB_ITEM g_itemListBox[LB_NUM_LINES] =
{
{ 0, "Первый элемент"},
  { 1, "Второй элемент"},
  { 2, "Третий элемент"},
  { 3, "Четвертый элемент"},
  { 4, "Пятый элемент"}
};
```

```
void OnOption1(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
.....
if (bFirst == TRUE)
{
strcpy(g_szEdit,DynWizEdit);
bFirst = FALSE;
}
...
...
```

//Статический текст для поля для выбора

```
CreateStatic(0,162,DynWizListStatic);
...
```

//Поле для выбора

```
hWnd = CreateListbox(0,177,"Headline",LB_NUM_LINES,&g_iListBox);
```

MoveWindow(hWnd,0,177,(rect.right-rect.left),50,TRUE);

//С помощью функции CreateListbox создается только поле. Содержимое строки необходимо ввести с помощью //функции SendMessage.

```
for (i=0; i<LB_NUM_LINES; i++)
{
SendMessage(hWnd,LB_INSERTSTRING,(WPARAM)-
1,(LPARAM)g_itemListBox[i].szItemText);
}
}
```

4.5.12.5 CreateCheckBox

Введение

В диалоговом окне Set options (Настройка параметров) отображается флажок для координат x, y. С помощью этого флажка можно включить параметр. В диалоговом окне можно использовать несколько флажков.

Синтаксис

HWND CreateCheckBox (int x, int y, char* Text, BOOL* pSelect)

Параметры

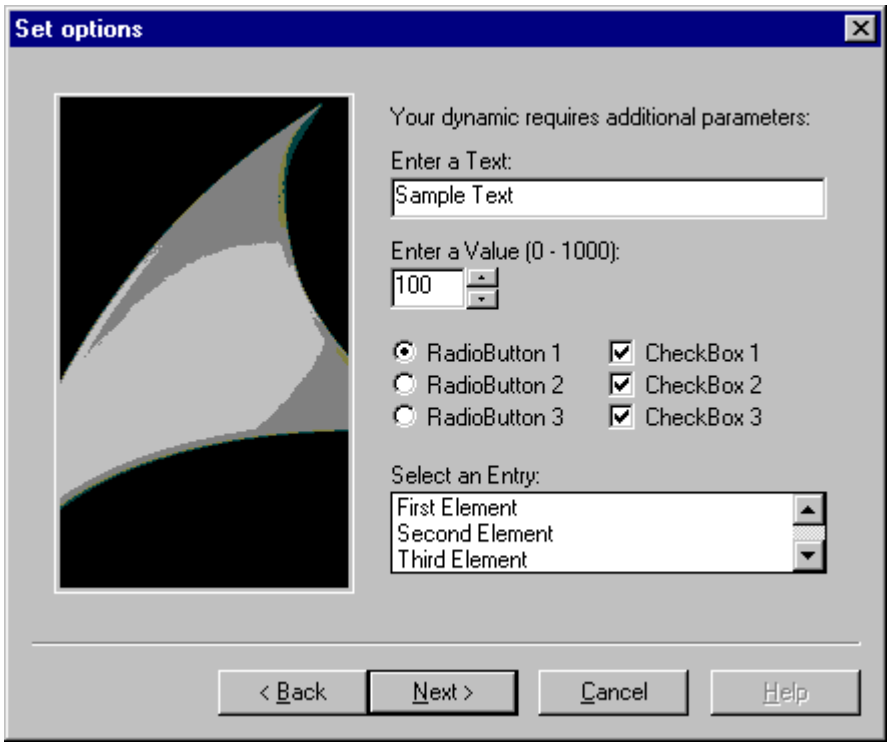
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
char* Text	Текст, отображаемый справа от флажка.
BOOL* pSelect	Указатель на переменную результата. Переменная результата должна являться предварительно заданным значением по умолчанию (True/False).

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
pSelect	Состояние включения FALSE = не включено TRUE = включено

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображаются три флажка, каждый из которых соответствует параметру. Каждый параметр можно включить независимо друг от друга.




```
BOOL g_bCheck1 = TRUE;
```

```
BOOL g_bCheck2 = TRUE;
BOOL g_bCheck3 = TRUE;

void OnOption1(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
.....
if (bFirst == TRUE)
{
...
}
...
...
//Флажок
iMid = (rect.right-rect.left)/2 ;

CreateCheckBox(iMid,100,"Флажок 1",&g_bCheck1);
CreateCheckBox(iMid,116,"Флажок 2",&g_bCheck2);
CreateCheckBox(iMid,132,"Флажок 3",&g_bCheck3
}
```

4.5.12.6 CreateFrame

Введение

В диалоговом окне Set options (Настройка параметров) отображается прямоугольная рамка. Левый верхний угол рамки определяется координатами x, y. Правый нижний угол рамки аналогичен правому нижнему углу окна параметров.

Синтаксис

HWND CreateFrame (int x, int y, char* Title)

Параметры

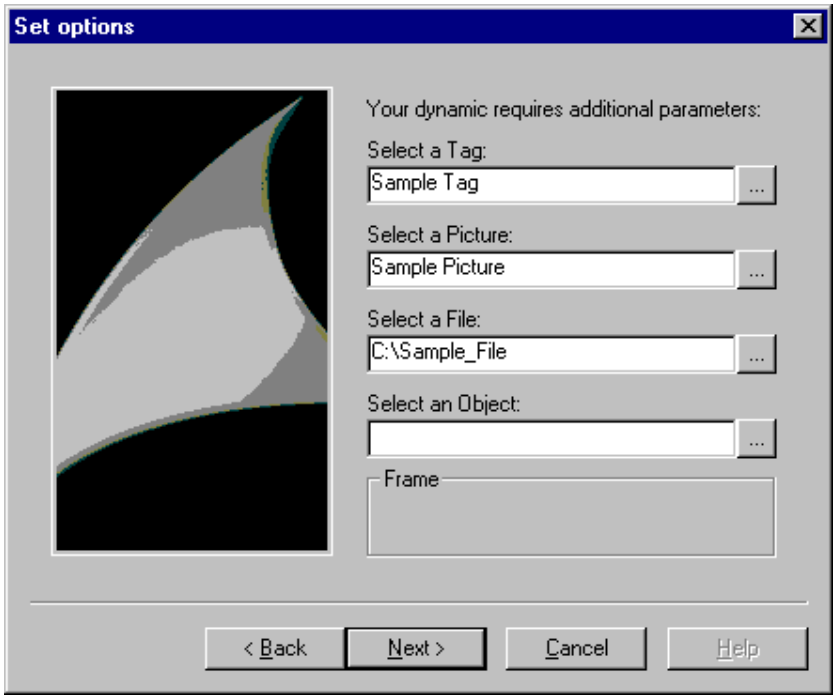
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
char* Title	Название в верхнем крае прямоугольника

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображается рамка с заголовком Frame (Рамка).




```
void OnOption2(void)
{
//Рамка
 CreateFrame(0,150,"Рамка");
}
...
...
```

4.5.12.7 CreateRadioButton

Введение

В диалоговом окне Set options (Настройка параметров) отображается переключатель для координат x, y. С помощью этого переключателя можно включить параметр.

Использование переключателей оправдывается, только если в диалоговом окне их несколько. Единовременно активен только один переключатель.

Синтаксис

HWND CreateRadioButton (int x, int y, char* Text, BOOL* pSelect)

Параметры

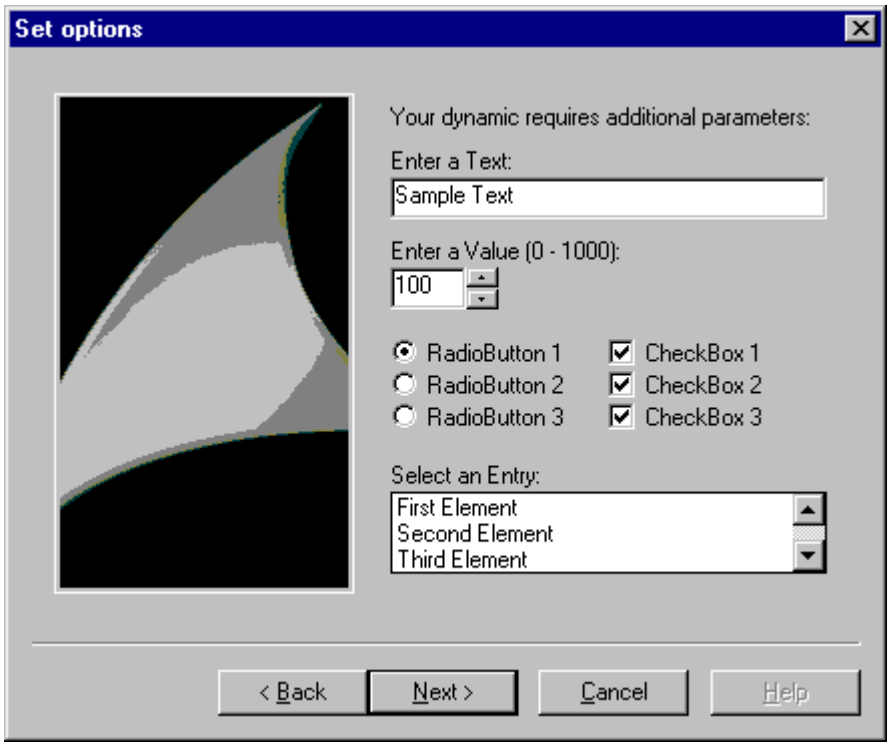
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
char* Text	Название параметра, активируемого переключателем. Текст отображается справа от переключателя.
BOOL* pSelect	Указатель на переменную результата. Переменная результата должна являться предварительно заданным значением по умолчанию (True/False).

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
pSelect	Состояние включения: FALSE = не включено TRUE = включено

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображаются три переключателя, каждый из которых соответствует параметру. Можно включить только один параметр.




```
BOOL g_bOption1 = TRUE;
```

```
BOOL g_bOption2 = FALSE;
BOOL g_bOption3 = FALSE;

void OnOption1(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
.....
if (bFirst == TRUE)
{
...
}
...
...
```

//Переключатели

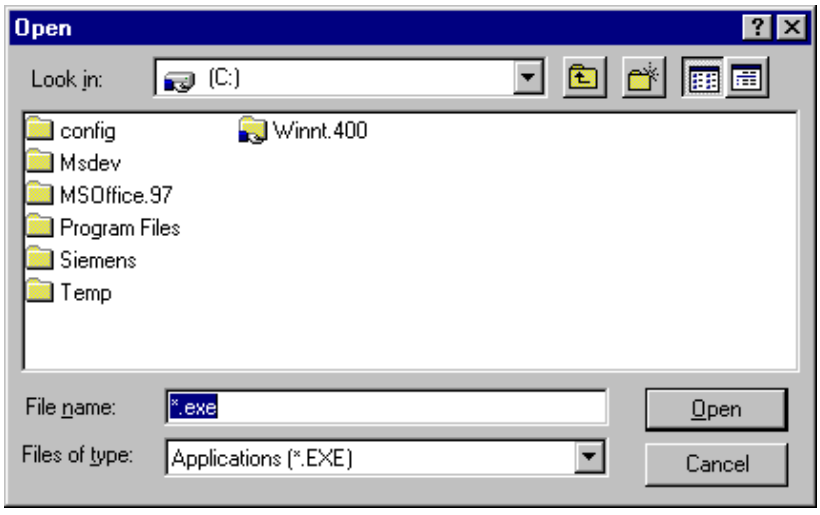
```
CreateRadioButton(0,100,"Переключатель 1",&g_bOption1);
CreateRadioButton(0,116,"Переключатель 2",&g_bOption2);
CreateRadioButton(0,132,"Переключатель 3",&g_bOption3);
}
```

4.5.12.8 CreateFileBrowser

Введение

В диалоговом окне Set options (Настройка параметров) отображается поле ввода с кнопкой Browse (Обзор) для координат x, y. Имя файла можно ввести в это поле ввода.

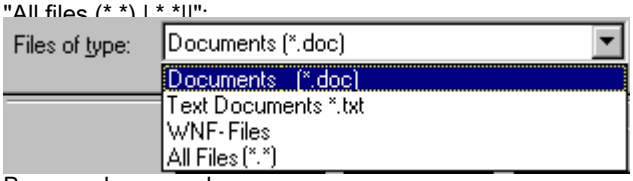
При нажатии кнопки Browse (Обзор) откроется диалоговое окно выбора файла.



Синтаксис

HWND CreateFileBrowser (int x, int y, DWORD Flags, char* Filter, char* Dateiname)

Параметры

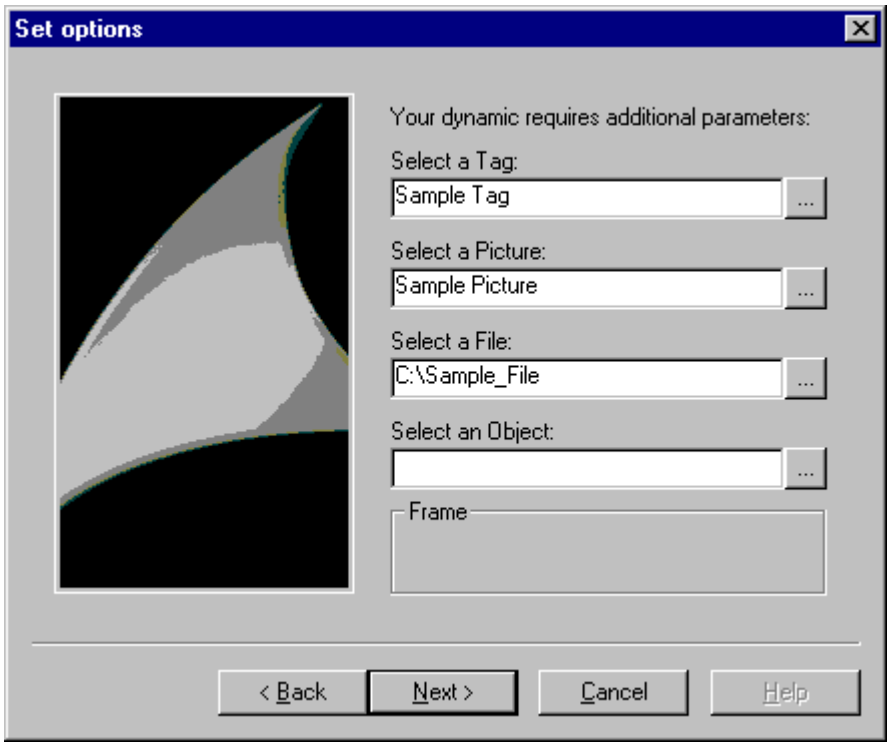
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
DWORD Flags	Управляющий флаг окна выбора: FB_WITHPATH = имя файла и путь FB_SAVE_AS = вместо диалогового окна Open (Открыть) отображается диалоговое окно Save as (Сохранить как).
char* Filter	Фильтр для отображения типа данных в поле для выбора диалогового окна выбора файла. Указанное расширение определяет отображаемые типы данных в поле для выбора. Фильтр состоит из пары строк. Первая строка — имя фильтра. Вторая строка — функция фильтра в формате *.typ, где "typ" обозначает расширение файла. В поле для выбора отображаются только файлы с таким расширением. 1. и вторая строка разделены |. Несколько фильтров можно разделить по строкам с помощью |. Последний фильтр ограничен ||. Примеры: char* Filter1 = "Graphic pictures (*.PDL) | *.PDL||";  char* Filter2 = "Documents (*.doc) | *.doc|" "Text files *.txt | *.txt|" "WNF files | *.wnf|" "All files (*.*) | *.*||";  В конце функции фильтра не допускается использовать пробелы.
char* File name	Буфер ввода для имени файла. Имя пути можно определить в качестве значения по умолчанию. Это стандартное значение дает следующий эффект. Имя пути отображается в окне ввода по умолчанию. При нажатии кнопки Browse (Обзор) устанавливается путь в диалоговом окне выбора файла. Если файл имеет расширение "*.typ", все файлы этого типа отобразятся в поле выбора в диалоговом окне выбора.

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
File name	Буфер ввода содержит имя файла.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображается поле ввода с кнопкой Browse (Обзор). При нажатии кнопки Browse (Обзор) откроется диалоговое окно выбора файла.




