

SIEMENS

SIMATIC HMI

WinCC V7.0

Пользовательский архив

Печатная версия справки

Введение в
пользовательские архивы

1

Редактор User Archive
(Пользовательский архив)

2

Сценарии
пользовательских архивов

3

Правила техники безопасности

В этом руководстве содержатся примечания, которые необходимо соблюдать для обеспечения личной безопасности и предотвращения материального ущерба. В этом руководстве примечания, относящиеся к личной безопасности, обозначены символом предупреждения о безопасности; примечания, касающиеся только материального ущерба, не обозначаются символом предупреждения о безопасности. Эти перечисленные ниже примечания различаются по степени опасности.

⚠ ОПАСНО
Означает, что невыполнение надлежащих мер безопасности повлечет за собой смерть или серьезную травму.
⚠ ПРЕДУПРЕЖДЕНИЕ
Означает, что невыполнение надлежащих мер безопасности может повлечь за собой смерть или серьезную травму.
⚠ ВНИМАНИЕ!
При наличии символа предупреждения о безопасности это примечание означает, что невыполнение надлежащих мер безопасности может повлечь за собой несерьезные травмы.
ВНИМАНИЕ!
При отсутствии символа предупреждения о безопасности это примечание означает, что невыполнение надлежащих мер безопасности может повлечь за собой материальный ущерб.
ПРИМЕЧАНИЕ
Означает, что игнорирование соответствующей информации может привести к нежелательным результатам или последствиям.

При возникновении нескольких степеней опасности используется предупреждение, относящееся к наивысшей степени опасности. Предупреждение, относящееся к возможным травмам, с символом предупреждения о безопасности может также содержать предупреждение о возможном материальном ущербе.

Квалифицированный персонал

Установка и использование устройства/системы должны осуществляться исключительно в соответствии с настоящей документацией. Ввод устройства/системы в эксплуатацию и их эксплуатацию должен выполнять только **квалифицированный персонал**. В контексте правил техники безопасности настоящей документации под квалифицированным персоналом подразумеваются лица, которым разрешено осуществлять ввод в эксплуатацию, заземление и маркировку устройств, систем и электрических цепей в соответствии с установленными правилами и стандартами по технике безопасности.

Использование по назначению

Обратите внимание на следующие моменты.

⚠ ПРЕДУПРЕЖДЕНИЕ
Это устройство можно использовать только в целях, указанных в каталоге или техническом описании, и только совместно с устройствами либо компонентами сторонних производителей, одобренных или рекомендуемых компанией Siemens. Для обеспечения правильной, надежной эксплуатации продукта требуется надлежащая транспортировка, хранение, расположение и сборка, а также аккуратная эксплуатация и техническое обслуживание.

Торговые знаки

Все названия, отмеченные знаком ®, являются зарегистрированными торговыми знаками компании Siemens AG. Прочие торговые знаки, упомянутые в настоящем документе, могут быть торговыми знаками соответствующих владельцев. Использование таких торговых знаков третьими лицами в собственных целях может быть нарушением прав этих владельцев.

Отказ от ответственности

Мы проверили содержимое настоящего документа, чтобы убедиться в том, что оно соответствует описанному в нем оборудованию и программному обеспечению. Поскольку невозможно предусмотреть все изменения, полное соответствие не гарантируется. Тем не менее компания осуществляет проверку информации, представленной в настоящем документе, а также вносит все необходимые исправления в последующие издания.

Содержание

1	Введение в пользовательские архивы	7
2	Редактор User Archive (Пользовательский архив)	9
2.1	Введение в редактор User Archive (Пользовательский архив)	9
2.1.1	Редактор User Archive (Пользовательский архив)	9
2.1.2	Элемент управления WinCC UserArchiveControl	10
2.1.3	Стандартная функция языка сценариев WinCC	11
2.1.4	Области применения пользовательского архива	11
2.1.5	Набор функций пользовательских архивов	13
2.2	Редактор User Archive (Пользовательский архив)	15
2.2.1	Структура редактора User Archive (Пользовательский архив)	15
2.2.2	Меню редактора User Archive (Пользовательский архив)	15
2.2.2.1	Меню редактора User Archive (Пользовательский архив)	15
2.2.2.2	Команда меню Restore (Восстановить)	16
2.2.2.3	Команда меню Export (Экспортировать) (меню Project (Проект))	17
2.2.2.4	Команда меню Import (Импортировать) (меню Project (Проект))	18
2.2.2.5	Команда меню Check (Проверить)	19
2.2.2.6	Команда меню Runtime Data (Данные среды исполнения)	19
2.2.2.7	Команда меню Options (Настройки)	20
2.2.2.8	Команда меню Split (Разбить)	20
2.2.2.9	Команда меню Import (Импорт) (меню Runtime Data (Данные среды исполнения))	21
2.2.2.10	Команда меню Export (Экспорт) (меню Runtime Data (Данные среды исполнения))	23
2.2.3	Панель инструментов редактора User Archive (Пользовательский архив)	25
2.2.4	Окно таблицы редактора User Archive (Пользовательский архив)	27
2.3	Конфигурация	28
2.3.1	Конфигурация	28
2.3.2	Конфигурация пользовательского архива	28
2.3.2.1	Создание нового пользовательского архива	28
2.3.2.2	Основные свойства полей пользовательского архива	30
2.3.2.3	Настройки связи	31
2.3.2.4	Настройка управляющих тегов	32
2.3.2.5	Настройка разрешений и флажков	33
2.3.2.6	Создание новых полей пользовательского архива	36
2.3.2.7	Основные свойства пользовательских архивов	37
2.3.2.8	Настройка значений полей пользовательского архива	38
2.3.2.9	Настройки Authorizations and Flags (Разрешения и флажки)	39
2.3.2.10	Свойства пользовательских архивов	40
2.3.2.11	Пользовательские архивы: вкладка Sequence (Последовательность)	41
2.3.2.12	Свойства полей пользовательского архива	42
2.3.2.13	Поля пользовательского архива: вкладка Sequence (Последовательность)	43
2.3.2.14	Свойства управляющих тегов	45
2.3.2.15	Пример использования управляющих тегов	47
2.3.3	Изменение конфигурации пользовательских архивов	52
2.3.4	Конфигурация представлений	53
2.3.4.1	Создание новых представлений	53
2.3.4.2	Создание новых столбцов представлений	55

2.3.4.3	Основные свойства столбцов представлений.....	56
2.3.4.4	Свойства представлений.....	57
2.3.4.5	Основные свойства представлений.....	58
2.3.4.6	Настройки связи между представлений.....	59
2.3.4.7	Настройка последовательности представлений.....	60
2.3.4.8	Свойства столбцов представлений.....	61
2.3.4.9	Настройка последовательности столбцов представлений.....	62
2.3.4.10	Советы по конфигурации.....	63
2.3.4.11	Изменение языка.....	64
2.3.5	Пример пользовательского архива.....	66
2.4	Обмен данными с SIMATIC S5/S7.....	67
2.4.1	Интерфейс фрейма сообщения SIMATIC.....	67
2.4.2	Обмен данными с S5 и S7 с помощью тегов WinCC.....	68
2.4.3	Обмен данными с S5 и S7 с помощью тегов необработанных данных.....	68
2.4.3.1	Обмен данными с S5 и S7 с помощью тегов необработанных данных.....	68
2.4.3.2	Отправка заданий/данных в WinCC.....	69
2.4.3.3	Отправка квитирования обработки /данных в SIMATIC S5/S7.....	69
2.4.3.4	Структура заголовка связи.....	69
2.4.3.5	Заголовок задания.....	70
2.4.3.6	Данные задания.....	71
2.4.3.7	Заголовок квитирования.....	71
2.4.3.8	Описание типов заданий и кодов ошибок.....	72
2.4.4	Различия в форматах данных между WinCC и S5/S7.....	73
2.5	Приложение.....	74
2.5.1	Приложение.....	74
2.5.2	Язык SQL.....	74
2.5.3	Алфавитный список ключевых слов SQL.....	75
2.5.4	Количественная структура.....	76
2.5.5	Характеристика чтения и записи тегов.....	77
2.5.6	Характеристика времени открытия экрана.....	78
3	Сценарии пользовательских архивов.....	79
3.1	Стандартные функции языка сценариев WinCC.....	79
3.2	Конфигурация макросов.....	80
3.3	Функции для редактирования и отображения пользовательских архивов.....	80
3.4	Дескрипторы для конфигурации пользовательских архивов.....	82
3.5	Дескрипторы функций среды исполнения.....	82
3.6	Пример сценария.....	85
3.7	Приложение.....	93
3.7.1	Описание стандартных функций языка сценариев WinCC.....	93
3.7.2	Стандартные функции для пользовательских архивов.....	93
3.7.2.1	uaGetLastError.....	93
3.7.2.2	uaGetLastHResult.....	96
3.7.3	Функции для конфигурации пользовательских архивов.....	96
3.7.3.1	Функции для конфигурации пользовательских архивов.....	96
3.7.3.2	uaAddArchive.....	97
3.7.3.3	uaAddField.....	97
3.7.3.4	uaGetArchive.....	98
3.7.3.5	uaGetField.....	98
3.7.3.6	uaGetNumArchives.....	99
3.7.3.7	uaGetNumFields.....	99

3.7.3.8	UaQueryConfiguration	100
3.7.3.9	uaReleaseConfiguration	100
3.7.3.10	uaRemoveAllArchives	101
3.7.3.11	uaRemoveAllFields	102
3.7.3.12	uaRemoveArchive	102
3.7.3.13	uaRemoveField	103
3.7.3.14	uaSetArchive	103
3.7.3.15	uaSetField	104
3.7.3.16	Структура для программирования дескриптора uaCONFIGFIELDA	105
3.7.3.17	Структура для программирования дескриптора uaCONFIGFIELDW	105
3.7.3.18	Структура для программирования дескриптора uaAddArchive	106
3.7.3.19	Структура для программирования дескриптора uaCONFIGARCHIVEA	107
3.7.4	Основные функции среды исполнения	108
3.7.4.1	Основные функции среды исполнения	108
3.7.4.2	uaConnect	109
3.7.4.3	uaDisconnect	109
3.7.4.4	uaGetLocalEvents	109
3.7.4.5	uaIsActive	110
3.7.4.6	uaOpenArchives	110
3.7.4.7	uaOpenViews	111
3.7.4.8	uaQueryArchive	111
3.7.4.9	uaQueryArchiveByName	112
3.7.4.10	uaReleaseArchive	113
3.7.4.11	uaSetLocalEvents	113
3.7.4.12	uaUsers	114
3.7.5	Функции среды исполнения для архивов	114
3.7.5.1	Функции среды исполнения для архивов	114
3.7.5.2	uaArchiveClose	115
3.7.5.3	uaArchiveDelete	116
3.7.5.4	uaArchiveExport	116
3.7.5.5	uaArchiveGetCount	117
3.7.5.6	uaArchiveGetFieldLength	118
3.7.5.7	uaArchiveGetFieldName	118
3.7.5.8	uaArchiveGetFields	119
3.7.5.9	uaArchiveGetFieldType	119
3.7.5.10	uaArchiveGetFieldValueDate	120
3.7.5.11	uaArchiveGetFieldValueDouble	120
3.7.5.12	uaArchiveGetFieldValueFloat	121
3.7.5.13	uaArchiveGetFieldValueLong	122
3.7.5.14	uaArchiveGetFieldValueString	122
3.7.5.15	uaArchiveGetFilter	123
3.7.5.16	uaArchiveGetID	123
3.7.5.17	uaArchiveGetName	124
3.7.5.18	uaArchiveGetSort	124
3.7.5.19	uaArchiveImport	125
3.7.5.20	uaArchiveInsert	125
3.7.5.21	uaArchiveMoveFirst	126
3.7.5.22	uaArchiveMoveLast	126
3.7.5.23	uaArchiveMoveNext	127
3.7.5.24	uaArchiveMovePrevious	127
3.7.5.25	uaArchiveOpen	127
3.7.5.26	uaArchiveReadTagValues	128
3.7.5.27	uaArchiveReadTagValuesByName	129
3.7.5.28	uaArchiveRequery	129
3.7.5.29	uaArchiveSetFieldValueDate	130