```
char* DynWizFileBrowserStatic = "Выберите файл:";
char* DynWizFileBrowser = "C:\\пример_файла";
char* DynWizFilter = "Текстовые файлы (*.txt) | *.txt|"
  "Все файлы (*.*) | *.*||";
...
char g_szFileBrowser[256];
...
void OnOption2(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
RECT rect;
...
if (bFirst == TRUE)
{
...
strcpy(g_szFileBrowser,DynWizFileBrowser);
First = FALSE;
}
...
...
```

```
//Статический текст для поля ввода с кнопкой Browse (Обзор)
CreateStatic(0,95,DynWizFileBrowserStatic);
//Диалоговое окно выбора файла
hWnd =
CreateFileBrowser(0,110,FB_WITHPATH,DynWizFilter,g_szFileBrowser);
MoveWindow(hWnd,0,110,(rect.right-rect.left),21,TRUE);
}
```

4.5.12.9 CreateVarBrowser/CreateVarBrowserEx

Введение

В диалоговом окне Set options (Настройка параметров) отображается поле ввода с кнопкой Browse (Обзор) для координат x, y. Имя тега можно ввести в это поле ввода. При нажатии кнопки Browse (Обзор) откроется диалоговое окно выбора тега WinCC. Функция CreateVarBrowserEx позволяет настроить дополнительные параметры фильтра тега. Этот фильтр ограничивает отображение тегов в диалоговом окне выбора тегов. Фильтрацию можно выполнить по типу данных, группе тегов, имени тега и соединению.

Синтаксис

HWND CreateVarBrowser (int x, int y, char* VarName)

HWND CreateVarBrowserEx (int x, int y, LPDM_VARFILTER VarFilter, char* VarName)

Параметры

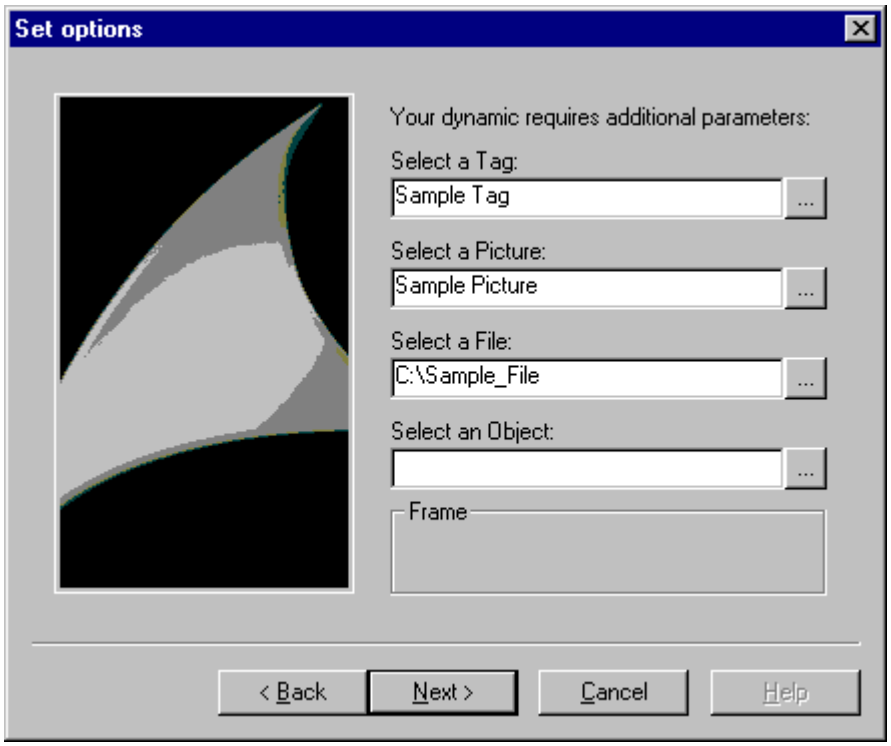
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
LPDM_VARFILTER VarFilter	Дополнительное обозначение указателя на фильтр тега. Если задан указатель ZERO, фильтр не активен. Фильтр тега должен определяться с помощью структуры DM_VARFILTER. Дополнительную информацию по этой теме см. в документации WinCC ODK.
char* VarName	Содержит имя тега. Имя тега может иметь предварительно заданное значение по умолчанию. Эта запись всегда отображается.

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
VarName	Буфер ввода содержит имя тега.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображается поле ввода с кнопкой Browse (Обзор). При нажатии кнопки Browse (Обзор) откроется диалоговое окно выбора тега WinCC.




```
char* DynWizVarBrowser = "Пример тега";
char* DynWizPicBrowserStatic = "Выберите кадр:";
...
char g_szVarBrowser[256];
...
void OnOption2(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
RECT rect;
...
if (bFirst == TRUE)
{
...
strcpy(g_szVarBrowser,DynWizVarBrowser);
First = FALSE;
}
...
...
```

```
//Статический текст для поля ввода с кнопкой Browse (Обзор)
CreateStatic(0,95,DynWizFileBrowserStatic);
//Диалоговое окно выбора тега
hWnd =
CreateFileBrowser(0,110,FB_WITHPATH,DynWizFilter,g_szFileBrowser);
GetWindowRect(GetParent(hWnd), &rect);
MoveWindow(hWnd,0,110,(rect.right-rect.left),21,TRUE);
}
```

4.5.12.10 CreatePackageBrowser/CreatePackageBrowserEx

Введение

В диалоговом окне Set options (Настройка параметров) отображается поле ввода с кнопкой Browse (Обзор) для координат x, y. Имя можно ввести в это поле ввода. Обозреватель пакетов открывается при нажатии кнопки Browse (Обзор) справа от поля ввода. Флаг или ProgID определяют тип данных, отображаемых из пакета.

С помощью функции CreatePackageBrowserEx можно передать ProgID вместо флага.

Синтаксис

HWND CreatePackageBrowser (int x, int y, DWORD flags, char* Name)

HWND CreatePackageBrowserEx (int x, int y, char* ProgID, char* Name)

Параметры

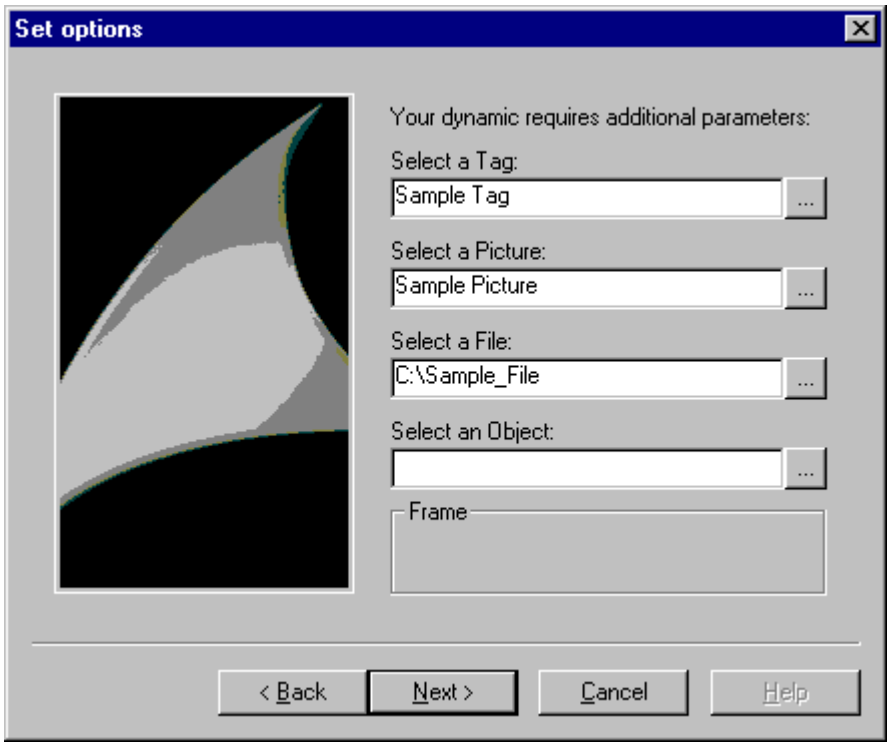
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
DWORD flags	В настоящее время можно использовать только PB_PICTURE. Это обеспечивает выбор кадра.
char* ProgID	Программный идентификатор компонента, используемого для создания выбора. При передаче WinCC.CCFileASOStub.1 осуществляется обращение к выбору кадра.
char* Name	Содержит имя. Имя может иметь предварительно заданное значение по умолчанию. Эта запись всегда отображается.

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
Name	Буфер ввода содержит имя.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображается поле ввода с кнопкой Browse (Обзор). При нажатии кнопки Browse (Обзор) откроется диалоговое окно выбора кадра.