3.7.5.30	uaArchiveSetFieldValueDouble	131
3.7.5.31	uaArchiveSetFieldValueFloat.....	131
3.7.5.32	uaArchiveSetFieldValueLong.....	132
3.7.5.33	uaArchiveSetFieldValueString	132
3.7.5.34	uaArchiveSetFilter	133
3.7.5.35	uaArchiveSetSort	133
3.7.5.36	uaArchiveUpdate.....	134
3.7.5.37	uaArchiveWriteTagValues.....	135
3.7.5.38	uaArchiveWriteTagValuesByName.....	135
Индекс	137

Введение в пользовательские архивы

Во введении в пользовательские архивы можно получить следующую информацию:

- Области применения пользовательских архивов
- Компоненты редактора пользовательских архивов
- Конфигурация и среда исполнения
- Набор функций пользовательского архива

Редактор User Archives (Пользовательские архивы) WinCC может использоваться для непрерывного сохранения данных технических процессов на сервере ПК. В редакторе Graphics Designe (Графический дизайнер) можно настроить элемент управления WinCC UserArchiveControl, что обеспечит табличное отображение интерактивных данных из пользовательского архива в среде исполнения.

Пользовательские архивы также применяются, чтобы подготовить данные для систем автоматизации, например S5, S7. При необходимости данные можно считать с контроллера в виде рецептов или уставок.

В редакторе User Archive (Пользовательский архив) предлагается два типа таблиц базы данных.

- Пользовательские архивы: пользовательские архивы – таблицы базы данных, в которых пользователь может создавать собственные поля данных. Пользовательские архивы используются для хранения данных, и в них предлагается стандартный доступ к этим данным в соответствии с правилами баз данных SQL.
- Представления: представления получают данные из пользовательских архивов и используются для резюмирования данных, например для получения обзоров групп продуктов.

Существует два варианта создания или редактирования пользовательских архивов.

- Редактор User Archive (Пользовательский архив) для удобной интерактивной конфигурации пользовательских архивов.
- Функции для редактирования пользовательских архивов в языке сценариев WinCC.

С помощью функций языка сценариев WinCC можно обеспечить разнообразные макросы для среды исполнения. В экране среды исполнения можно настроить таблицу, соединенную напрямую с отображением процесса системы автоматизации.

Дополнительные источники информации

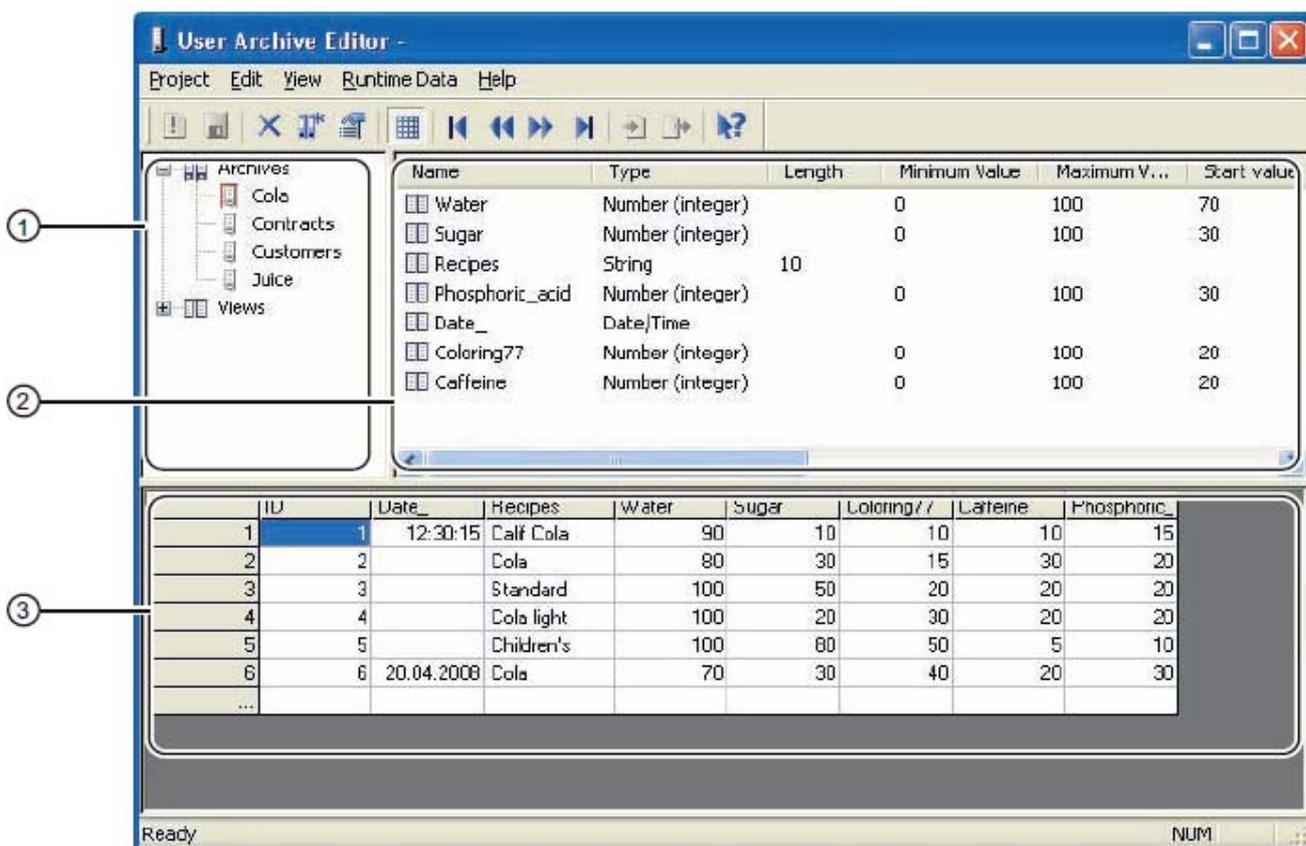
Области применения пользовательского архива (стр. 11)

Редактор User Archive (Пользовательский архив)

2.1 Введение в редактор User Archive (Пользовательский архив)

2.1.1 Редактор User Archive (Пользовательский архив)

Редактор User Archive (Пользовательский архив) предоставляет удобный способ для создания и сохранения пользовательских архивов. Рабочая область редактора User Archive (Пользовательский архив) делится на три части.



- ① Окно навигации
- ② Окно данных
- ③ Окно таблицы

- **Окно навигации** для выбора пользовательских архивов и представлений.
- **Окно данных** для отображения и изменения полей. В окне данных отображаются поля пользовательских архивов и представлений, которые были выбраны в окне навигации.
- **Окно таблицы** для отображения и изменения интерактивных данных выбранных пользовательских архивов и представлений. В окне таблицы редактора возможно онлайн-соединение с отображениями процессов ПЛК.

Окна навигации и данных редактора обеспечивают быстрый доступ ко всем элементам пользовательских архивов с помощью интерфейса пользователя, похожего на проводник. Создание и изменение пользовательских архивов производится с помощью диалоговых окон и мастеров, удобных для пользователя.

Дополнительные источники информации

Области применения пользовательского архива (стр. 11)

2.1.2 Элемент управления WinCC UserArchiveControl

Элемент управления WinCC UserArchiveControl можно настроить с помощью редактора Graphics Designer (Графический дизайнер). Элемент управления UserArchiveControl предназначен для отображения и изменения данных пользовательского архива в среде исполнения.

Основные свойства элемента управления WinCC UserArchiveControl:

- Управление элементом UserArchiveControl осуществляется с помощью значков и сочетаний клавиш.
- В UserArchiveControl можно создавать, изменять и удалять содержимое полей.
- Функция просмотра также упрощает доступ к большим пользовательским архивам.
- Можно осуществлять импорт и экспорт пользовательских архивов.
- Критерии выбора и условия сортировки можно использовать для структурирования отображаемых данных.
- При прямом соединении с системой автоматизации можно интерактивно считывать и записывать данные.

Во время конфигурации элемент управления UserArchiveControl подключается к выбранному пользовательскому архиву или форме и только после этого может получать доступ к этому пользовательскому архиву или форме. Для получения доступа необходимо разрешить пользовательский архив/форму (защита доступа). Элементу управления UserArchiveControl можно присвоить определенные разрешения в редакторе User Administrator (Администратор пользователей).

Если отменить защиту доступа, необходимо повторно подключить элемент UserArchiveControl в графическом дизайнера к пользовательскому архиву, чтобы элемент UserArchiveControl смог определить отключение защиты доступа.

Защита доступа к архиву или полю проверяется при открытии экрана элемента управления UserArchiveControl. Защиту доступа к управляющим тегам элементов

защищенных архивов необходимо устанавливать отдельно с помощью свойств объекта, например, изображения, поля ввода/вывода или кнопки.

Примечание

В версиях до WinCC V7 конфигурация отображения пользовательских архивов осуществлялась с помощью элемента управления User Archives Table Element (Табличный элемент пользовательских архивов).

Дополнительные источники информации

Области применения пользовательского архива (стр. 11)

2.1.3 Стандартная функция языка сценариев WinCC

Функции языка сценариев WinCC делятся на:

- **Функции конфигурации** для настройки пользовательских архивов
- **Функции среды исполнения** для конфигурации действий при работе среды исполнения

Функции среды исполнения активируются с помощью действий в экране среды исполнения, например щелчка кнопки мыши на определенной кнопке. Язык сценариев WinCC основан на языке высокого уровня C, а функции базы данных – на стандарте SQL.

Дополнительные источники информации

Введение в пользовательские архивы (стр. 7)

Области применения пользовательского архива (стр. 11)

2.1.4 Области применения пользовательского архива

Набор функций

При конфигурации пользовательских архивов можно создать собственные таблицы баз данных с помощью редактора User Archive (Пользовательский архив) или функций языка сценариев WinCC.

В редакторе User Archive (Пользовательский архив) можно создавать новые записи данных и редактировать данные существующих записей даже во время конфигурации.

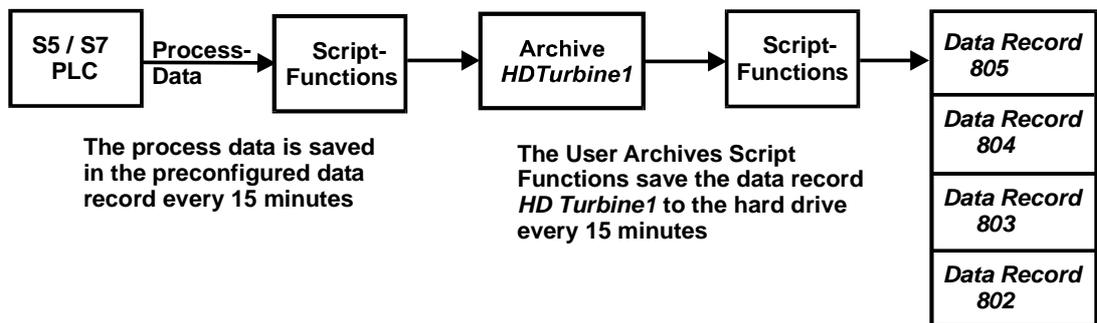
В среде исполнения пользовательские архивы (синоним таблицы баз данных) могут быть представлены в виде таблиц в окнах кадров элемента управления WinCC UserArchiveControl. С помощью тегов необработанных данных или тегов WinCC можно осуществлять непрерывный обмен данными с AS.

Пример с записью рабочих данных турбины

Оператор по выработке энергии устанавливает пользовательский архив HDTurbine1. Данный пользовательский архив используется для мониторинга рабочего состояния турбины высокого давления. Пользовательский архив HDTurbine1 имеет следующие поля данных.

HDTurbine1
Индекс
Скорость
Давление на входе
Давление на выходе
Температура пара1
Температура пара2
Частота колебаний
Амплитуда колебаний
Температура хранения1
Температура хранения2

В среде исполнения можно сохранять рабочие данные турбины с заданными интервалами в виде записей пользовательских архивов на запоминающем устройстве ПК (жестком диске).



S5/S7 PLC	ПЛК S5/S7
Process Data	Данные процесса
Script-Functions	Сценарий – функции
Archive HD Turbine 1	Архив турбины HD 1
Script-Functions	Сценарий – функции
Data Record	Запись данных
The process data is saved in the preconfigured data record every 15 minutes	Данные процесса сохраняются в предварительно настроенной записи данных каждые 15 минут
The User Archives Script Functions save the data record <i>HD Turbine 1</i> to the hard drive every 15 minutes	Функции сценария пользовательских архивов сохраняют запись данных <i>турбины HD 1</i> на жесткий диск каждые 15 минут

С помощью функций языка сценариев WinCC позже можно анализировать данные пользовательских архивов или визуализировать их с помощью элемента управления WinCC UserArchiveControl.

Пример с рецептами для изготовления напитков

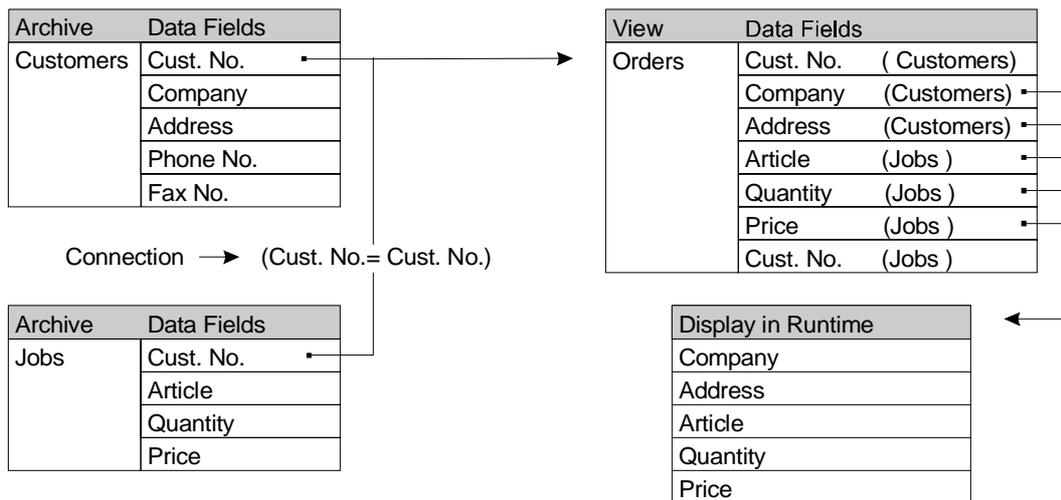
В качестве примера отправки данных в ПЛК служат рецепты. Изготовитель напитков, производящий, к примеру, напиток Cola и апельсиновый сок, использует пользовательские архивы, чтобы приготовить рецепты напитков для AS.

Пользовательский архив	Поля данных
Cola	Вода
	Сахар
	Краситель7
	Фосфорная кислота
	Кофеин

Пользовательские архивы используют интерфейс данных к AS, предоставляемый WinCC, посредством необработанных данных менеджеров данных WinCC или тегов WinCC. Для передачи данных из/в AS система WinCC предоставляет необходимый набор C-макросов.

Способы применения представлений

WinCC предлагает еще один мощный инструмент Views (представления) для работы с несколькими архивами сервера. С помощью представлений можно сводить поля данных разных пользовательских архивов. Так, например, можно создавать логические операции на языке SQL для полей данных различных пользовательских архивов для отображения требуемых связей в среде исполнения в виде форм. Применяемые пользовательские архивы должны иметь хотя бы один общий фактор.



Archive	Архив
Data Fields	Поля данных
Customers	Клиенты
Cust. No.	№ клиента
Company	Компания
Address	Адрес
Phone No.	Тел.
Fax No.	Факс
Jobs	Задания
Article	Артикул
Quantity	Кол-во
Price	Цена
View	Представление
Orders	Заказы
Display in Runtime	Отображение в среде исполнения

В примере пользователь создает форму для заказов. Он получает необходимую информацию для заказов из архивов Customers (Заказчики) и Jobs (Задания). Номер заказчика является общим элементом двух архивов и используется в качестве связующего маркера для создания формы. Пользователь желает, чтобы в среде исполнения отображались только нужные поля подключенных пользовательских архивов.

Примечание

Программное обеспечение, обеспечивающее прямой ODBC доступ к пользовательским архивам версий старше 4.02, не сможет получить доступ к пользовательским архивам версии 4.02 или более поздних.

2.1.5 Набор функций пользовательских архивов

Следующие возможности пользовательских архивов приводятся ниже.

Конфигурация

- Пользовательские архивы и представления можно создавать в таблице, что обеспечит простое и прямое назначение данных полям пользовательских архивов или форм (деление на столбцы и ряды).
- Интерактивное отображение в среде исполнения в формате таблицы
- Ввод/вывод данных с помощью полей ввода-вывода (назначение полей пользовательского архива посредством C-макросов/управляющих тегов)

Регистрация

- Регистрация данных конфигурации и среды исполнения в таблице с помощью отчетов WinCC.
- Экспорт данных в формате CSV (обработка с помощью внешних программ, например Excel).

Передача из/в AS (S5, S7 и т. д.)

- Всех записей данных пользовательского архива (с помощью тегов необработанных данных).
- Отдельных полей данных записи данных (с помощью тегов WinCC).
- Связь через все интерфейсы, предоставляемые WinCC.

Возможности редактирования

- В представлениях таблицы.
- С помощью полей ввода-вывода (с C-макросами/управляющими тегами).

Управление

- В таблицах с помощью стандартных значков.
- С помощью C-макросов.

Удаление или создание новых записей данных

- В таблицах записи данных создаются с помощью кнопок.
- С помощью C-макросов создаются и удаляются записи данных.

Управляющие теги

- Мастер поддерживает создание тегов WinCC в качестве управляющих тегов.
- Возможности быстрого доступа к пользовательским архивам для сценариев и AS.
- Косвенная адресация для C-макросов.

Количественная структура

В редакторе User Archive (Пользовательский архив) можно настроить до 500 архивов и 500 представлений архивов. На один архив можно создать до 500 полей.

Архивы

Максимальное количество записей данных в архиве ограничено и зависит от числа сконфигурированных столбцов и записей данных в архиве. Произведение столбцов и записей данных не может быть более 320000. В число столбцов также следует включать ID (Идентификатор), созданный системой и, если таковые выбраны, Last User (Последний пользователь) и Last Access (Последний доступ).

Пример

В архиве сконфигурировано 15 столбцов и выбран столбец Last Access (Последний доступ). Таким образом, включая столбец ID (Идентификатор), сконфигурировано всего 17 столбцов, что составляет максимум $320000/17 = 18823$ записей данных.

Дополнительные источники информации

Области применения пользовательского архива (стр. 11)

2.2 Редактор User Archive (Пользовательский архив)

2.2.1 Структура редактора User Archive (Пользовательский архив)

Конфигурация

Для работы в редакторе User Archive (Пользовательский архив) можно использовать меню, панель инструментов, горячие клавиши или непосредственно мышь.

Ниже содержится информация по следующим темам:

- Меню редактора User Archive (Пользовательский архив)
- Панель инструментов редактора User Archive (Пользовательский архив)

2.2.2 Меню редактора User Archive (Пользовательский архив)

2.2.2.1 Меню редактора User Archive (Пользовательский архив)

Работа меню

Работа меню описывается в этом разделе. Функции, соответствующие стандарту Windows, в данном разделе не описываются.

В редакторе User Archive (Пользовательский архив) содержатся следующие меню.

Меню	Команда меню	Комбинация клавиш
Project (Проект)	Restore (Восстановить)	Ctrl + N
	Save (Сохранить)	Ctrl + S
	Export... (Экспорт...)	
	Import... (Импорт...)	
	Check... (Проверка...)	
	Exit (Выход)	
Edit (Правка)	Cut (Вырезать)	Ctrl + X
	Copy (Копировать)	Ctrl + C
	Paste (Вставить)	Ctrl + V
	Runtime Data (Данные среды исполнения)	Ctrl + R
	Options (Настройки)	Ctrl + O
	View (Вид)	Toolbar (Панель инструментов)
Status Bar (Строка состояния)		
Split (Разбить)		
Update (Обновить)		F 5
Runtime data (Данные среды исполнения)	Import (Импорт)	
	Export (Экспорт)	
Help (Справка)	Help Topics (Разделы справки)	
	Log-File... (Журнал...)	
	About... (О программе...)	

Примечание

Функции Cut (Вырезать), Paste (Вставить) и Copy (Копировать) можно использовать только в окне данных. Только один пользовательский архив, поле или форму можно вырезать, скопировать или вставить одновременно. Сохранение возможно только при отсутствии ссылки на пользовательский архив (происходит обращение к пользовательскому архиву, например, во время отображения окна таблицы в среде конфигурирования (CS) или среде исполнения).