```
char* DynWizPicBrowserStatic = "Выберите кадр:";
char* DynWizPicBrowser = "Пример кадра";

...
char g_szPicBrowser[256];
...
void OnOption2(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
RECT rect;
...
if (bFirst == TRUE)
{
...
&#9;strcpy(g_szPicBrowser,DynWizPicBrowser);

First = FALSE;
}
...
...
```

```
//Статический текст для поля ввода с кнопкой Browse (Обзор)
CreateStatic(0,50,DynWizPicBrowserStatic);
//Диалоговое окно выбора кадра
hWnd = CreatePackageBrowser(0,65,PB_PICTURE,g_szPicBrowser);
MoveWindow(hWnd,0,65,(rect.right-rect.left),21,TRUE);
}
```

4.5.12.11 CreateObjectBrowser

Введение

В диалоговом окне Set options (Настройка параметров) отображается поле ввода с кнопкой Browse (Обзор) для координат x, y. Имя объекта или свойства можно ввести в это поле ввода. При нажатии кнопки Browse (Обзор) откроется диалоговое окно выбора. В этом диалоговом окне выбора можно выполнить поиск имени объекта либо свойства или выбрать его.

Синтаксис

HWND CreateObjectBrowser (int x, int y, char* Title, DWORD flags, char* ObjectName)

Параметры

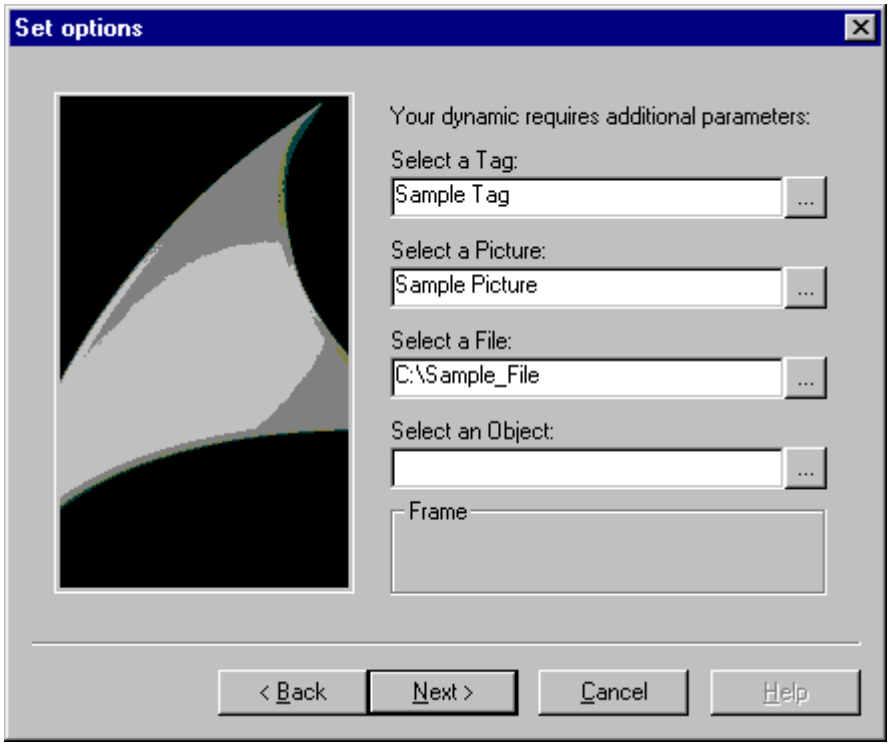
Параметры	Описание
int x	Отображает значение координаты X.
int y	Отображает значение координаты Y.
char* title	Название диалогового окна выбора.
DWORD flags	Можно передать два разных флага: OB_OBJECTS — отображение всех объектов. OB_PROPERTIES — дополнительная возможность выбора свойства.
char* ObjectName	Буфер ввода для имени объекта или свойства. Для буфера ввода можно установить значение по умолчанию.

Возвращаемое значение

	Возвращаемое значение
HWND	Возвращает манипулятор объекта.
ObjectName	Буфер ввода для имени объекта или свойства.

Пример

В приведенной ниже выдержке из файла Demo.wnf указано использование этой функции. В диалоговом окне Set options (Настройка параметров) модуля Demo Wizard (Мастер демонстрации) отображается поле ввода с кнопкой Browse (Обзор). При нажатии кнопки Browse (Обзор) откроется диалоговое окно выбора объекта.




```
char* DynWizObjectBrowserStatic = "Выберите объект:";
char* DynWizObjectBrowser = "Объект";
char* DynWizObject = "Выбор объекта окна";
;
...
char g_szObjectBrowser[256];
...
void OnOption2(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = ZERO;
RECT rect;
...
if (bFirst == TRUE)
{
...
strcpy(g_szObjectBrowser,DynWizObjectBrowser);
First = FALSE;
}
...
...
```

//Статический текст для поля ввода с кнопкой Browse (Обзор)

```
CreateStatic(0,50,&#9;CreateStatic(0,140,DynWizObjectBrowserStatic);
);
```

//Диалоговое окно выбора окна

```
hWnd =
CreateObjectBrowser(0,155,DynWizObject,OB_OBJECTS,g_szObjectBrowser)
;
MoveWindow(hWnd,0,155,(rect.right-rect.left),21,TRUE);
}
```

4.5.13 Функции мастера для создания динамики

4.5.13.1 GenerateBLOB

Введение

Функция GenerateBLOB (BLOB — большой двоичный объект) создает макрос, который можно добавить к свойству графического объекта. Макрос состоит из 3 частей.

Вступление. Это заголовок функции C.

Пример.
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName,char* lpszObjectName,char*
lpszPropertyName, UINT nFlags, int x, int y)
{
Вступление зависит от триггера, запускающего макрос (в вышеприведенном примере: нажатие левой кнопки мыши).

Эпилог. Это завершение функции C, которое обозначается символом "}".

Центральная часть. Эта часть содержит фактические возможности функции C.
Пример. ProgramExecute("notepad.exe");

Эта функция создает и компилирует код C макроса. В результате компиляции генерируется код P. Этот код интерпретируется и обрабатывается системой среды исполнения WinCC. В случае неправильного кода C код P не создается.

Функция создает BLOB, в котором хранятся части макроса (код C, код P, триггер...). Перед завершением функции мастера необходимо снова удалить BLOB. Дополнительную информацию по удалению функции BLOB см. в разделе «DeleteBLOB».

Синтаксис

AP_BLOB GenerateBLOB (char* Prolog, char* Epilog, char* Format, ...)

Параметры

Параметры	Описание
char* Prologue	Вступление макроса в качестве строки ASCII.
char* Epilogue	Эпилог макроса в качестве строки ASCII.
char* Format	Центральная часть макроса в качестве строки ASCII или строки формата в соответствии со стандартной функцией printf.

Примечание

Коды C создаются посредством функции C sprintf. Параметр обрабатывается в качестве строки формата, т. е. оцениваются управляющие символы формата (например, \ % "). Если эти символы необходимо передать в код C (например, в качестве троки формата для вызова printf в макросе), их необходимо указать вместе с символом "\".

Пример.

\ → \\

% → \%

" → \"

Возвращаемое значение

Функция возвращает структурный тег типа AP_BLOB со следующими структурными компонентами.

Структурный компонент	Возвращаемое значение
DWORD dwPCodeSize	Длина созданного кода P в байтах
LPVOID lpPCode	Указатель на созданный код P
int nErrors	Число ошибок компилятора
int nWarnings	Число предупреждений компилятора

Пример

В приведенной ниже выдержке из файла Execute Programm.wnf указано использование этой функции. Функция мастера создает сценарий C, который запускает другое приложение (в данном примере: notepad.exe).

```
...
...
void OnGenerate(void)
{
PCMN_ERROR pError;
AP_BLOB *blob;
char code[500];
char sError[500];
..
Slash2DblSlash(g_Picture, strlen(g_Picture));
..
sprintf(code,"%sProgramExecute(\"%s\");",ifcode,g_Picture);
..
//Вступление
blob = GenerateBLOB("#include \"apdefap.h\"\r\n"
```

```
"void OnClick(char* lpszPictureName," "char*lpszObjectName,char*
lpszPropertyName,"
"UINT nFlags,int x, int y) {",
//Эпилог
"}",
//Центральная часть
code);

BEGIN_JCR_BLOBERRORS

SetAction(ZERO, blob, g_Trigger);

END_JCR_BLOBERRORS

DeleteBLOB(blob);
}
```

Созданный сценарий C

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName,
char* lpszObjectName,
char* lpszPropertyName,
UINT nFlags, int x, int y)
{
ProgramExecute("notepad.exe");
}
```

4.5.13.2 DeleteBLOB

Введение

Функция GenerateBLOB создает BLOB. По завершении функции мастера необходимо снова удалить BLOB. Удаление BLOB осуществляется с помощью функции DeleteBLOB.

Синтаксис

void DeleteBLOB (AP_BLOB* blob)

Параметры

Параметры	Описание
AP_BLOB* blob	Указатель на переменную результата функции GenerateBLOB.

Пример

DeleteBLOB(blob);

4.5.13.3 SetAction

Введение

Макрос добавляется к выбранному графическому объекту с указанным триггером.

Если триггером является событие, оно указывается напрямую в качестве параметра вызова.

Если триггер представляет собой динамизацию свойства, его необходимо предварительно ввести в BLOB с помощью функций AddVarTrigger или AddTimeTrigger.

Примечание

Если макрос требуется добавить не к выбранному объекту, а к другому, необходимо использовать функцию API PDLCSSetAction. Дополнительную информацию по функции PDLCSSet Action см. в руководстве WinCC ODK.

Синтаксис

BOOL SetAction (char* Property, AP_BLOB* Blob, DWORD Trigger)

Параметры

Параметры	Описание
char* Property	Имя свойства. Всегда используйте английское имя свойства. Для триггера по событию необходимо передать указатель ZERO.
AP_BLOB* Blob	Указатель на переменную результата функции GenerateBLOB.
DWORD TriggerID	Идентификатор триггера: NOTDEFINED = триггер введен в BLOB MOUSECLICK = щелчок мышью MOUSELBUTTONDOWN = нажатие левой кнопки мыши MOUSELBUTTONUP = отпускание левой кнопки мыши MOUSERBUTTONDOWN = нажатие правой кнопки мыши MOUSERBUTTONUP = отпускание правой кнопки мыши KEYBOARDDOWN = нажатие клавиши KEYBOARDUP = отпускание клавиши OBJECTCHANGE = изменение объекта PROPERTYCHANGE = изменение свойства PICTUREOPEN = выбор кадра PICTURECLOSE = закрытие кадра

Возвращаемое значение

	Возвращаемое значение
Возвращаемое значение BOOL	TRUE = функция выполнена успешно. FALSE = функция не выполнена успешно.

Дополнительные источники информации

GenerateBLOB (стр. 67)

4.5.13.4 AddTimeTrigger

Введение

Функция дополняет макрос триггером типа «цикличный триггер».

Синтаксис

BOOL AddTimeTrigger (AP_BLOB* Blob, char* Name, DWORD TriggerType, DWORD GraphCycleType, DWORD CycleID)

Параметры

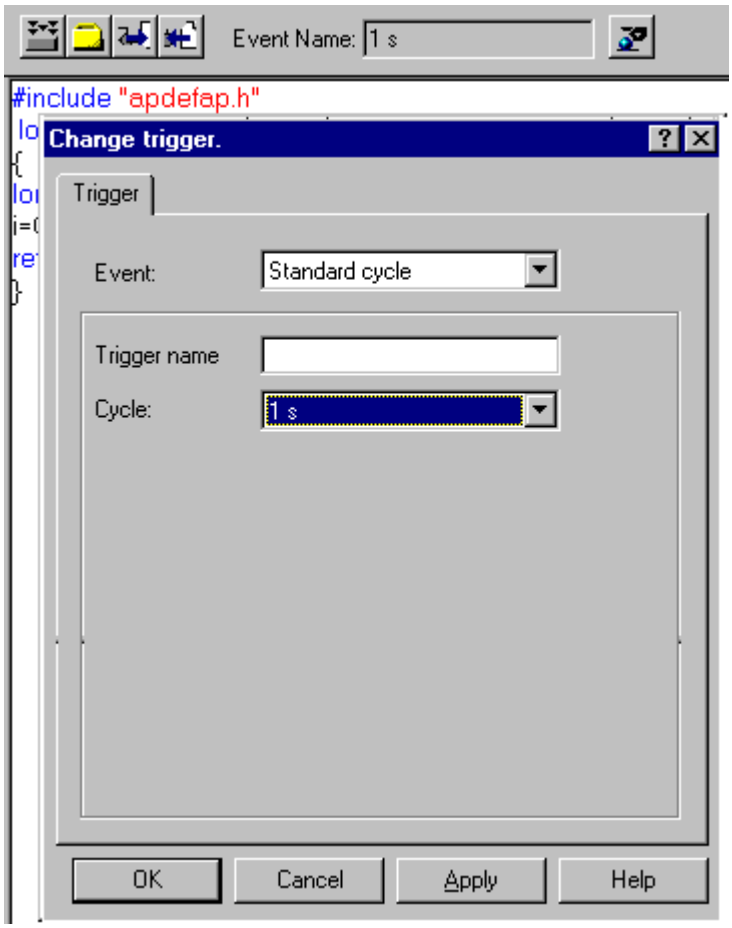
Параметры	Описание
AP_BLOB* Blob	Указатель на переменную результата функции GenerateBLOB.
char* Name	Имя события. Это может быть любая строка ASCII. Имя отображается в виде имени события в окне макроса.
DWORD TriggerType	Тип цикличного триггера: 2 = временной цикл (стандартный цикл) 4 = цикл графического объекта
DWORD GraphCycleType	Тип цикла графического объекта: 2 = цикл окна 1 = цикл кадра
DWORD CycleID	Цикл триггера: 0 = при изменении 1 = 250 мс 2 = 500 мс 3 = 1 с 4 = 2 с 5 = 5 с 6 = 10 с 7 = 1 мин 8 = 5 мин 9 = 10 мин 10 = 1 ч 11 = пользовательский цикл 1 12 = пользовательский цикл 2 13 = пользовательский цикл 3 14 = пользовательский цикл 4 15 = пользовательский цикл 5

Возвращаемое значение

	Возвращаемое значение
BOOL	TRUE = функция выполнена успешно. FALSE = функция не выполнена успешно.

Пример

Время между срабатываниями двух макросов составляет 1 с.




```
BOOL FctRet;
..
FctRet = AddTimeTrigger(blob, "1 sec", 2, 0, 3);
```

4.5.13.5 AddVarTrigger/AddVarTriggerEx

Введение

Функция дополняет макрос триггером типа «триггер тега».

Синтаксис

BOOL AddVarTrigger (AP_BLOB* Blob, char* EventName, char* VarName)

BOOL AddVarTriggerEx (AP_BLOB* Blob, char* EventName, char* VarName, DWORD CycleID)

Параметры

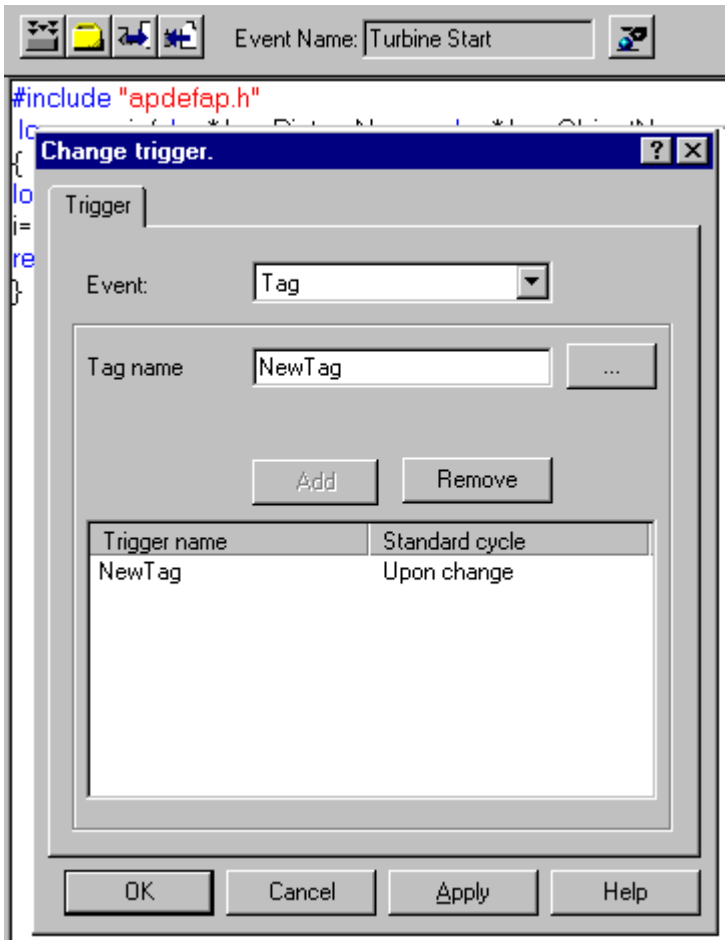
Параметры	Описание
AP_BLOB* Blob	Указатель на переменную результата функции GenerateBLOB.
char* EventName	Имя события. Это может быть любая строка ASCII. Имя отображается в виде имени события в окне макроса.
char* VarName	Имя тега WinCC, инициирующего срабатывание или участвующего в нем.
DWORD CycleID	Цикл триггера: 0 = при изменении 1 = 250 мс 2 = 500 мс 3 = 1 с 4 = 2 с 5 = 5 с 6 = 10 с 7 = 1 мин 8 = 5 мин 9 = 10 мин 10 = 1 ч 11 = пользовательский цикл 1 12 = пользовательский цикл 2 13 = пользовательский цикл 3 14 = пользовательский цикл 4 15 = пользовательский цикл 5 Для функции AddVarTrigger предварительно задано значение CycleID = 4.

Возвращаемое значение

	Возвращаемое значение
BOOL	TRUE = функция выполнена успешно. FALSE = функция не выполнена успешно.

Пример

В качестве триггера введен тег StartTurbine1 в типе триггера Tag (Ter). Макрос запускается при изменении значения одного из этих тегов.




```
BOOL FctRet

FctRet = AddVarTriggerEx(blob,"Turbine Start","StartTurbine1",0);
```

4.5.13.6 SetValidateFct

Введение

Имя функции проверки передается в Dynamic Wizard (Мастер динамики). Функция проверки позволяет проверять настройки и параметры триггера. В случае отрицательного результата проверки можно инициировать повторный ввод.

Функция проверки вызывается при нажатии кнопки Continue (Продолжить) в диалоговых окнах Select options (Выбор параметров) или Set trigger (Настройка триггера). В случае положительного результата проверки диалоговое окно закроется и отобразится следующая страница. В случае отрицательного результата проверки диалоговое окно остается открытым. Продолжение возможно только после ввода правильных параметров.

Функция проверки выполняется сразу после ее настройки в Dynamic Wizard (Мастер динамики). Она также действует для последующих страниц параметров. Если не требуется применять функцию проверку или необходимо выполнить другую функцию проверки, необходимо установить пустую функцию (с положительным результатом проверки) либо другую функцию проверки.

Синтаксис

BOOL SetValidateFct (LPCSTR FctName)

Параметры

Параметры	Описание
LPCSTR FctName	Имя функции проверки в качестве строки ASCII.

Возвращаемое значение

	Возвращаемое значение
BOOL	Результат проверки TRUE = положительный результат проверки. FALSE = отрицательный результат проверки.

Пример

В приведенной ниже выдержке из файла Instanzobjekt.wnf указано использование этой функции.

Функция мастера дополнена функцией проверки.

```
...
...
// Параметр проверки 1
BOOL ValidateOpt1(void)
{
// Свойство выбрано
return (strcmp(g_NewInst, ""));
}

void OnOption1(void)
{
HWND hWnd;
RECT rect;
DM_VARFILTERdmFilter = {DM_VARFILTER_TYPE, 1, ZERO, ZERO, ZERO, ZERO
};

SetValidateFct("ValidateOpt1");
sprintf(g_NewInst, "");
..
}
```

4.5.13.7 EnumProperty/EnumPropertyEx

Введение

Функция EnumProperty перечисляет свойства объекта. Функция EnumPropertyEx позволяет указать свойства объекта, которые необходимо перечислить.

Синтаксис

BOOL EnumProperty (char* FName, LPVOID pItem, DWORD dwFlags);

BOOL EnumPropertyEx (LPCTSTR Projectname, LPCTSTR Picturename, LPCTSTR Objectname, char* FName, LPVOID pItem, DWORd dwFlags);

Параметры

Параметры	Описание
LPCTSTR Projectname	Указатель на имя проекта, в том числе каталог и расширение файла.
LPCTSTR Picturename	Указатель на имя кадра, объекты которого необходимо перечислить. Регистр клавиатуры учитывается.
LPCTSTR object name	Указатель на имя объекта
char* FName	Имя функции обратного вызова, которая вызывается один раз для каждого свойства объекта.
LPVOID pItem	Указатель на данные приложения, переданные в функцию обратного вызова. Этот указатель не оценивается функцией, но он снова становится доступен в функции обратного вызова.
DWORD dwFlags	dwFlags указывает типы свойства, которые необходимо перечислить. В настоящее время возможны следующие спецификации: **PropertyHasDynamic** (значение: 0x0001) перечисляются только свойства объектов с динамикой. **PropertyHasEvents** (значение: 0x0002) перечисляются только свойства объектов с событиями. **PropertyIsDynamicable** (значение: 0x0003) перечисляются только свойства объектов, которые можно динамизировать.

Возвращаемое значение

	Возвращаемое значение
BOOL	TRUE = перечисляются свойства типа объекта FALSE = ошибка

Пример

В приведенной ниже выдержке из файла Dynamic Property.wnf указано использование этой функции.

```
...
...
// Функция обратного вызова
BOOL EnumFct(char *property, VARTYPE vt, LPVOID pItem)
{
sprintf(g_prop[SendMessage((HWND)pItem,LB_INSERTSTRING,(WPARAM)-
1,(LPARAM)property)],property);
return TRUE;
}

void OnOption1(void)
{
HWND hWnd,LBHwnd;
RECT rect;
static BOOL bFirst = TRUE;

if(bFirst)
{
...
}
...

CreateStatic(0, 10,"Свойства текущего объекта:");
LBHwnd=CreateListbox(0, 30, g_Headline, 8, &g_indexProperty);
EnumProperty("EnumFct", LBHwnd, 3);
GetWindowRect(GetParent(LBHwnd), &rect);
...
```

4.5.14 Функции WinCC для мастера

4.5.14.1 GetProjectName

Введение

Определяется путь к текущему проекту WinCC.

Синтаксис

LPCSTR GetProjectName (void)

Возвращаемое значение

	Возвращаемое значение
LPCSTR	Указатель на строку ASCII в файле MCP

Пример

LPCSTR Name;

Name = GetProjectName();

К примеру, функция выдает следующий результат:
C:\Siemens\WinCC\WinCCProjects\Example.mcp

4.5.14.2 GetPictureName

Описание

Определяется имя текущего кадра (*.pdl).

Синтаксис

LPCSTR GetPictureName (void)

Возвращаемое значение

	Возвращаемое значение
LPCSTR	Указатель на строку ASCII в файле PDL

Пример

LPCSTR Name;

Name = GetPictureName();

К примеру, функция выдает следующий результат: TurbineControl.PDL

4.5.14.3 GetDefaultWNFPath

Описание

Определяется путь к текущему каталогу WNF.

Синтаксис

LPCSTR GetDefaultWNFPath (void)

Возвращаемое значение

	Возвращаемое значение
LPCSTR	Указатель на строку ASCII в имени пути

Пример

LPCSTR Name;

Name = GetDefaultWNFPath();

К примеру, функция выдает следующий результат:
C:\Siemens\WinCC\wscripts\wscripts.deu\

4.5.14.4 GetObjectName

Введение

Определяется имя выбранного графического объекта в текущем кадре.

Синтаксис

LPCSTR GetObjectName (void)

Возвращаемое значение

	Возвращаемое значение
LPCSTR	Указатель на строку ASCII в имени объекта

Пример

LPCSTR Name;

Name = GetObjectName();

К примеру, функция выдает следующий результат: Button1

4.5.14.5 InsertXRefSection

Описание

Функция вставляет в передаваемый исходный код раздел в соответствии с записью Xref таким образом, что передаваемые теги имена кадров вводятся согласно определению.

Синтаксис

BOOL InsertXRefSection (char * SourceCode, char* TagName[], int TagCount, char* PictName[], int PictCount)

Параметры

Параметры	Описание
char *SourceCode	Буфер кода (CodeBuffer), в которой вставляется раздел Xref
char *TagName[]	ZERO или поле имен тегов, вставляемых в раздел Xref.
int TagCount	Число имен тегов в поле DayName[]
char *PictName[]	ZERO или поле имен кадров, вставляемых в раздел Xref.
int PictCount	Число имен кадров в поле PictName[]

Возвращаемое значение

	Возвращаемое значение
BOOL	Значение результата указывает успешность выполнения функции.
char *TagName[]	Поле определяет теги, передаваемые в то же положение
char *PictName[]	Поле определяет кадры, передаваемые в то же положение

Beispiel

```
char* szPictureArray[1];
char szPictName[255];
char szSourceCode[1100];

strcpy(szPictName, "Newpdl.pdl");
szPictureArray[0] = szPictName;
strcpy(szSourceCode, "");
InsertXrefSection(szSourceCode,NULL,0,szPictureArray,1);
```

К примеру, функция возвращает следующий результат:

szSourceCode:

// WINCC:TAGNAME_SECTION_START

// syntax: #define TagNameInAction "DMTagName"

// next TagID : 1

// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START

// syntax: #define PicNameInAction "PictureName"

// next PicID : 1

#define PIC_0 " Newpdl.Pdl"

// WINCC:PICNAME_SECTION_END

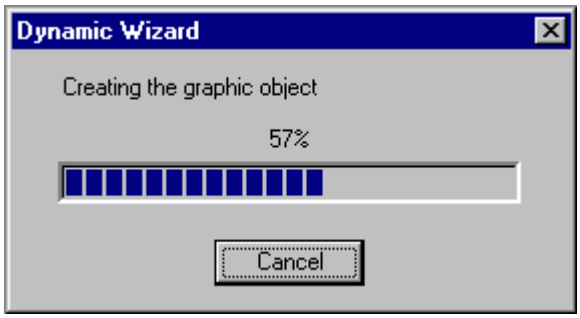
szPictureArray[0]: "PIC_0"

4.5.15 Функции хода выполнения для мастера

4.5.15.1 Функции хода выполнения для мастера

Введение

Функции хода выполнения служат для отображения хода выполнения (в %) процедуры в окне хода выполнения.



При создании индикатора выполнения (CreateProgressDlg) указываются начальное и конечное значения и приращение. Начальное значение соответствует ходу выполнения 0%, а конечное значение — 100%. Приращение определяет этапы изменения хода выполнения.

Как правило, начальное значение равно «0», а приращение — «1». Конечное значение соответствует числу выполняемых действий в процессе.

Во время процедуры ход выполнения осуществляется приращениями (Progress_StepIt) или для него устанавливается заданное значение (Progress_SetPos).

По завершении процедуры отображение хода выполнения должно быть снова удалено (DestroyProgressDlg)

В индикаторе выполнения (Progress_SetStatus) можно отобразить текст, например Creating graphic objects (Создание графических объектов). Этот текст можно также изменить во время обработки для разграничения различных частичных процедур.