2.2.2.2 Команда меню Restore (Восстановить)

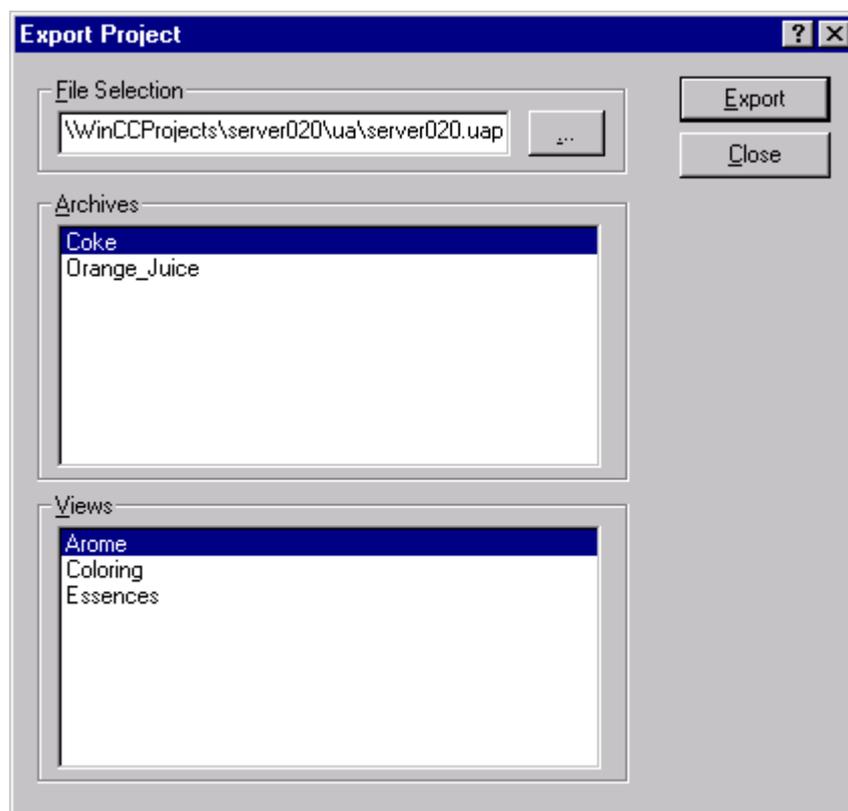
Restore (Восстановить)

С помощью команды Restore (Восстановить) можно отменить последние изменения и восстановить данные, сохраненные последними, не закрывая редактора. Кроме того, эту функцию можно использовать, чтобы принять произведенные и сохраненные изменения со времени открытия редактора сценариев или внешних программ. Подобные внешние изменения не выполняются с редактором автоматически.

2.2.2.3 Команда меню Export (Экспортировать) (меню Project (Проект))

Экспорт (меню проекта)

С помощью этой команды меню можно экспортировать пользовательский архив и представления (данные CS) открытого проекта WinCC.



При выборе команды Export (Экспортировать) меню Project (Проект) появится диалоговое окно, в котором можно выбирать между пользовательскими архивами и представлениями, созданными в открытом проекте WinCC. Можно выбирать как один, так и несколько объектов сразу. В поле выбора файла путь проекта запущенного проекта, имя файла имени проекта и расширение файла «uar» устанавливается автоматически. После нажатия на кнопку выбора файла откроется диалоговое окно, в котором можно установить свободное размещение данных. После выбора пути для размещения данных нажмите кнопку Export (Экспортировать), чтобы начать экспорт. Закройте диалоговое окно после экспорта выбранных пользовательских архивов и представлений.

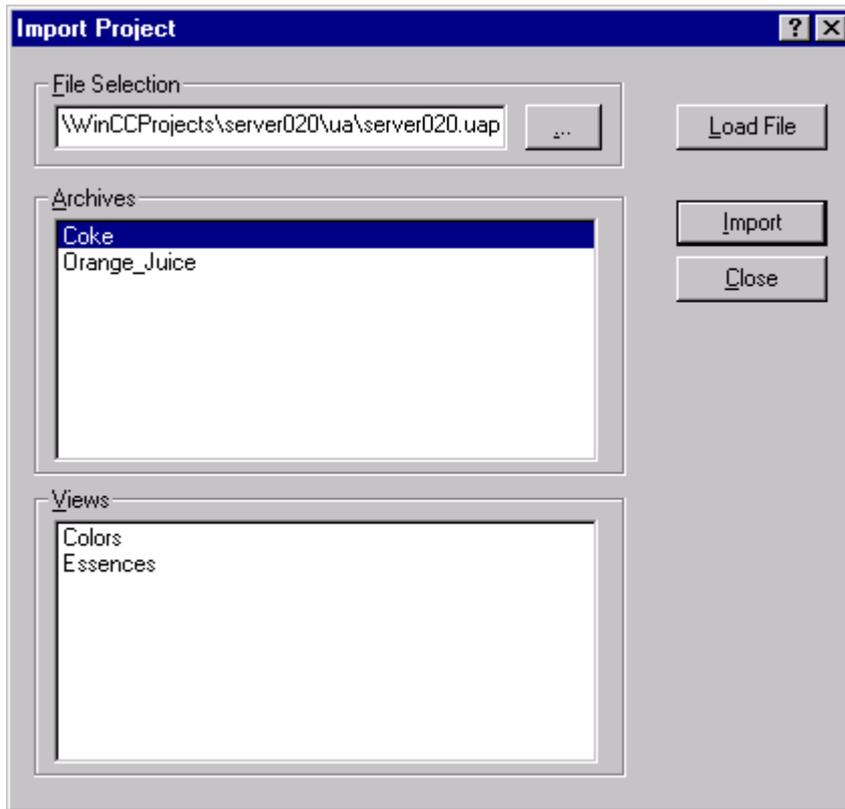
Примечание

Чтобы экспортировать данные среды исполнения, воспользуйтесь командой Export (Экспортировать) в меню Runtime data (Данные среды исполнения).

2.2.2.4 Команда меню Import (Импортировать) (меню Project (Проект))

Импорт (меню проекта)

С помощью этой команды меню можно импортировать пользовательский архив и представления (данные CS) открытого проекта WinCC.



Чтобы импортировать пользовательские архивы и представления, сначала необходимо экспортировать соответствующие пользовательские архивы и представления, чтобы создать файл uap. Для начала процедуры импорта нажмите в меню Project (Проект) на команде Import (Импортировать). Откроется диалоговое окно для выбора пользовательских архивов и представлений, которые нужно импортировать. Можно выбирать как один, так и несколько объектов сразу. В поле выбора файла путь проекта запущенного проекта и имя файла, состоящее из имени проекта и расширения файла «uap», устанавливается автоматически. При нажатии на кнопку для выбора файла откроется диалоговое окно, где можно выбрать файл, который хотите импортировать. После этого нажмите кнопку Import (Импортировать), чтобы начать импорт. Закройте диалоговое окно после импорта выбранных пользовательских архивов и представлений.

Чтобы сохранить целостность структуры данных, представление всегда экспортируется вместе со связанными архивами и импортируется вместе с этим представлением. Во время импорта не перезаписывайте архивы с одинаковыми именами. Чтобы оставить имя архива, удалите архивы с таким же именем из проекта до импорта. Необходимо предварительно сохранить существующие данные среды исполнения, т. к. они также удаляются при удалении архивов.

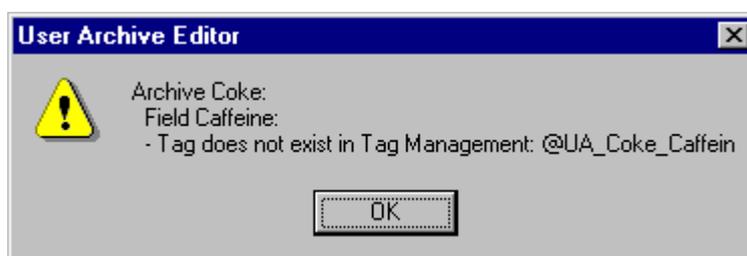
Примечание

Чтобы экспортировать данные среды исполнения, воспользуйтесь командой Import (Импорт) в меню Runtime data (Данные среды исполнения).

2.2.2.5 Команда меню Check (Проверить)

Проверка

С помощью этой команды меню можно проверить, существуют ли теги, на которые ссылались в редакторе User Archives (Пользовательские архивы), в проводнике WinCC. В случае обнаружения ошибки пользователь получит следующее сообщение: No errors were found during the check (При проверке не выявлено никаких ошибок). При возникновении ошибки пользователь получит следующее сообщение.



Отобразятся пользовательский архив, связанное с ним поле и отсутствующий тег в системе управления тегами.

Примечание

Эта функция не проверяет структурные теги.

2.2.2.6 Команда меню Runtime Data (Данные среды исполнения)

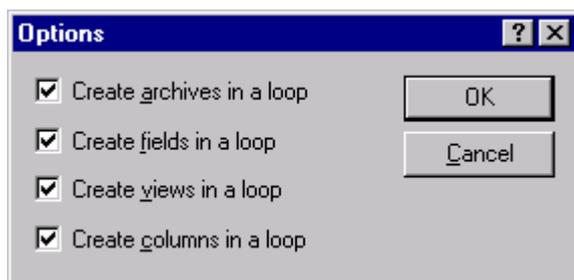
Данные среды исполнения

С помощью этой команды можно редактировать интерактивные данные в окне таблицы. Галочка в меню показывает, что команда Runtime Data (Данные среды исполнения) запущена.

2.2.2.7 Команда меню Options (Настройки)

Options (Настройки)

С помощью этой команды меню можно настроить процесс создания пользовательских архивов и изображений. После выбора этого меню появляется следующее диалоговое окно.



Create user archives in a loop: (Создавать пользовательские архивы циклически)

Если флажок установлен, автоматически появляется диалоговое окно для ввода будущих пользовательских архивов после ввода пользовательского архива и его полей.

Create fields in a loop: (Создавать поля циклически)

Если флажок установлен, автоматически появляется диалоговое окно для ввода будущих полей данных после ввода поля данных пользовательского архива.

Create views in a loop: (Создавать представления циклически)

Если флажок установлен, автоматически появляется диалоговое окно для ввода будущих представлений после введения представления и его столбцов.

Create columns of a view in loop: (Создавать столбцы представления циклически)

Если флажок установлен, автоматически появляется диалоговое окно для ввода будущих столбцов после введения столбцов.

2.2.2.8 Команда меню Split (Разбить)

Split (Разбить)

С помощью этой команды можно изменять размер трех разделительных окон в редакторе User Archives (Пользовательские архивы).

2.2.2.9 Команда меню Import (Импорт) (меню Runtime Data (Данные среды исполнения))

Импорт (меню данных среды исполнения)

С помощью этой команды можно импортировать записи (данные среды исполнения) в выбранный пользовательский архив.

В файле импорта отсутствует информация о типе данных и количестве столбцов. Поэтому структура импортируемых данных и целевого архива должна быть одинаковой или импорт производится в пользовательский архив, из которого данные были ранее экспортированы.

Во время экспорта идентификаторы записей вводятся в данные экспорта, чтобы упростить уникальное назначение импортируемых данных во время импорта. Если в WinCC во время импорта обнаружится, что один из импортируемых идентификаторов уже существует в пользовательском архиве, появится сообщение об ошибке, а в файле журнала UALogFile.txt создается запись, в которой будет указан данный идентификатор.

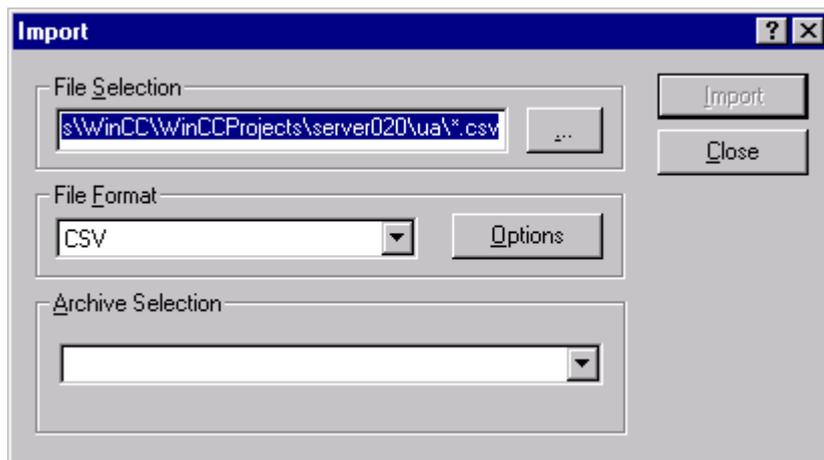
Данные с новым идентификатором записи добавляются в пользовательский архив в виде новой записи.