В большинстве случаев процедуру невозможно разделить таким образом, чтобы обеспечить хронологические линейные отображение хода выполнения. Тем не менее, в действительности это является необходимостью. Отображение хода выполнения как такового достаточно.

Дополнительные источники информации

DestroyProgressDlg (стр. 84)

Progress_SetPos (стр. 83)

Progress_StepIt (стр. 83)

Progress_SetStatus (стр. 83)

CreateProgressDlg (стр. 82)

4.5.15.2 CreateProgressDlg

Введение

Индикатор выполнения отображает ход выполнения процедуры обработки от 0 до 100%.

Синтаксис

PROGRESS_DLG CreateProgessDlg (int nLower, int nUpper, int nStepInc)

Параметры

Параметры	Описание
int nLower	Начальное значение хода выполнения (соответствует 0%)
int nUpper	Конечное значение хода выполнения (соответствует 100%)
int nStepInc	Приращение хода выполнения

Возвращаемое значение

	°Возвращаемое значение
PROGRESS_DLG	Манипулятор объекта

4.5.15.3 Progress_SetStatus

Описание

Текст вводится в качестве заголовка в индикатор выполнения.

Синтаксис

void Progress_SetStatus (PROGRESS_DLG hDlg, char* ActionName)

Параметры

Параметры	Описание
PROGRESS_DLG hDlg	Манипулятор объекта
char* ActionName	Текст заголовка

4.5.15.4 Progress_StepIt

Описание

Ход выполнения процедуры обработки приращивается по одному этапу.

Синтаксис

void Progress_StepIt (PROGRESS_DLG hDlg)

Параметры

Параметры	Описание
PROGRESS_DLG hDlg	Манипулятор объекта

4.5.15.5 Progress_SetPos

Описание

В индикаторе выполнения для хода выполнения установлено заданное значение. Значение должно находиться между начальным и конечным значениями.

Синтаксис

void Progress_SetPos (PROGRESS_DLG hDlg, int nPos)

Параметры

Параметры	Описание
PROGRESS_DLG hDlg	Манипулятор объекта
int nPos	Значение хода выполнения

4.5.15.6 DestroyProgressDlg

Введение

Индикатор выполнения будет закрыт.

Синтаксис

void DestroyProgressDlg (PROGRESS_DLG hDlg)

Параметры

Параметры	Описание
PROGRESS_DLG hDlg	Манипулятор объекта

4.5.16 Функции окон для мастера

4.5.16.1 Функции окон для мастера

Введение

Ниже приведено краткое описание функций Windows, которые должны или могут использоваться в связи с системными функциями мастера (особенно с функциями окна для ввода параметров).

Более подробную информацию см. в справочном руководстве для программистов Microsoft Developers Studio/Win32 SDK.

Дополнительные источники информации

MessageBox (стр. 88)

ShowWindow (стр. 88)

GetWindow (стр. 87)

SendMessage (стр. 87)

MoveWindow (стр. 86)

GetWindowRect (стр. 85)

GetParent (стр. 85)

4.5.16.2 GetParent

Введение

Манипулятор родительского окна определяется для окна, например манипулятор окна параметров.

Синтаксис

HWND GetParent (HWND hWnd)

Параметры

Параметры	Описание
HWND hWnd	Манипулятор окна, для которого необходимо определить родительское окно

Возвращаемое значение

	Возвращаемое значение
HWND	Манипулятор родительского окна ZERO = родительное окно отсутствует.

4.5.16.3 GetWindowRect

Введение

Определяются размер и координаты окна, например размер окна параметров.

Синтаксис

BOOL GetWindowRect (HWND hWnd, LPRECT lpRect)

Параметры

Параметры	Описание
HWND hWnd	Манипулятор окна
LPRECT lpRect	Указатель на структурированную переменную результата

Возвращаемое значение

	Возвращаемое значение
BOOL	TRUE = функция выполнена успешно. FALSE = функция не выполнена успешно.
LPRECT lpRect	Структурированная переменная результата структуры LPRECT со структурнымии компонентами. LONG left: координата X левого верхнего угла LONG top: координата Y левого верхнего угла LONG right: координата X правого верхнего угла LONG bottom: координата Y правого нижнего угла.

Дополнительные источники информации

Добавление сценария Motor.wnf в базу данных (стр. 94)

CreateEdit (стр. 47)

4.5.16.4 MoveWindow

Введение

Положение и размер окна изменяются, например положение и размер полей ввода в окне параметров.

Синтаксис

BOOL MoveWindow (HWND hWnd, int x, int y, int nWidth, int nHeight, BOOL bRepaint)

Параметры

Параметры	Описание
HWND hWnd	Манипулятор окна
int x	Координата X левого верхнего угла
int y,	Координата Y левого верхнего угла
int nWidth	Ширина
int nHeight	Высота
BOOL bRepaint	TRUE = окно перерисовывается.

Возвращаемое значение

	Возвращаемое значение
BOOL	TRUE = функция выполнена успешно. FALSE = функция не выполнена успешно.

Дополнительные источники информации

Создание функции мастера динамики для объекта двигателя (стр. 94)

4.5.16.5 SendMessage

Введение

Сообщение отправляется в окно. Функция используется для заполнения, например, поля для выбора.

Синтаксис

LRESULT SendMessage (HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)

Параметры

Параметры	Описание
HWND hWnd	Манипулятор окна
UINT Msg,	Тип сообщения: LB_INSERTSTRING = вставка текста в ListBox
WPARAM wParam	1. Параметр сообщения: -1 = текст добавляется в конец.
LPARAM lParam	2. Параметр сообщения: Указатель на текст

Возвращаемое значение

	Возвращаемое значение
LRESULT	Манипулятор объекта

4.5.16.6 GetWindow

Введение

Определяется манипулятор окна, которое имеет определенное отношение к другому окну (исходное окно).

Синтаксис

GetWindow (HWND hWnd, UINT uCmd)

Параметры

Параметры	Описание
HWND hWnd	Манипулятор исходного окна
UINT uCmd	Отношение GW_HWNDFIRST = верхнее окно

Возвращаемое значение

	Возвращаемое значение
HWND	Манипулятор найденного окна или ZERO

4.5.16.7 ShowWindow

Введение

Указывается тип отображения окна.

Синтаксис

ShowWindow (HWND hWnd, int nCmdShow)

Параметры

Параметры	Описание
HWND hWnd	Манипулятор окна
int nCmdShow	Состояние отображения окна SW_HIDE = не отображается

Возвращаемое значение

	Возвращаемое значение
BOOL	TRUE = окно отображается FALSE = окно не отображается

4.5.16.8 MessageBox

Введение

Функция служит для отображения сообщения для пользователя, если произошла ошибка или требуется действие со стороны пользователя.

Сообщение отображается с использованием текста, заголовка и кнопки, указанных пользователем.

Синтаксис

int MessageBox (HWND hWnd, LPCTSTR lpText, LPCTSTR lpCaption, UINT uType)

Параметры

Параметры	Описание
HWND hWnd	Манипулятор родительского окна ZERO = сообщение не имеет родительного окна.
LPCTSTR lpText	Текст сообщения
LPCTSTR lpCaption	Текст заголовка
UINT uType	Тип окна MB_OK = сообщение с кнопкой OK MB_OKCANCEL = сообщение с кнопками OK и Cancel (Отмена)

Возвращаемое значение

	Возвращаемое значение
int	Идентификатор нажимаемой кнопки: IDOK = нажата кнопка OK IDCANCEL = нажата кнопка Cancel (Отмена)

Beispiel

int RetMsg;

RetMsg = MessageBox (ZERO, "Ошибка при вызове функций API", "Системная ошибка", MB_OK);

4.6 Примеры

4.6.1 Примеры

Введение

В тексте этого описания приведены два примера функций Dynamic Wizard (Мастер динамики):

- Мастер демонстрации
- Динамический объект двигателя

Дополнительные источники информации

Динамический объект двигателя (стр. 93)

Мастер демонстрации (стр. 90)

4.6.2 Мастер демонстрации

4.6.2.1 Мастер демонстрации

Введение

В файле Demo.wnf создается Dynamic Wizard (Мастер динамики), который называется Demo Wizard (Мастер демонстрации). Этот Dynamic Wizard (Мастер динамики) отображает основные функции, которые позволяют сделать ввод данных удобным для пользователя. Тем не менее, Demo Wizard (Мастер демонстрации) не выполняет действия.

Дополнительные источники информации

Добавление сценария Demo.wnf в базу данных (стр. 92)

Создание текста справки (стр. 91)

Создание функции мастера динамики для мастера демонстрации (стр. 90)

4.6.2.2 Создание функции мастера динамики для мастера демонстрации

Требования

Проект WinCC должен быть открыт.

Процедура

1. В Проводнике Windows скопируйте файл Demo.wnf из каталога Siemens\WinCC\documents\german в каталог Siemens\WinCC\wscript\wscript.deu.
2. Запустите Dynamic Wizard Editor (Редактор мастера динамики).
3. Выберите в меню File (Файл) окна Dynamic Wizard Editor (Редактор мастера динамики) пункт Open (Открыть). Откроется диалоговое окно выбора файла.
4. Выберите файл Demo.wnf. Нажмите Open (Открыть). Файл Demo.wnf отобразится в окне редактора.
5. Нажмите значок  в панели инструментов, чтобы скомпилировать сценарий. Результат отобразится в окне вывода.

Дополнительные источники информации

Добавление сценария Demo.wnf в базу данных (стр. 92)

4.6.2.3 Создание текста справки

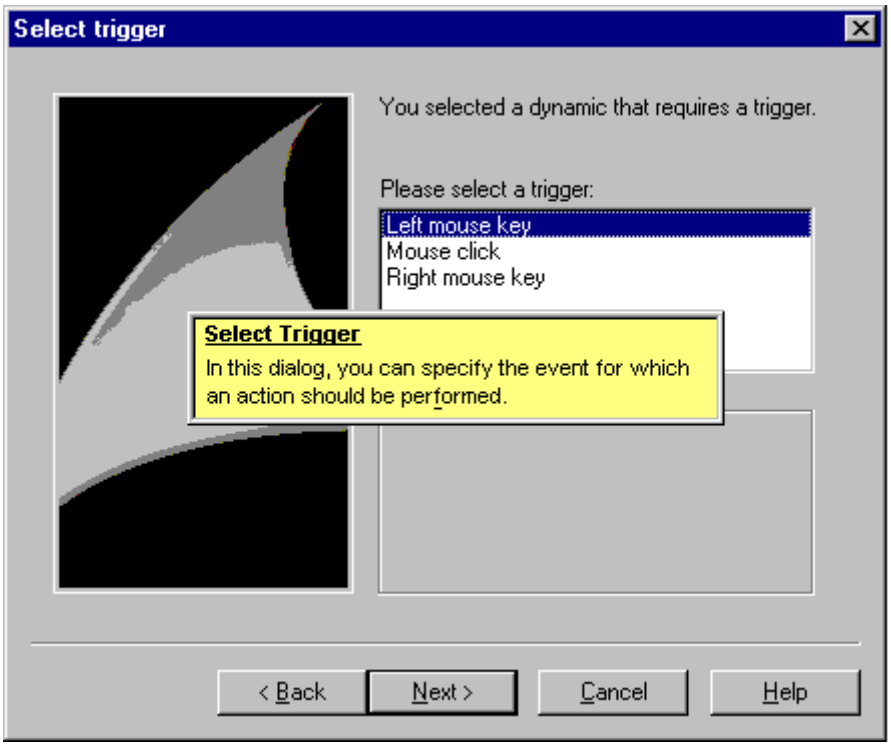
Введение

В этом разделе создается справка для диалогового окна Select trigger (Выбор триггера).

Процедура

1. Нажмите значок  в панели инструментов. Откроется редактор справки.
2. В поле Wizard - Group (Мастер — группа) выберите Example (Пример).
3. В поле Wizard - Name (Мастер — имя) выберите Demo Wizard (Мастер демонстрации).
4. В поле Page (Страница) выберите TriggerPage.
5. В поле Help - Text (Справка — текст) введите следующий текст: "Выбор триггера В этом диалоговом окне можно указать результат, для которого должно выполняться действие."
6. Закройте редактор справки, нажав кнопку OK.

7. Запустите Demo Wizard (Мастер демонстрации). В диалоговом окне Select trigger (Выбор триггера) нажмите кнопку Help (Справка).