Примечание

При импортировании данных из текущего пользовательского архива, которые были отредактированы вне WinCC, необходимо переписать существующие данные архива, предварительно удалив в нем все записи. В противном случае во время импорта будут выдаваться сообщения об ошибках из-за одинаковых идентификаторов записей.

Чтобы импортировать структуры для пользовательского архива и представления, необходимо использовать команду Import (Импорт) в меню Project (Проект).

Эта команда не работает, если активирована функция Runtime Data (Данные среды исполнения) (меню Edit (Правка)).



В поле File Selection (Выбор файла) введите путь и информацию импортируемого файла пользовательского архива. Кнопка «...» поможет во время выбора файла. Путь файла задается автоматически в папку «ua» в пути проекта активного пользовательского архива.

В поле File format (Формат файла) можно выбрать формат файла пользовательского архива, из которого будет считываться информация. С помощью кнопки Options (Параметры) задайте требуемый разделительный знак. Разделителем, установленным по умолчанию, является точка с запятой «;».

В поле Archive Selection (Выбор архива) выберите один из пользовательских архивов текущего проекта в качестве целевого архива. После этого станет доступна кнопка Import (Импорт).

После нажатия кнопки Import (Импорт) будет произведен импорт.

Примечание

В проекте клиент/сервер необходимо учитывать следующее, если на сервере находится пользовательский архив, например по адресу «с:\Projects\Test\UA», доступ к нему предоставляется по этому пути. Клиент получает доступ посредством сетевого диска, например «l:\Test\UA». После этого стандартным путем пользовательского архива на клиенте будет «l:\Test\UA». Тем не менее, на сервере не существует такого каталога. Чтобы выполнить импорт или экспорт данных пользовательского архива на клиента, необходимо изменить стандартный путь на клиенте – в этом примере требуется указать путь «C:\Projects\Test\UA».

2.2.2.10 Команда меню Export (Экспорт) (меню Runtime Data (Данные среды исполнения))

Экспорт (меню данных среды исполнения)

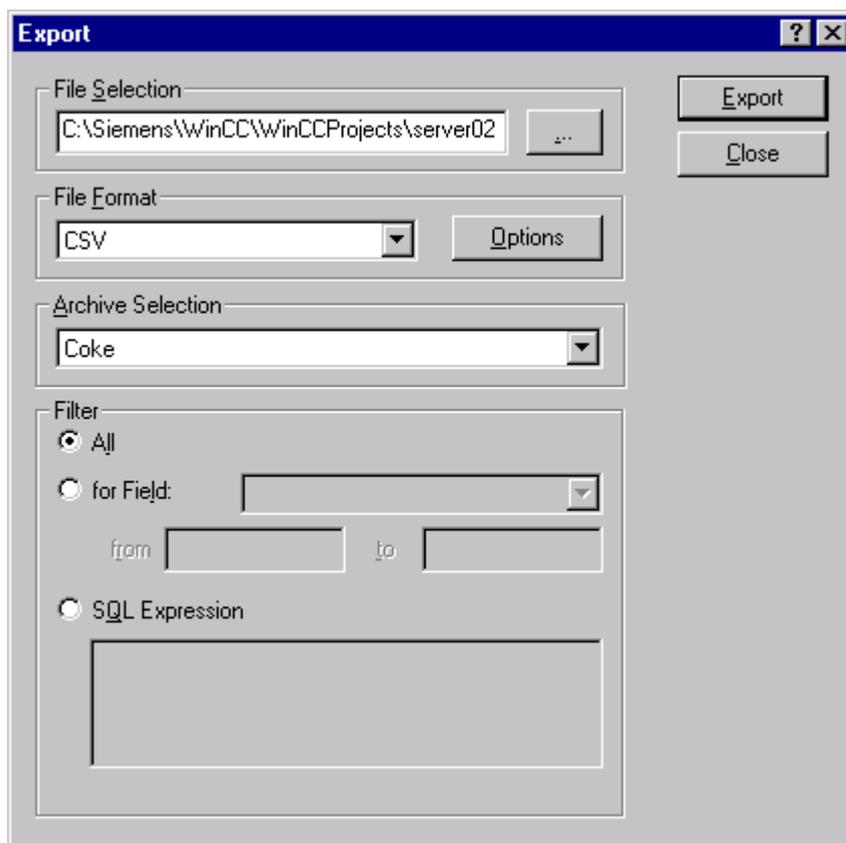
С помощью этой команды меню можно экспортировать записи данных (данные среды исполнения) выбранных пользовательских архивов. Экспортированные данные можно редактировать в другой программе пользователя, например MS-Excel, а затем импортировать в пользовательский архив.

Примечание

Эта команда не работает, если активирована функция Runtime Data (Данные среды исполнения) (меню Edit (Правка)).

Чтобы экспортировать структуры пользовательского архива и представления, необходимо использовать команду Export (Экспорт) в меню Project (Проект).

После версии WinCC V5.1 также экспортируются заголовки столбцов, которые нельзя изменять.



В поле File Selection (Выбор файла) введите путь и информацию экспортируемого файла пользовательского архива. Кнопка «...» поможет во время выбора файла. Путь файла задается автоматически в папку «ua» в пути проекта активного пользовательского архива.

В поле File format (Формат файла) можно выбрать формат экспортируемого файла пользовательского архива. С помощью кнопки Options (Параметры) задайте требуемый разделительный знак. Разделителем, установленным по умолчанию, является точка с запятой «;».

В поле Archive Selection (Выбор архива) выберите один из пользовательских архивов текущего проекта в качестве целевого архива. После этого станет доступна кнопка Export (Экспорт).

В поле Filter (Фильтр) можно при необходимости указать фильтр для экспорта пользовательского архива. В поле Filter on field (Фильтр по полю) укажите, к какому полю относится фильтр. В полях «from... to» (от... до) задайте диапазон значений, которые пройдут через фильтр.

После того как станет доступна кнопка SQL Statement (Оператор SQL), в поле ввода можно ввести оператор фильтрации на языке SQL. Дополнительную информацию об операторах SQL см. в приложении.

После нажатия кнопки Export (Экспорт) будет произведен экспорт.

Во время экспорта идентификаторы записей вводятся в экспортируемый файл, чтобы упростить уникальное назначение импортируемых данных во время импорта.

Примечание

В проекте клиент/сервер необходимо учитывать следующее, если на сервере находится пользовательский архив, например по адресу «с:\Projects\Test\UA», доступ к нему предоставляется по этому пути. Клиент получает доступ посредством сетевого диска, например «l:\Test\UA». После этого стандартным путем к пользовательскому архиву на клиенте будет «l:\Test\UA». Тем не менее, на сервере не существует такого каталога. Чтобы выполнить экспорт данных пользовательского архива, необходимо изменить стандартный путь на клиенте – в этом примере требуется указать путь «C:\Projects\Test\UA».

2.2.3 Панель инструментов редактора User Archive (Пользовательский архив)

Панели инструментов

В редакторе User Archive (Пользовательский архив) можно работать, щелкая кнопкой мыши на панели инструментов. Отдельные символы на панели инструментов описаны ниже в алфавитном порядке.

Символ	Описание
	Browse (Обзор)
	Properties (Свойства)
	Export (Экспорт)
	Help (Справка)
	Import (Импорт)
	Delete (Удалить)
	New (Создать)
	Runtime Data (Данные среды исполнения)
	Save (Сохранить)
	Restore (Восстановить)

Browse (Обзор)

Символы Browse (Обзор) обеспечивают удобный просмотр пользовательского архива в среде исполнения.

Properties (Свойства)

Используйте символ Property (Свойство) для редактирования свойств пользовательских архивов или полей данных. Щелкнув правой кнопкой мыши на пользовательском архиве или поле данных, можно выполнить редактирование свойств пользовательского архива или поля данных.

В среде исполнения можно перемещаться по таблице только при активированном символе Properties (Свойства) во время редактирования полей данных с помощью клавиш управления курсором. Поля в таблице можно редактировать сразу. Кнопка Properties (Свойства) станет доступна только после того, как выбрано поле данных.

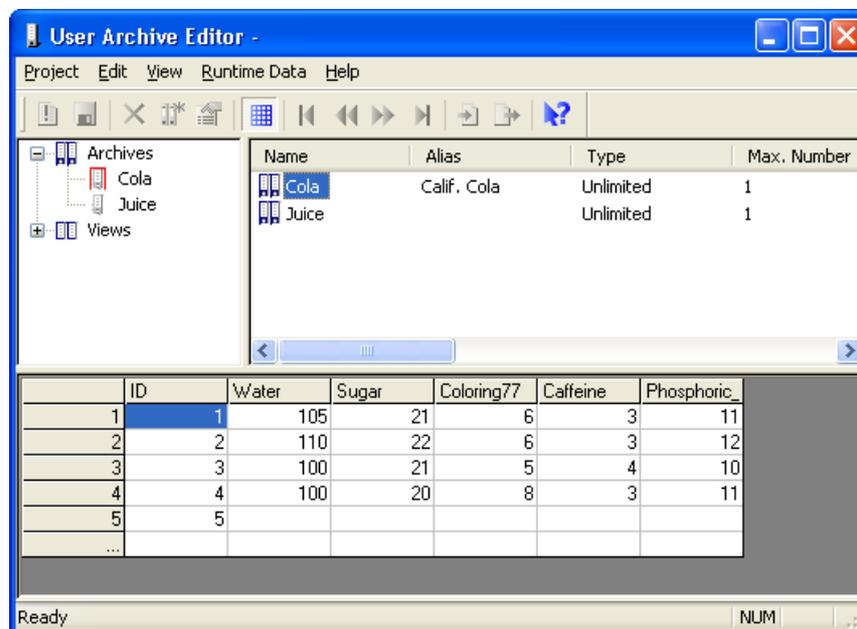
Delete (Удалить)

Используйте символ Delete (Удалить) для удаления пользовательских архивов или полей данных. Щелкнув правой кнопкой мыши на пользовательском архиве или поле данных, можно выполнить удаление архива или поля данных. Кроме того, также можно сначала выбрать пользовательский архив или поле данных, а затем нажать клавишу Delete (Удалить).

New (Создать)

Используйте символ New (Создать) для создания новых пользовательских архивов или полей данных. Создать новый пользовательский архив можно, щелкнув правой кнопкой мыши на одном из верхних окон.

2.2.4 Окно таблицы редактора User Archive (Пользовательский архив)



Включить или отключить таблицу полей можно с помощью пункта меню Edit – Runtime Data (Правка – Данные среды исполнения) или нажатием соответствующей кнопки. Дважды щелкните одно из полей, чтобы разрешить ввод данных. Маркировка производится с помощью текстового курсора. В среде исполнения можно перемещаться по таблице только при активированном символе Properties (Свойства) во время редактирования полей данных с помощью клавиш управления курсором. Поля в таблице можно редактировать сразу. Кнопка Properties (Свойства) станет доступна только после того, как выбрано поле данных. Функции редактирования также могут использоваться в поле таблицы с помощью контекстного меню. Чтобы скопировать записи данных во внешнюю программу, выберите требуемые строки таблицы и скопируйте их в буфер обмена с помощью комбинации клавиш Ctrl + C. Вставить строки во внешнюю программу можно с помощью горячей клавиши Ctrl + V. Вставить внешние данные подобным образом в окно таблицы редактора User Archive (Пользовательский архив) невозможно.

Примечание

При изменении одного или нескольких значений в таблице полей редактора User Archive (Пользовательский архив) или в таблице UserArchiveControl необходимо выйти из записи данных (выбрав другую ячейку или ряд таблицы) после создания записи, чтобы значение было занесено в базу данных и выведено на всех отображениях.

2.3 Конфигурация

2.3.1 Конфигурация

На первом этапе осуществляется конфигурация пользовательского архива. Для этого можно использовать мастера, которые предоставляют удобный способ конфигурации, управляемый пользователем. Необходимо выполнить следующие этапы конфигурации.

Конфигурация пользовательских архивов

- Создание пользовательских архивов
- Настройка полей пользовательского архива

Конфигурация представлений

- Создание представления
- Настройка полей данных
- Настройка связей

Конфигурация элемента управления WinCC UserArchiveControl

- Создание элемента управления UserArchiveControl

Дополнительные источники информации

Изменение конфигурации пользовательских архивов (стр. 58)

2.3.2 Конфигурация пользовательского архива

2.3.2.1 Создание нового пользовательского архива

Изменение конфигурации пользовательских архивов

Создание нового пользовательского архива

Запустите редактор User Archives (Пользовательские архивы) из проводника WinCC.

- Для этого нажмите пользовательские архивы и в контекстном меню выберите пункт Open (Открыть). На экране появится раздел команд редактора User Archives (Пользовательские архивы).

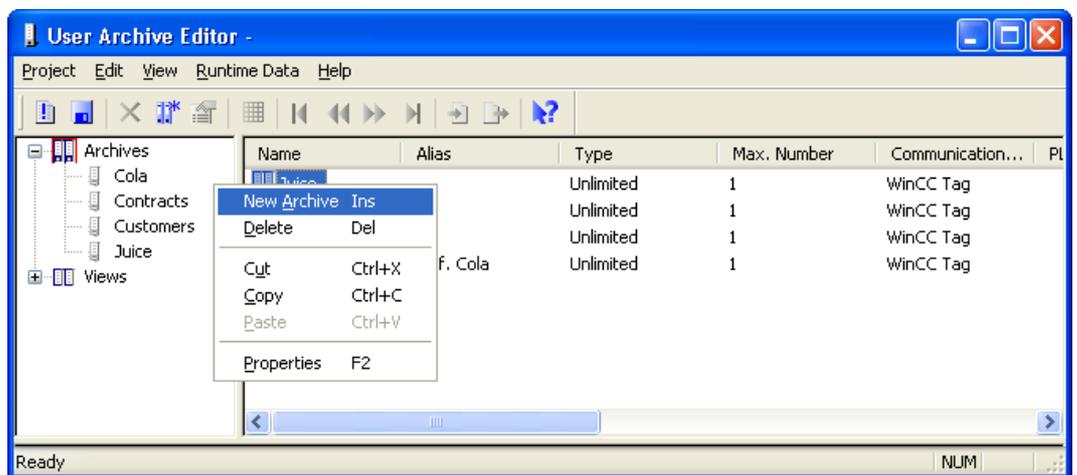
Примечание

На один пользовательский архив можно создать 500 полей.

Параметр по умолчанию Put archive in cycle (Поместить архив в цикл) позволяет один за другим создавать несколько архивов. Чтобы создать только один пользовательский архив, можно отключить этот параметр в меню Edit - Options (Правка – Параметры).

Выполните следующие действия для создания пользовательского архива.

1. В окне навигации щелкните левой кнопкой мыши на Archives (Архивы).
2. Затем щелкните в окне навигации или данных правой кнопкой мыши. Появится контекстное меню, показанное на изображении ниже.



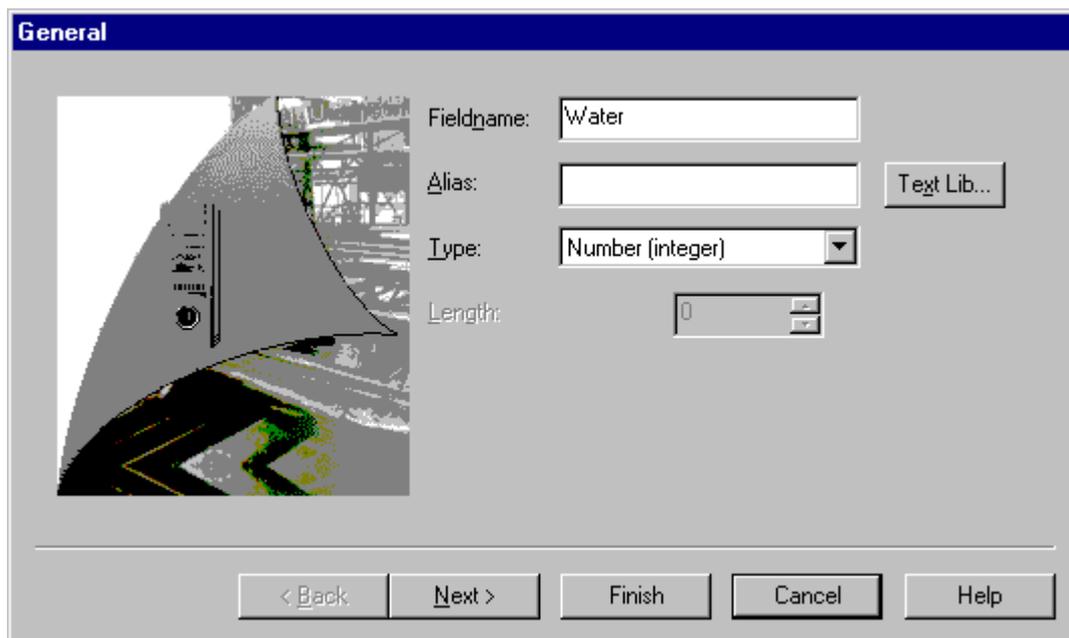
3. Выберите параметр New Archive (Новый архив).

Появится мастер для конфигурации пользовательского архива.

2.3.2.2 Основные свойства полей пользовательского архива

Настройка основных свойств полей пользовательского архива

В диалоговом окне General (Общие) пользователь определяет поле пользовательского архива, которое необходимо создать, а также тип поля. Для типа поля String можно задать дополнительно длину поля.



- В поле ввода Field name (Имя поля) можно ввести имя для первого поля пользовательского архива; в примере таким именем является первый компонент рецепта Water.
- В поле Alias (Заголовок) можно присвоить полю второе имя, чтобы дать полю комментарий или иметь возможность переключения языка в среде исполнения с помощью текстовой библиотеки. Ввод этих данных является необязательным.
 - Для получения дополнительной информации о переключении языка см. интерактивную справку.

Введенные имена служат для дальнейшего назначения полей для представления таблицы.

В поле ввода Type (Тип) можно указать один из следующих типов тегов:

- Integer — 32-битовое число со знаком
- Float — 32-битовое число с плавающей точкой IEEE 754
- Double — 64-битовое число с плавающей точкой IEEE 754
- string — текстовый тег 8-битной кодировки
- Date/time — нет специального типа данных

Примечание

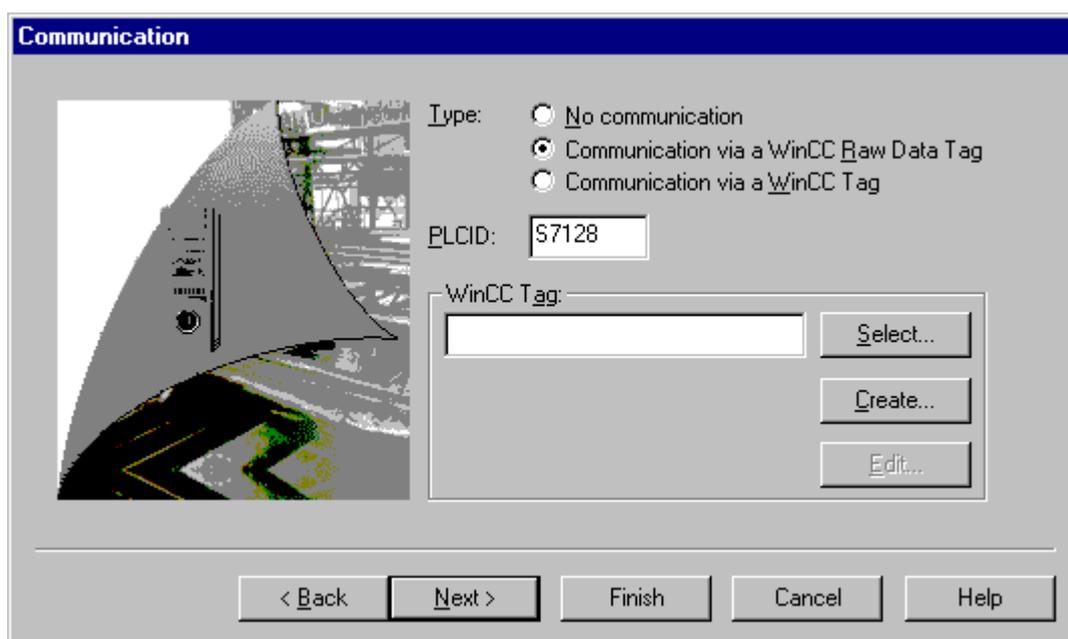
Для тегов типа Date/Time (Дата/время) формат ввода для даты и времени зависит от настроек операционной системы.

Дополнительные источники информации

Изменение языка (стр. 71)

2.3.2.3 Настройки связи**Настройки связи**

В диалоговом окне Communication (Связь) можно установить связь между ПЛК и пользовательским архивом.



В пункте Type (Тип) можно задать тип связи:

- **none (отсутствует):** связь невозможна;
- **via raw data tag (с помощью тега необработанных данных):** доступ к AS с помощью тега необработанных данных;
- **via WinCC tag (с помощью тега WinCC):** доступ к AS с помощью тегов WinCC.

Чтобы получить связь с помощью необработанных тегов данных, выберите Through Raw Data Tags (С помощью тегов необработанных данных). Введите PLCID в качестве идентификации архива. PLCID состоит из 8 символов ASCII и является уникальным в рамках проекта WinCC. Он определяет соответствующий пользовательский архив и

является необходимым условием для возврата данных отображения процесса AS в надлежащий пользовательский архив.

Выбрав through raw data tag (с помощью тега необработанных данных), нажмите Select (Выбрать) и затем выберите тег необработанных данных.

При выборе связи с помощью тега WinCC теги назначаются в диалоговом окне свойств полей пользовательских архивов.