4.6.2.4 Добавление сценария Demo.wnf в базу данных

Введение

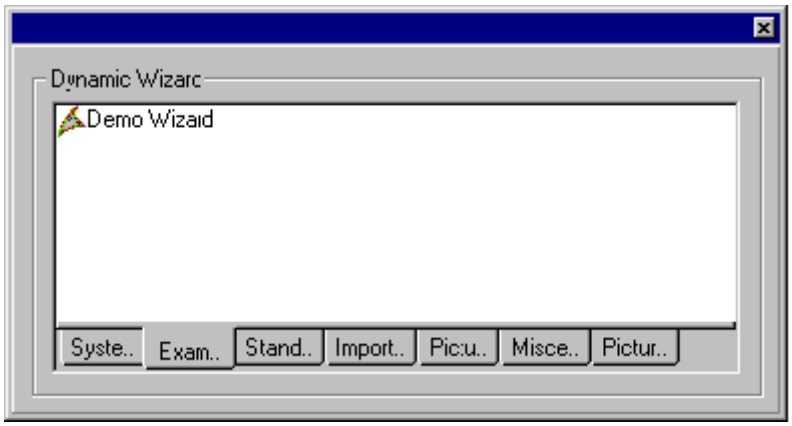
Чтобы использовать функцию Dynamic Wizard (Мастер динамики) Demo.wnf в графическом дизайнере, ее необходимо интегрировать в базу данных Dynamic Wizard (Мастер динамики).

Для этого необходимо выполнить следующие действия:

1. Импорт сценариев мастера
2. Создание файла cwd

Процедура

1. Нажмите значок  в панели инструментов. Откроется диалоговое окно выбора файла.
2. Выберите файл Demo.wnf. Нажмите Open (Открыть).
3. Нажмите значок  в панели инструментов, чтобы создать новую базу данных.
4. Выберите в меню View (Вид) окна Dynamic Wizard Editor (Редактор мастера динамики) пункт Dynamic Wizard (Мастер динамики).



1. Перейдите на вкладку Example (Пример). Дважды щелкните запись Demo Wizard (Мастер демонстрации).

4.6.3 Динамический объект двигателя

4.6.3.1 Динамический объект двигателя

Введение

В файле сценария Motor.wnf создается Dynamic Wizard (Мастер динамики), который называется Make Motor Dynamic (Динамизация объекта двигателя).

Примечание

Этот файл был создан специально для динамизации пользовательского объекта Motor (Двигатель) и его невозможно применить к объекту любого другого типа.

Дополнительные источники информации

Динамизация пользовательского объекта Motor (Двигатель) (стр. 95)

Добавление сценария Motor.wnf в базу данных (стр. 94)

Создание функции мастера динамики для объекта двигателя (стр. 94)

4.6.3.2 Создание функции мастера динамики для объекта двигателя

Требования

Проект WinCC должен быть открыт.

Процедура

1. В Проводнике Windows откройте Winzip-файл Motor.zip из каталога Siemens\WinCC\documents\german.
2. Распакуйте файл Motor.wnf в каталог "..\WinCC\wscripts\wscripts.deu".
3. Распакуйте файл Motor_dyn.pdl в каталог "..\WinCC\WinCCProjects\Имя_проекта_WinCC\GraCs".
4. Запустите Dynamic Wizard Editor (Редактор мастера динамики).
5. Выберите в меню File (Файл) окна Dynamic Wizard Editor (Редактор мастера динамики) пункт Open (Открыть). Откроется диалоговое окно выбора файла.
6. Выберите файл Motor.wnf. Нажмите Open (Открыть). Файл Motor.wnf отобразится в окне редактора.
7. Нажмите значок  в панели инструментов, чтобы скомпилировать сценарий. Результат отобразится в окне вывода.

Дополнительные источники информации

Добавление сценария Motor.wnf в базу данных (стр. 94)

4.6.3.3 Добавление сценария Motor.wnf в базу данных

Введение

Чтобы использовать функцию Dynamic Wizard (Мастер динамики) Motor.wnf в графическом дизайнере, ее необходимо интегрировать в базу данных Dynamic Wizard (Мастер динамики).

Для этого необходимо выполнить следующие действия:

1. Импорт сценариев мастера
2. Создание файла cwd

Процедура

1. Нажмите значок  в панели инструментов. Откроется диалоговое окно выбора файла.
2. Выберите файл Motor.wnf. Нажмите Open (Открыть).
3. Нажмите значок  в панели инструментов, чтобы создать новую базу данных.

Дополнительные источники информации

Динамизация пользовательского объекта Motor (Двигатель) (стр. 95)

4.6.3.4 Динамизация пользовательского объекта Motor (Двигатель)

Введение

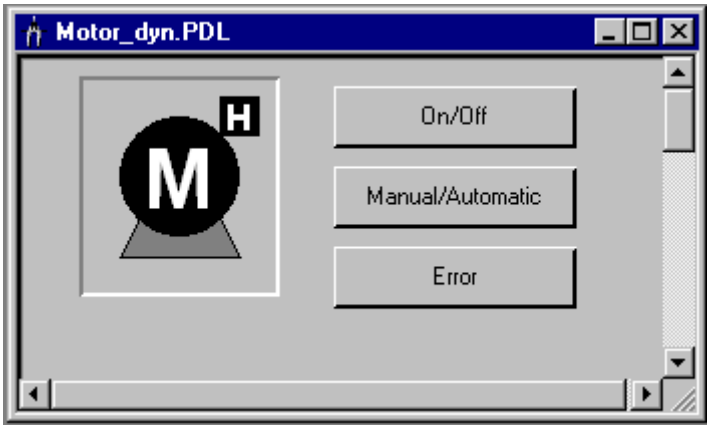
Пользовательский объект Motor (Двигатель) связан со структурным тегом WinCC типа MotorStruct посредством мастера дианмики Dynamic Motor (Динамический объект двигателя). В этом разделе для этого объекта создаются различные C-макросы и соединения тегов. Этот мастер невозможно использовать для объектов другого типа.

Требования

- Создайте внутренний тег T08i_course_wiz_selected с типом данных Text tag 8-bit character set (Текстовый тег 16-битовой кодировки).
- Создайте структуру с именем MotorStruct и три внутренних элемента с именами Active (Активный), Hand (Стрелка) и Error (Ошибка) и типом данных BIT.
- Создайте внутренний тег с именем STR_Course_wiz1 и типом данных MotorStruct.

Процедура

1. Запустите графический дизайнер. Выберите команду Open (Открыть) в меню File (Файл). Выберите кадр Motor_dyn.pdl в диалоговом окне выбора файла.
2. Выберите пользовательский объект Motor (Двигатель). На вкладке Example (Пример) отобразится мастер Dynamic Motor (Динамический объект двигателя).



1. Запустите Dynamic Wizard (Мастер динамики). Нажмите кнопку Continue (Продолжить) в диалоговом окне Welcome to the Dynamic Wizard (Добро пожаловать в мастер динамики). Откроется диалоговое окно Set options (Настройка параметров).
2. Нажмите кнопку Browse (Обзор) в диалоговом окне Set options (Настройка параметров). Откроется диалоговое окно выбора тега. Выберите STR_Course_wiz1 в структурного тега. Закройте диалоговое окно, нажав кнопку OK.
3. Нажмите кнопку Continue (Продолжить) в диалоговом окне Set options (Настройка параметров). Откроется диалоговое окно Finished! (Готово!). Закройте диалоговое окно, нажав кнопку OK.
4. Сохраните кадр. Запустите среду исполнения графического дизайнера.

5. Используйте кнопки для имитации значений тегов выбранного двигателя.

Дополнительные источники информации

Создание структуры и структурного тега (стр. 96)

4.6.3.5 Создание структуры и структурного тега

Введение

В данном разделе содержится описание конфигурации структуры MotorStruc и структурного тега STR_Course_wiz1. Структура и структурный тег используются в примере Dynamic motor (Динамический объект двигателя).

Процедура

1. Выберите в контекстном меню типов структур пункт New structure type (Новый тип структуры). Откроется диалоговое окно свойств структуры.
2. Измените имя структуры на MotorStruc. Нажмите кнопку New element (Создать элемент) и создайте внутренний тег Active (Активный) с типом данных BIT.
3. Нажмите кнопку New element (Создать элемент) и создайте внутренний тег Hand (Стрелка) с типом данных BIT.
4. Нажмите кнопку New element (Создать элемент) и создайте внутренний тег Error (Ошибка) с типом данных BIT. Закройте диалоговое окно, нажав кнопку OK.
5. Во фрейме навигации нажмите знак «плюс» перед значком для управления тегами. В контекстном меню внутренних тегов выберите пункт New tag (Создать тег). Создайте тег WinCC с именем STR_Course_wiz1 и типом данных MotorStruct.

Средство просмотра документации 5

5.1 Средство просмотра документации WinCC

Краткое описание

Задания печати системы отчетов WinCC можно перенаправить в файл. При наличии больших объемов данных для каждой страницы отчета создается отдельный файл.

Эти файлы можно просмотреть и распечатать с помощью **WinCC Documentation Viewer** (Средство просмотра документации WinCC).

5.2 Установка средства просмотра документации WinCC

WinCC Documentation Viewer (Средство просмотра документации WinCC) можно установить двумя способами.

Процедура

1. Во время установки WinCC в диалоговом окне Programs (Программы) выберите пункт WinCC V7.0 complete (Полная установка WinCC V7.0).

   WinCC установится вместе с пакетами SmartTools, WinCC ConfigurationTool и WinCC Archive ConfigurationTool.

Запустите WinCC Documentation Viewer (Средство просмотра документации WinCC), выбрав SIMATIC > WinCC > Tools (Инструменты).

Альтернативная процедура

Можно также установить WinCC Documentation Viewer (Средство просмотра документации WinCC) с DVD-диска WinCC.

1. Перейдите в каталог на DVD-диске WinCC "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Дважды щелкните значок setup.exe.
3. Выберите в диалоговом окне Components (Компоненты) пункт WinCC Documentation Viewer (Средство просмотра документации WinCC).
4. Нажмите кнопку Continue (Продолжить). Следуйте инструкциям в диалоговых окнах.

Примечание

Если проект WinCC запущен, можно просмотреть и распечатать только файлы emf этого проекта. Если система WinCC не запущена, с помощью WinCC Documentation Viewer (Средство просмотра документации WinCC) можно открыть и распечатать все файлы emf.

5.3 Описание

Введение

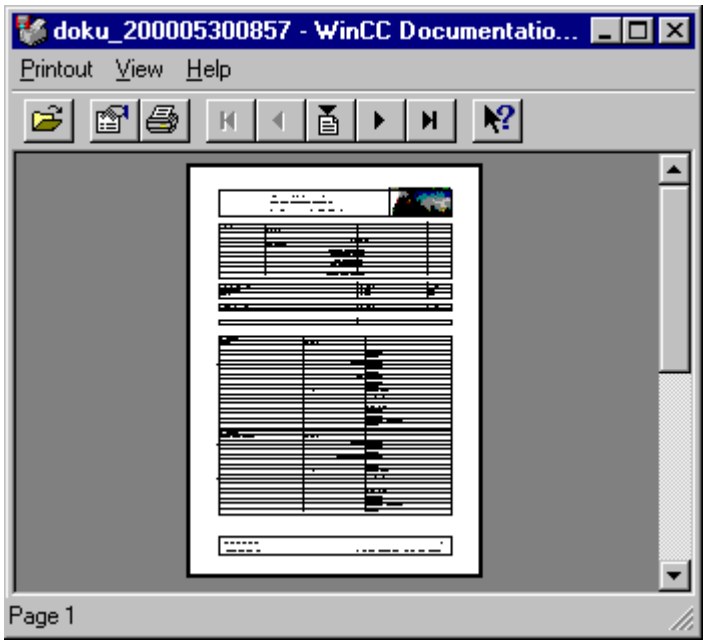
Задания печати можно перенаправить в файл. При наличии больших объемов данных для каждой страницы отчета создается отдельный файл.

Эти файлы можно просмотреть и распечатать с помощью WinCC Documentation Viewer (Средство просмотра документации WinCC).

Примечание

Если проект WinCC уже открыт при запуске WinCC Documentation Viewer (Средство просмотра документации WinCC), можно просмотреть и распечатать только файлы emf этого проекта.
Если система WinCC открыта, но не активна при запуске средства просмотра, с помощью средства просмотра можно открыть и распечатать все файлы emf.

При отключении среды исполнения средство просмотра закроется в любом случае.



WinCC Documentation Viewer (Средство просмотра документации WinCC) состоит из трех областей.

В верхней границе экрана расположена строка меню. Описание пунктов меню приведено в прямой справке.

Панель инструментов находится непосредственно под строкой меню. Часто используемые функции, например Page Up (Предыдущая страница) и Page Down (Следующая страница) загружаются на эту панель в виде значков. Описание функций отдельных значков приведено в прямой справке.

В окне отображается текущий документ. Область отображения можно увеличить, выполнив два щелчка.

Снизу экран ограничен строкой состояния, в которой отображается информация о текущей операции.

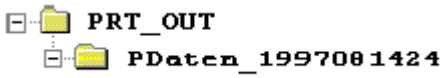
5.4 Создание файлов .emf

Введение

Задания печати можно перенаправить в файл. При наличии больших объемов данных для каждой страницы отчета создается отдельный файл. Вывод на печать перенаправляется в один или несколько файлов .emf. Файлы сохраняются с именем Page <nnnnnn>.emf в пути, где <nnnnn> обозначает пятизначный последовательный номер.

Имя пути выглядит следующим образом: из пути проекта (например, C:\VFSWinCC\PRT\) и <storage> + <YYYYMMDDHHMM> (YYYY = год, MM = месяц, DD = день, HH = час, MM = минута).

Если ввести значение PDdata в поле Storage (Хранение), путь со следующей структурой создается для задания печати в каталоге проекта.



Процедура

1. Выберите команду Project documentation setup (Настройка документации проекта) в меню File (Файл) в редакторах WinCC.
2. Перейдите на вкладку Printer setup (Настройка принтера) в диалоговом окне Print job properties (Свойства задания печати).
3. Установите флажок File (*.emf) (Файл) на вкладке Printer Setup (Настройка печати). Если одновременный вывод на принтер не требуется, снимите флажок Printer (Принтер).
4. В поле Storage (Хранение) введите имя пути, по которому требуется сохранить файл. Закройте диалоговое окно, нажав кнопку OK.
5. Выберите пункт Print project documentation (Печать документации проекта) в меню File (Файл). Вывод на печать перенаправляется в один или несколько файлов .emf.

Файлы сохраняются с именем Page <nnnnnn>.emf в пути, где <nnnnn> обозначает пятизначный последовательный номер.

WinCC CrossReferenceAssistant

6

6.1 WinCC CrossReferenceAssistant

Краткое описание

WinCC CrossReferenceAssistant представляет собой инструмент, с помощью которого осуществляется поиск имен кадров и тегов в сценариях и который дополняет сценарии таким образом, что компонент WinCC **Cross Reference** (Перекрестная ссылка) выполняет поиск имен кадров и тегов и перечисляет их в списке перекрестных ссылок.

6.2 Установка CrossReferenceAssistant

Для интерфейса пользователя инструмента WinCC CrossReferenceAssistant можно выбрать немецкий, английский и французский языки.

Процедура

1. Во время установки WinCC в диалоговом окне Programs (Программы) выберите пункт WinCC V7.0 complete (Полная установка WinCC V7.0).

   WinCC установится вместе с пакетами SmartTools, WinCC ConfigurationTool и WinCC Archive ConfigurationTool.

Запустите WinCC CrossReferenceAssistant, выбрав SIMATIC > WinCC > Tools (Инструменты).

Альтернативная процедура

WinCC CrossReferenceAssistant можно также установить с DVD-диска WinCC.

1. Перейдите в каталог на DVD-диске WinCC "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Дважды щелкните значок setup.exe.
3. Выберите запись CrossReferenceAssistant в диалоговом окне Components (Компоненты).
4. Нажмите кнопку Continue (Продолжить). Следуйте инструкциям в диалоговых окнах.

6.3 Общие сведения

Система WinCC может создавать списки CrossReference. Чтобы обеспечить надлежащее распознавание тегов в вызовах функций при создании этих списков, в WinCC добавлено правило конфигурации, которое предоставляет следующие возможности.

Чтобы иметь возможность поиска и замены имен тегов и кадров в C-макросах, сценарий должен быть написан следующим образом.

В начале сценария все имена тегов и кадров должны быть объявлены в двух разделах. В разделах запрещается вводить дополнительные инструкции.

Структура разделов выглядит следующим образом:

```
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
#define ApcVarName1 "VarName1"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
#define ApcPictureName1 "PictureName1"
#define ApcPictureName2 "PictureName2"
#define ApcPictureName3 "PictureName3"
// WINCC:PICNAME_SECTION_END
```

Вызов стандартный функций для чтения и записи тегов должен выполняться через заданные теги и кадры.

```
GetTagDWord (ApcVarName1);
OpenPicture(ApcBildname1);
SetPictureName( ApcPictureName2, "PictureWindow1",ApcPictureName3);
```

Если правило конфигурации не соблюдается, невозможно создать списки CrossReference, поскольку ссылки на теги и кадры в сценариях не могут быть разрешены.

С помощью WinCC CrossReferenceAssistant все вызовы функций, известные в Script Managment (Управление сценариями), заменяются на описанный выше формат. Преобразуются только функции проекта, кадры и макросы.

Средой исполнения для инструмента WinCC CrossReferenceAssistant является WinCC. Если система WinCC не запущена или преобразуемый проект не загружен, запуск WinCC или загрузка проекта осуществляются инструментом WinCC CrossReferenceAssistant.

Дополнительные источники информации

Известные функции (управление сценариями) (стр. 105)

6.4 Известные функции (управление сценариями)

Перечисленные ниже функции известны мастеру по умолчанию и реализуются во время преобразования.

Функции с тегами в качестве параметров

GetTagBit()

GetTagByte()

GetTagChar()

GetTagDouble()

GetTagDWord()

GetTagFloat()

GetTagRaw()

GetTagSByte()

GetTagSDWord()

GetTagSWord()

GetTagWord()

SetTagBit()

SetTagByte()

SetTagChar()

SetTagDouble()

SetTagDWord()

SetTagFloat()

SetTagRaw()

SetTagSByte()

SetTagSDWord()

SetTagSWord()

SetTagWord()

GetTagBitWait()

GetTagByteWait()

GetTagCharWait()

GetTagDoubleWait()

GetTagDWordWait()

GetTagFloatWait()

GetTagRawWait()

GetTagSByteWait()

GetTagSDWordWait()

GetTagSWordWait()
GetTagWordWait()

SetTagBitWait()
SetTagByteWait()
SetTagCharWait()
SetTagDoubleWait()
SetTagDWordWait()
SetTagFloatWait()
SetTagRawWait()
SetTagSByteWait()
SetTagSDWordWait()
SetTagSWordWait()
SetTagWordWait()

GetTagBitState()
GetTagByteState()
GetTagCharState()
GetTagDoubleState()
GetTagDWordState()
GetTagFloatState()
GetTagRawState()
GetTagSByteState()
GetTagSDWordState()
GetTagSWordState()
GetTagWordState()

SetTagBitState()
SetTagByteState()
SetTagCharState()
SetTagDoubleState()
SetTagDWordState()
SetTagFloatState()
SetTagRawState()
SetTagSByteState()
SetTagSDWordState()
SetTagSWordState()
SetTagWordState()

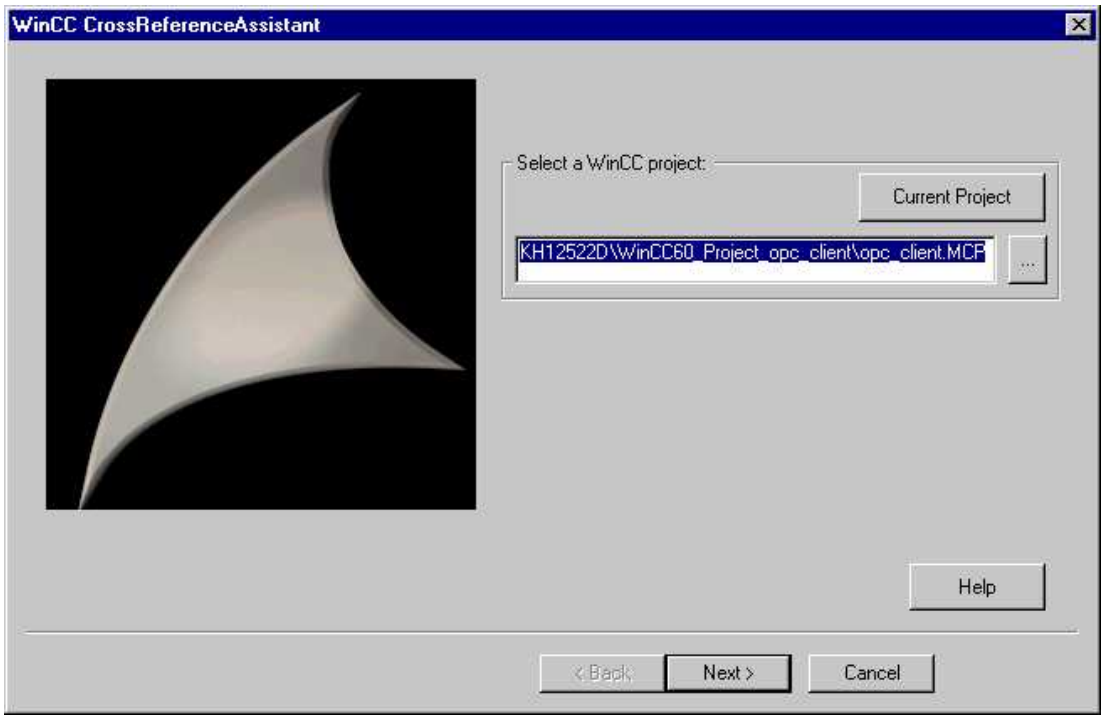
GetTagBitStateWait()
GetTagByteStateWait()
GetTagCharStateWait()
GetTagDoubleStateWait()
GetTagDWordStateWait()
GetTagFloatStateWait()
GetTagRawStateWait()
GetTagSByteStateWait()
GetTagSDWordStateWait()
GetTagSWordStateWait()
GetTagWordStateWait()

SetTagBitStateWait()
SetTagByteStateWait()
SetTagCharStateWait()
SetTagDoubleStateWait()
SetTagDWordStateWait()
SetTagFloatStateWait()
SetTagRawStateWait()
SetTagSByteStateWait()
SetTagSDWordStateWait()
SetTagSWordStateWait()
SetTagWordStateWait()

Функции с именами кадров в качестве параметров

SetPictureName()
GetPictureName()
GetVisible()
SetVisible()
GetLink()
SetLink()
Set_Focus()
OpenPicture()
GetLinkedVariable()

6.5 Выбор проекта



При нажатии кнопки **...** открывается диалоговое окно Open File (Открыть файл), в котором можно выбрать любой проект. При нажатии кнопки **Current project** (Текущий проект) инструмент WinCC CrossReferenceAssistant пытается импортировать и отобразить проект, который в настоящее время загружен в WinCC. Если система WinCC не запущена или проект не загружен, система запускается или загружается необходимый проект.

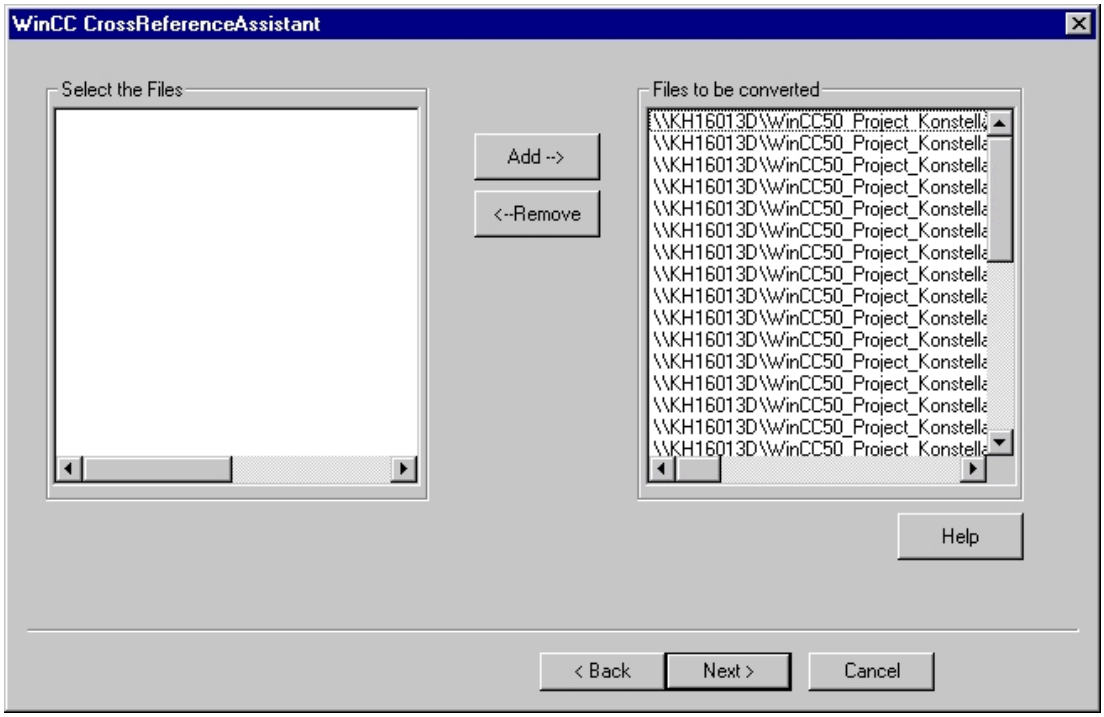
Если загружен другой проект, он будет закрыт и загрузится необходимый проект. Этот процесс займет некоторое время.

После ввода текста в строку ввода при запросе **Select a WinCC project** (Выберите проект WinCC) нажмите кнопку **Next >** (Далее >). Будет выполнена проверка проекта, является ли он действительным проектом WinCC. Если проект недействительный, фокус перемещается на строку ввода и открывается окно сообщения с описанием соответствующей ошибки.

При нажатии кнопки **Cancel** (Отмена) окно WinCC CrossReferenceAssistant закроется.

6.6 Выбор файла

Все кадры, функции проекта и C-макросы, относящиеся к проекту, отображаются в правом списке диалогового окна. В настройке по умолчанию преобразуются все файлы, относящиеся к проекту.



Пользователь может исключить некоторые файлы из преобразования, чтобы добавить их позднее. Для удаления файлов из списка преобразования выполните (множественный) выбор соответствующих файлов в списке **Files to be converted** (Преобразуемые файлы) и нажмите кнопку **<--Remove** (<--Удалить).