Примечание

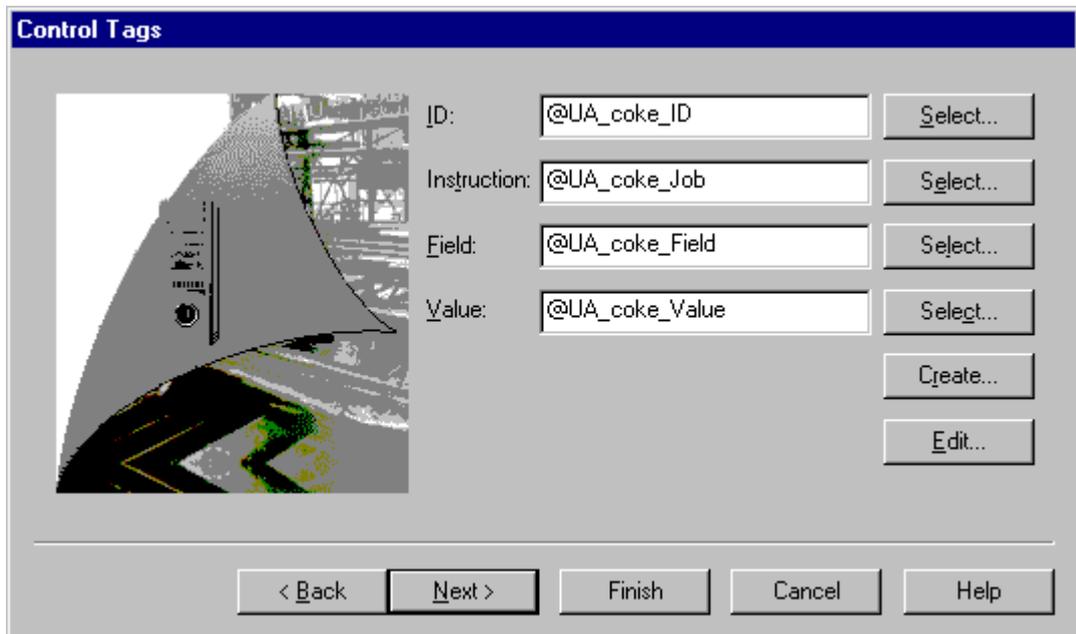
При связи с помощью тегов необработанных данных вся запись соединяется с тегом необработанных данных. При использовании тега WinCC тег соединяется с полем пользовательского архива.

Для связи пользовательского архива с тегом необработанных данных PLCID используется в качестве уникального имени архива. R_ID, сконфигурированный в используемом теге необработанных данных, не может применяться, т. к. он имеет отношение только для связи с AS. Несколько архивов пользователя могут получать данные с помощью одного тега необработанных данных.

2.3.2.4 Настройка управляющих тегов

Настройка управляющих тегов

На вкладке Control Tags (Управляющие теги) управляющие теги настраиваются в форме тегов WinCC, которые можно использовать для доступа к полю пользовательского архива.



В четырех полях ввода вкладки устанавливаются теги WinCC, которые можно использовать для доступа к идентификаторам записи данных, кодам заданий, полям архива и значениям поля архива.

Рядом с каждым полем ввода находится кнопка Select (Выбрать), с помощью которой можно открыть диалоговое окно выбора. Здесь можно отобразить и выбрать все существующие теги WinCC.

Воспользуйтесь кнопкой Create... (Создать...), чтобы создавать теги автоматически. Таким же образом можно создать новую группу тегов @UA[Имя архива] и сохранить созданные теги в виде @UA[Имя архива]ID, @UA[Имя архива]Job и т. д.

Примечание

Чтобы проверить функцию, необходимо всегда указывать или создавать все четыре управляющих тега пользовательского архива с помощью мастера.

Запрещается изменять тип данных управляющих тегов.

При создании пользовательского архива рекомендуется использовать мастер для создания и сохранения управляющих тегов.

С помощью четырех управляющих тегов можно инициировать пользовательский архив. Для этого необходимо задать соответствующие значения либо тегам ID (Идентификатор) и Job (Задание), либо тегам Job (Задание), Field (Поле) и Value (Значение).

Подробную информацию о ПЛК и пользовательском архиве можно найти в подразделе «Управляющие теги» в главе «Свойства управляющих тегов».

Если не хотите использовать управляющие теги, можно выйти из диалогового окна без ввода данных. Пример использования управляющих тегов см. в главе «Пример использования управляющих тегов».

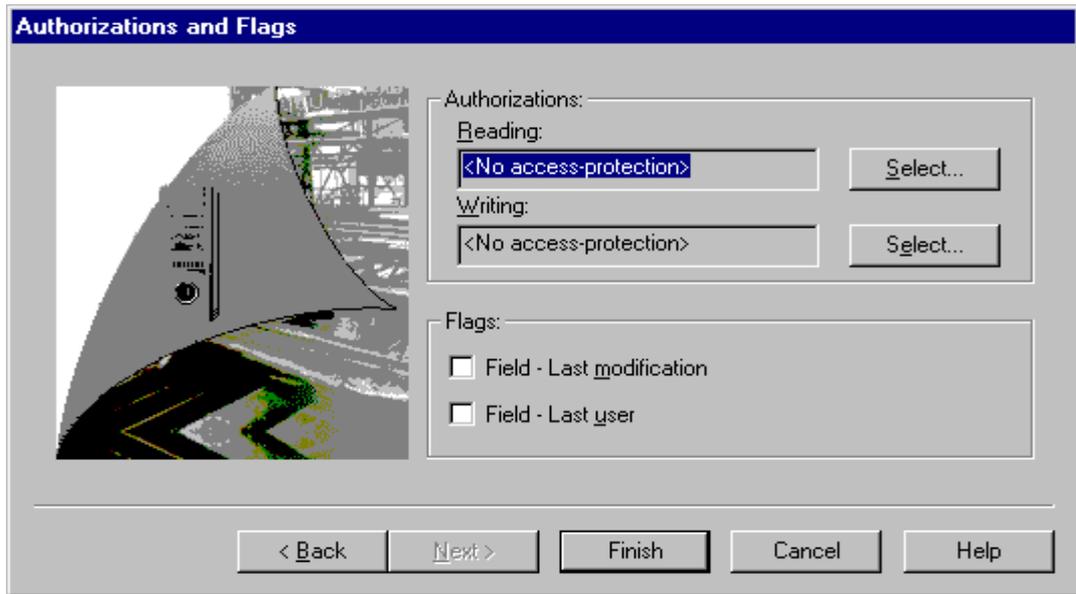
Дополнительные источники информации

Пример использования управляющих тегов (стр. 51)

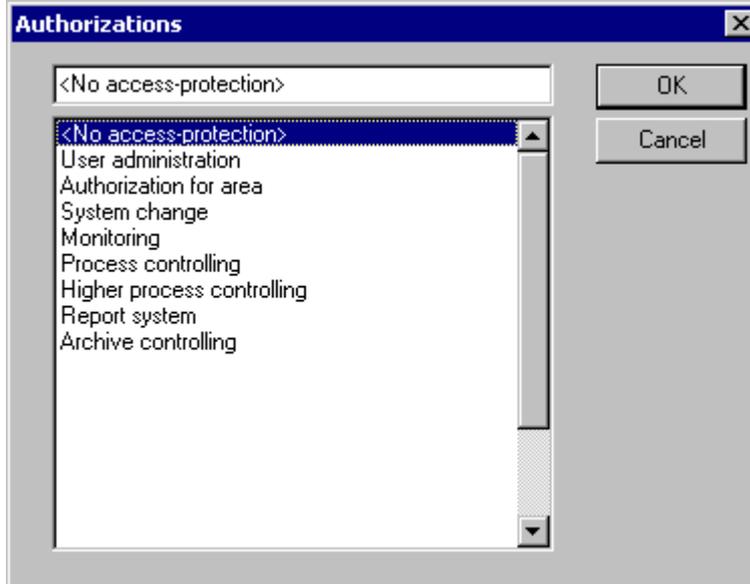
2.3.2.5 Настройка разрешений и флажков

Настройка разрешений и флажков

С помощью диалогового окна Authorizations and Flags (Разрешения и флажки) можно устанавливать права доступа к пользовательским архивам и задать вывод данных о последнем доступе/пользователе в отдельных столбцах.



Отображаются все заданные на данный момент разрешения на чтение и запись. Чтобы изменить эти настройки, можно нажать на одну из кнопок Select (Выбрать). Далее появится диалоговое окно Authorizations (Разрешения), в котором можно выбирать между разрешениями, установленными в редакторе User Administrator (Администратор пользователей):



При активировании параметра Field – Last modification (Поле – Последнее изменение) в столбце отображается дата и время последнего доступа. С помощью параметра Field – Last user (Поле – Последний пользователь) можно просмотреть пользователя, получавшего доступ последним.

1. Выбор разрешения
2. Выберите, например, поле Last User (Последний пользователь).

3. Выйдите из создания пользовательского архива, нажав кнопку Finish (Готово).

После создания пользовательского архива появится запрос Add fields? (Добавить поля?). При нажатии кнопки Yes (Да) откроется диалоговое окно General (Общие) для создания полей пользовательского архива. Для получения информации о создании полей пользовательского архива см. главу «Создание полей пользовательского архива».

При установке флажка Create archives in cycle (Создавать архивы циклически) в меню Edit – Options (Правка – Параметры) после создания полей появится вопрос Next Archive? (Следующий архив?). При выборе кнопки Yes (Да) появится первоначальное диалоговое окно General (Общие) для ввода следующего пользовательского архива.



В конце сохраните новый пользовательский архив, щелкнув кнопкой мыши на символе Save (Сохранить) или в меню Project – Save (Проект – Сохранить).

Примечание

Изменения в пользовательских архивах в базе данных принимаются после нажатия кнопки Save (Сохранить). Если пользовательский архив будет синхронизироваться с помощью опции Redundancy (Резервирование), необходимо установить флажок Last Access (Последний доступ).

Пользовательский архив Cola в примере обладает следующими свойствами.

Пользовательские архивы	Свойства
Cola	Имя: Cola
	Заголовок: Calif Cola
	Тип: неограниченный
	Макс. число записей: 1
	Тип связи: необработанный
	PLCID: S7112
	Имя переменной: CalifVarGroup
	Право на чтение: 0
	Право на запись: 0
	Флажки: U
	Пол.: 3

Последний доступ: 03/05/98 12:54

Дополнительные источники информации

Создание новых полей пользовательского архива (стр. 39)

2.3.2.6 Создание новых полей пользовательского архива

В этом разделе описывается создание полей данных пользовательского архива.

Параметр по умолчанию Put archive in cycle (Копировать архив циклически) позволяет один за другим создавать множество полей. Чтобы создать одно поле, можно отключить параметр в меню Edit – Options (Правка – Параметры).

1. Разверните раздел Archive (Архив) в окне навигации (нажмите на символ «+»). В окне навигации отобразится новый пользовательский архив Cola.
2. В окне навигации щелкните левой кнопкой мыши пользовательский архив Cola. The following pop-up menu is displayed:

<u>N</u> ew Field	Ins
<u>D</u> elete	Del
<u>R</u> estore RT Data	Ctrl + Y
<u>P</u> roperties	F2

3. Выберите пункт меню New Field (Новое поле).

Откроется диалоговое окно General (Общие).

Примечание

При изменении полей пользовательского архива может произойти потеря данных в следующих случаях.

Если для существующих данных новое условие целостности больше не может выполняться как, например для Unique, Not Null и т. д.

Если поле было переименовано.

Если новый тип данных больше не может преобразовать данные из источника.

Подробную информацию об изменении пользовательских архивов см. в главе «Изменение конфигурации пользовательских архивов».