Удаленные файлы отобразятся в левом списке и их можно снова добавить для преобразования. Для этого необходимо выбрать эти файлы в списке **Select files** (Выбранные файлы). При нажатии кнопки **Add-->** (Добавить-->) они будут добавлены в правый список **Files to be converted** (Конвертируемые файлы).

После выбора файлов нажмите кнопку **Next >** (Далее >). Указанные файлы считываются и анализируются.

При нажатии кнопки **< Previous** (< Назад) происходит переход обратно к выбору проекта. При нажатии кнопки **Cancel** (Отмена) окно WinCC CrossReferenceAssistant закроется.

Дополнительные источники информации

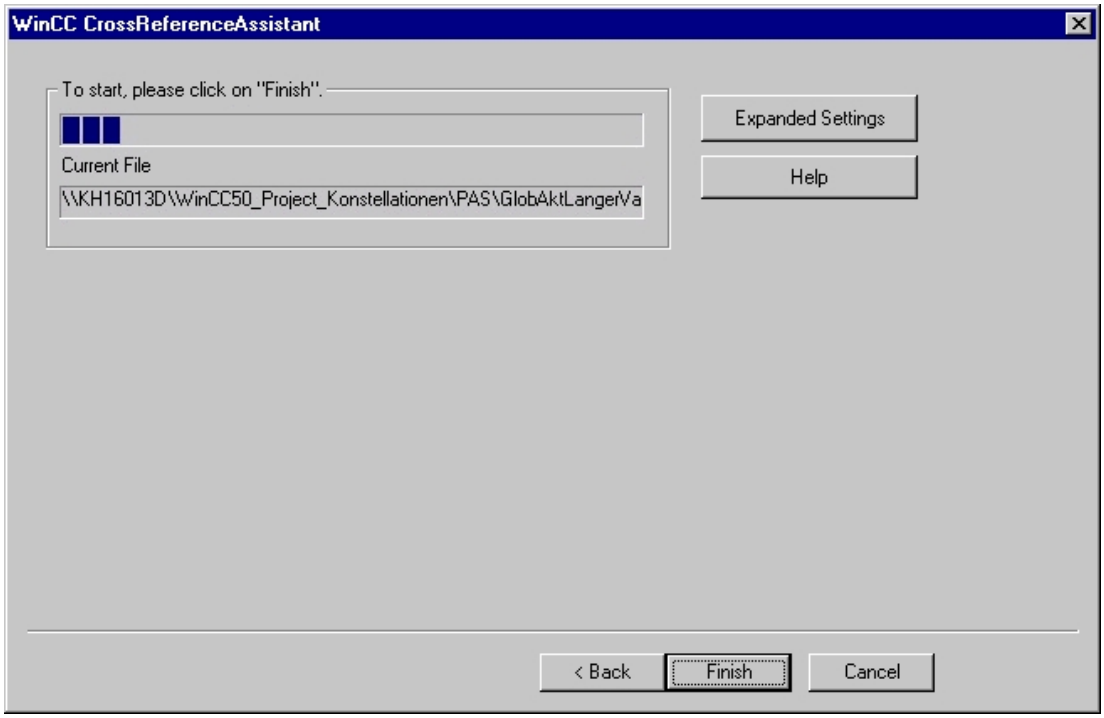
Выбор проекта (стр. 108)

6.7 Преобразование

Введение

На последней странице мастеров можно выполнить настройки в разделе Advanced Settings (Расширенные настройки) (см. «Расширенные настройки»), а также просмотреть ход выполнения и обрабатываемый в настоящий момент файл после запуска преобразования.

Описание



При нажатии кнопки < Back (< Назад) происходит переход обратно к окну File Selection (Выбор файла). При нажатии кнопки Cancel (Отмена) окно WinCC CrossReferenceAssistant закроется.

Для запуска преобразования сценария нажмите кнопку Finish (Готово). После начала преобразования невозможно выполнить переход назад (с помощью кнопки < Back (< Назад)) или нажать кнопку Advanced Settings (Расширенные настройки).

Во время преобразования выводится индикатор выполнения, который отображает выполнение преобразования в процентах. Можно также увидеть, какой файл преобразуется в настоящий момент.

Преобразование выполняется следующим образом. Сценарии проверяются на наличие вызовов функций, для которых ожидаются параметры кадров и тегов. Если такая функция будет найдена в сценарии, строка символов, передаваемая в качестве параметра, будет заменена определением (см. правила конфигурации).

Файл управления сценариями проверяет, для каких функций ожидаются параметры кадров или тегов. Поэтому, все эти функции необходимо ввести в этот файл и таким образом ввести в систему. Преобразование сценария может также использоваться для

внесения в список этих функций функций проекта и стандартных функций, для которых также ожидаются параметры кадров и тегов (расширенные настройки).

По завершении преобразования отображается сводка, в которой представлена информация о количестве преобразованных функций, кадров и сценариев в кадрах, а также тегов.

При возникновении ошибки можно получить более подробную информацию о причине ошибки, просмотрев файл журнала, созданный во время преобразования. Этот файл расположен в каталоге проекта и называется CCCrossReferenceAssistant.log.

Дополнительные источники информации

Расширенные настройки (стр. 111)

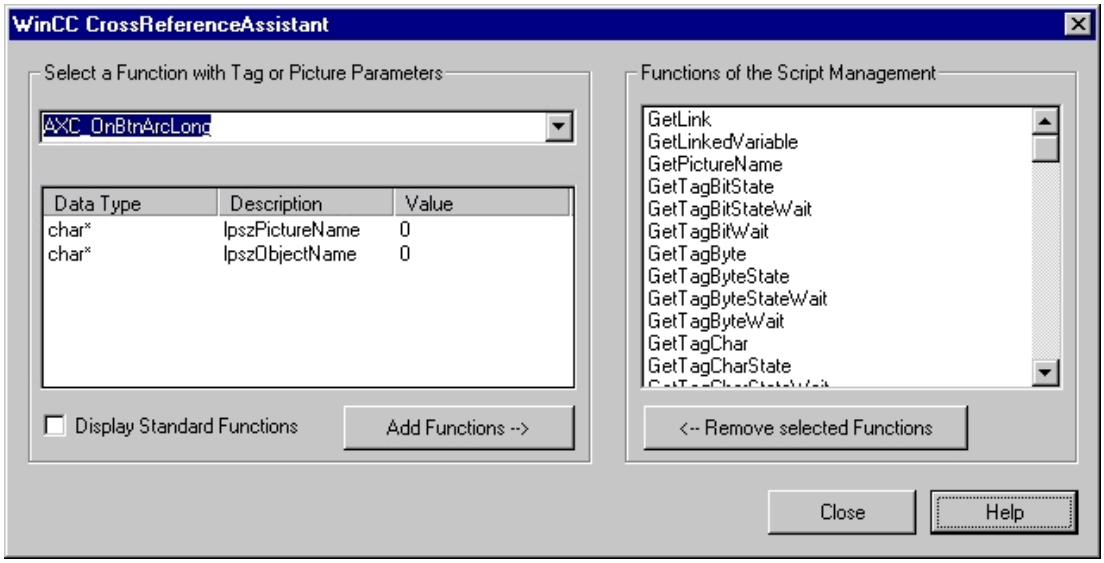
Общие сведения (стр. 104)

Выбор файла (стр. 109)

6.8 Расширенные настройки

При необходимости можно включить функции, созданные самостоятельно.

В списке **Select a function with tag and picture parameters:** (Выберите функцию с параметрами тегов и кадров) отображаются все функции проекта. Если установлен флажок **Display standard functions** (Отображать стандартные функции), то функции WinCC по умолчанию будут также отображены.



Пользователь может выбрать функцию из списка функций, для которых ожидается тег или кадр в качестве параметра в определенном положении. Все выбранные здесь функции включаются в файл управления сценариями для проекта.

Инструмент CrossReferenceAssistant распознает в качестве функций с параметрами кадров и тегов только те функции, которые были включены в систему как таковые. Чтобы обеспечить адаптацию вызовов пользовательских функций, для которых ожидаются параметры тегов, в соответствии с правилами конфигурации, эти функции необходимо включить в управление сценариями в ходе преобразования.

В диалоговом окне **Functions of the script management** (Функции управления сценариями) отображаются все функции, которые уже добавлены в управление сценариями. Когда это диалоговое окно отображается, считываются стандартные файлы и файлы конфигурации проектов и отображается общее содержимое этих файлов.

Для указания функции, для которой ожидается параметр тега или кадра, необходимо сначала выбрать ее в комбинированном поле **Select a function with tag or picture parameters:** (Выберите функцию с параметрами тегов или кадров).

С помощью списка параметров можно затем определить, представляет ли конкретный параметр тег или кадр. При нажатии кнопки **...** откроется всплывающее меню, с помощью которого пользователь может выбрать, является ли выбранный параметр тегом или кадром.

Эту процедуру необходимо повторить для всех параметров, к которым применяется один из критериев.

"Кнопка **Add function -->** (Добавить функцию -->) подтверждает ввод и добавляет выбранную функцию в список с правой стороны диалогового окна. В случае ошибки можно отменить действие, выбрав удаляемую функцию в списке **Functions of the script management** (Функции управления сценариями) и удалив ее из списка нажатием кнопки **<-- Remove selected functions** (<-- Удалить выбранные функции).

При нажатии кнопки **Close** (Закрыть) информация о группе записывается в файлы конфигурации, измененная информация учитывается во время преобразования и диалоговое окно закрывается.

Дополнительные источники информации

Общие сведения (стр. 104)

Файлы регистрации OnlOff.reg, OnlOn.reg

7

Краткое описание

Опциональный пакет Process Control Runtime (Среда исполнения для управления процессами) имеет начальный интерфейс, в котором имеется кнопка для вызова интерактивной справки во время работы среды исполнения. Эта интерактивная справка позволяет выполнять операции в системе Windows, даже если они заблокированы.

С помощью файла **OnlOff.reg** доступ к операциям в системе Windows можно заблокировать, скрыв кнопку интерактивной справки. При открытии этого файла в реестр Windows вносятся необходимые записи.

Для отмены этой операции откройте файл OnlOn.reg. Кнопка для вызова интерактивной справки отобразится в области кнопки.

Процедура установки

1. Оба файла расположены на DVD-диске с продуктом в каталоге SmartTools\CC_Onloff.
2. Скопируйте весь каталог CC_OnlOff с DVD-диска в каталог SmartTools на жестком диске, на котором установлена система WinCC, например C:/Programme/Siemens/WinCC.

WinCC Communication Configurator 8

WinCC Communication Configurator (CCComunicationConfigurator.exe) — это инструмент, с помощью которого можно легко установить параметры связи WinCC для доступной сетевой среды.

WinCC Communication Configurator выгодно использовать при отсутствии сети Ethernet LAN со скоростью передачи данных 100 Мбит/с. Также рекомендуется использовать конфигуратор, когда подключение время от времени нестабильно в результате высокой нагрузки (например, отсутствие подключения к серверам данных, поля ввода-вывода без отображаемого значения).

Связь WinCC настраивается с помощью стандартных параметров таким образом, что она очень чувствительна к ошибкам соединения, например для быстрого сообщения пользователю о произошедших неисправностях или для обеспечения быстрого переключения после сбоя на резервный сервер в случае клиентского компьютера.

В сети с низкой скоростью передачи данных или при высокой нагрузке на сеть/ЦПУ стабильность логических сетевых соединений WinCC зависит от этого чувствительного к ошибкам режима, поскольку ожидаемое время обратной связи невозможно обеспечить в низкоуровневых механизмах мониторинга работоспособности.

Communication Configurator (Конфигуратор связи) адаптирует параметры связи к существующему сценарию, чтобы достичь оптимального баланса между чувствительностью к ошибкам и устойчивостью соединения.

Примечание

Если WinCC Communication Configurator используется таким образом, эту операцию необходимо выполнять на клиенте WinCC и на сервере WinCC.

Communication Configurator (Конфигуратор связи) изменяет только параметры связи WinCC, но не параметры каналов связи операционной системы.

Поле/параметр	Описание
100 MBit/s (Мбит/с)	Применяется для сетей Ethernet LAN со скоростью передачи данных 100 Мбит/с (настройка по умолчанию).
10 MBit/s (Мбит/с)	Для сетей Ethernet LAN со скоростью передачи данных 10 Мбит/с.
1 MBit/s (Мбит/с)	Для сетей со скоростью передачи данных 1 Мбит/с.
0,1 MBit/s (Мбит/с)	Для сетей или каналов связи со скоростью передачи данных 0,1 Мбит/с. Эта настройка подходит для каналов, использующих ISDN (MultiLink), ISDN и модем.
Флажок Server pings client (Сервер опрашивает клиента)	Проверяет соединение с клиентом с помощью сервера.
Кнопка Default (По умолчанию)	Устанавливает значение по умолчанию Ethernet LAN (100 MBit/s) (Ethernet LAN (100 Мбит/с)) для настройки

Примечание

Рекомендуется использовать клиентов со скоростью передачи по меньшей мере 128 Кбит/с.