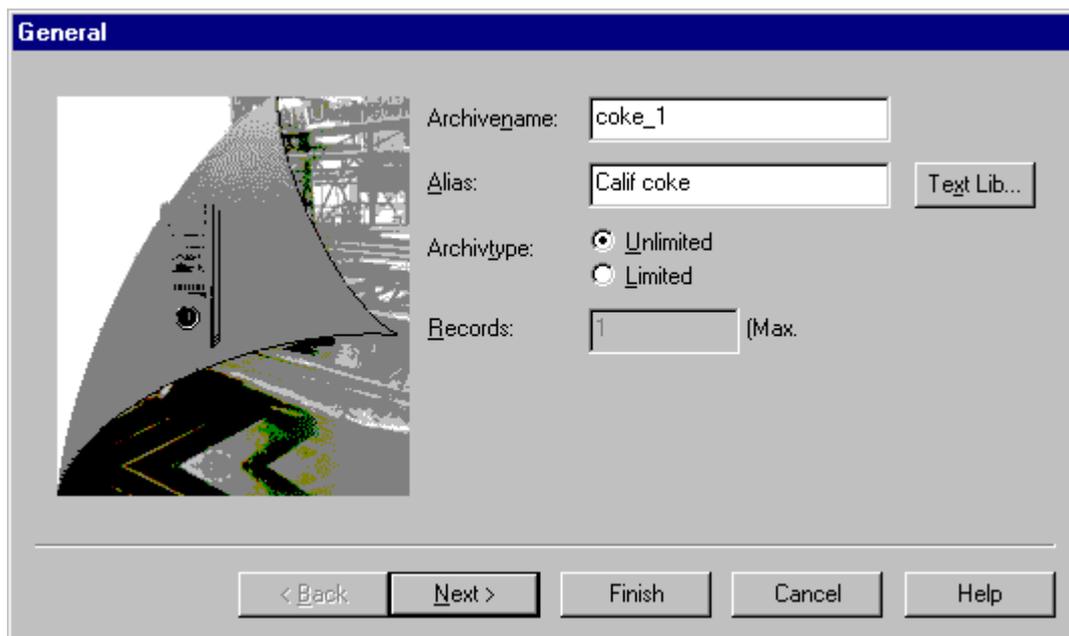
Дополнительные источники информации

Конфигурация изменений пользовательских архивов (стр. 58)

2.3.2.7 Основные свойства пользовательских архивов

Настройка основных свойств пользовательских архивов

В диалоговом окне General (Общие) можно создать новый пользовательский архив.



Введите имя пользовательского архива, например, Cola. В поле Alias (Заголовок) можно присвоить пользовательскому архиву второе имя, например, Calif Cola, чтобы дать пользовательскому архиву комментарий или иметь возможность переключения языка с помощью текстовой библиотеки в среде исполнения. Ввод этих данных является необязательным.

Для получения дополнительной информации об изменении языка см. интерактивную справку.

При выборе типа архива Limited (Ограниченный) можно задать максимальное число записей данных в поле Number (Число). При выборе типа архива Unlimited (Неограниченный) создаются пользовательские архивы с неограниченным числом записей данных.

Примечание

Ключевые (или зарезервированные) слова языка базы данных SQL нельзя использовать в именах архивов или полей. Более подробную информацию см. в главе «Алфавитный список ключевых слов SQL».

При создании записей данных не будет выполняться проверка на их завершенность или правильность.

Дополнительные источники информации

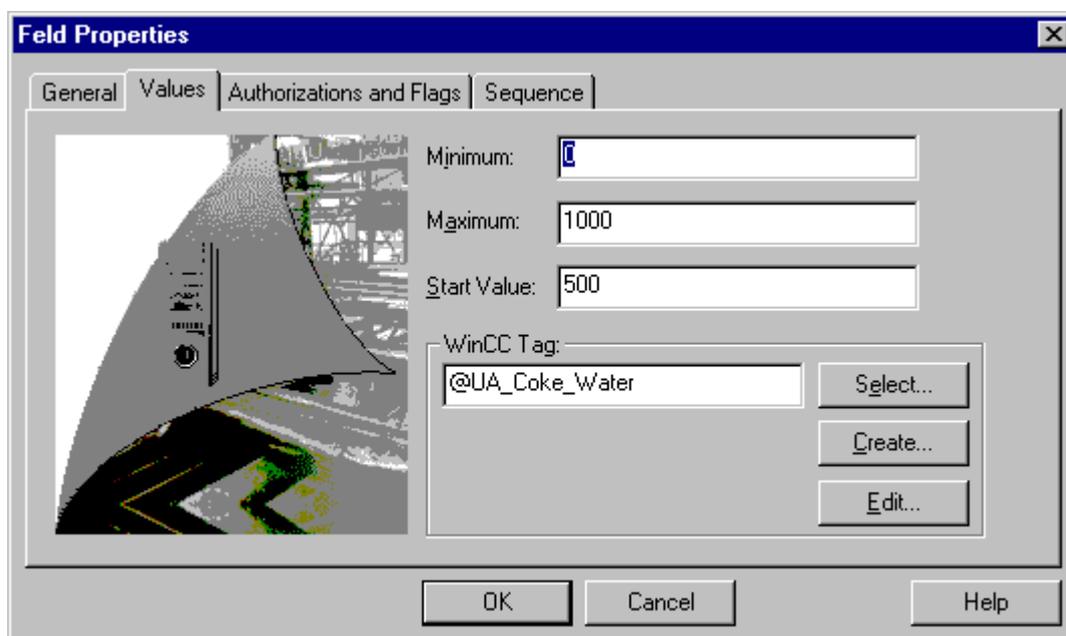
Изменение языка (стр. 71)

Алфавитный список ключевых слов SQL (стр. 82)

2.3.2.8 Настройка значений полей пользовательского архива

Настройка значений

В диалоговом окне Values (Значения) можно ввести минимальное, максимальное или начальное значение.



Минимальное, максимальное и начальное значение

Здесь можно установить соответствующие значения в зависимости от выбранного типа данных.

Тег WinCC

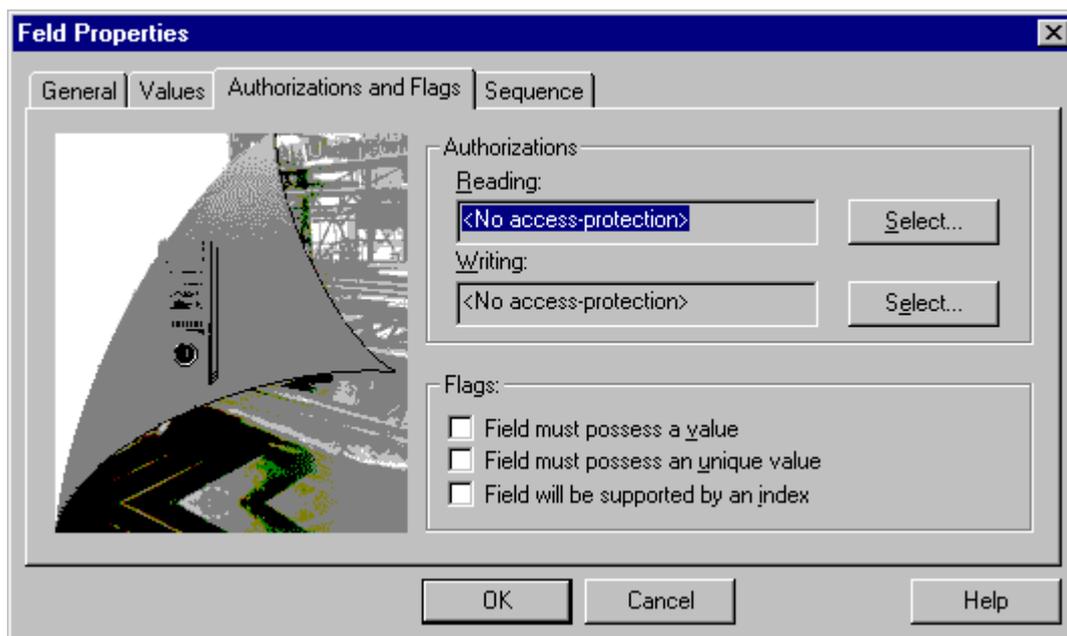
Здесь можно установить тег WinCC, который будет хранить значение поля пользовательского архива. Можно выполнить следующие действия...

1. ввести тег в поле ввода напрямую;
2. выбрать или создать тег интерактивно с помощью кнопки Select (Выбрать);
3. автоматически создать новый тег с помощью кнопки Create (Создать);
4. изменить свойства тегов с помощью кнопки Edit (Редактировать).

2.3.2.9 Настройки Authorizations and Flags (Разрешения и флажки)

Настройка разрешений и флажков

В диалоговом окне Authorizations and Flags (Разрешения и флажки) можно установить права доступа и атрибуты для полей пользовательских архивов.



Права

С помощью кнопки Select (Выбрать) можно установить права доступа на чтение и запись. Возможные разрешения устанавливаются в редакторе User Administrator (Администратор пользователей). Настройка прав доступа осуществляется как в разделе Create a User Archive (Создание пользовательского архива).

Флажки

В области Flags (Флажки) можно задать следующие атрибуты для выбранного поля данных.

1. Field must possess a value (Запрет пустых значений поля):
Значение поля должно быть отличным от нуля.
2. Field must possess an unique value (Уникальное значение поля):
У поля должно быть уникальное значение, т.е. значения в столбце должны отличаться друг от друга.
3. Field should be supported by an index (Индексируемое поле):

Поле поддерживает индексированное значение, если это возможно. Индексирование увеличивает эффективность команд поиска.

4. Выйдите из поля ввода данных, нажав кнопку Finish (Готово).

Теперь в пользовательском архиве создается новое поле данных Cola.

После установки флажка Create fields in cycle (Создавать поля циклически) в меню Edit – Options (Правка – Параметры) после создания полей появится вопрос Next Archive? (Следующий архив?). При выборе кнопки Yes (Да) появится первоначальное диалоговое окно General (Общие) для ввода следующего поля.

5. Сохранение нового пользовательского архива

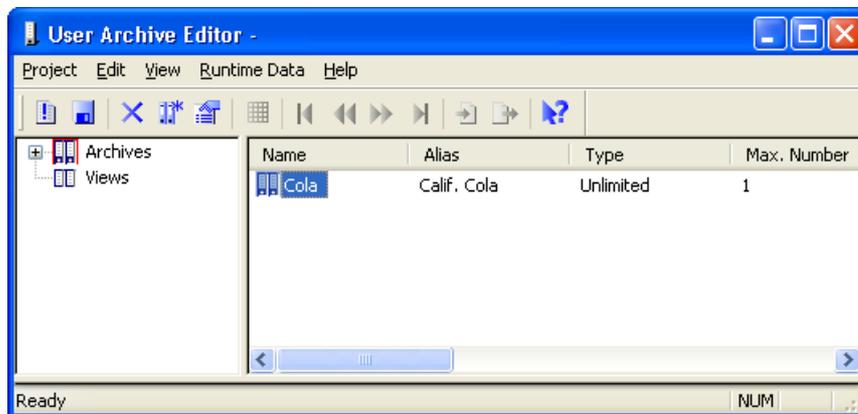
Примечание

Изменения в пользовательских архивах в базе данных принимаются после нажатия кнопки Save (Сохранить).

2.3.2.10 Свойства пользовательских архивов

Редактирование свойств формы...

- В окне навигации щелкните правой кнопкой мыши на пользовательском архиве Cola (сначала раскройте архив).
- В контекстном меню выберите Properties (Свойства).

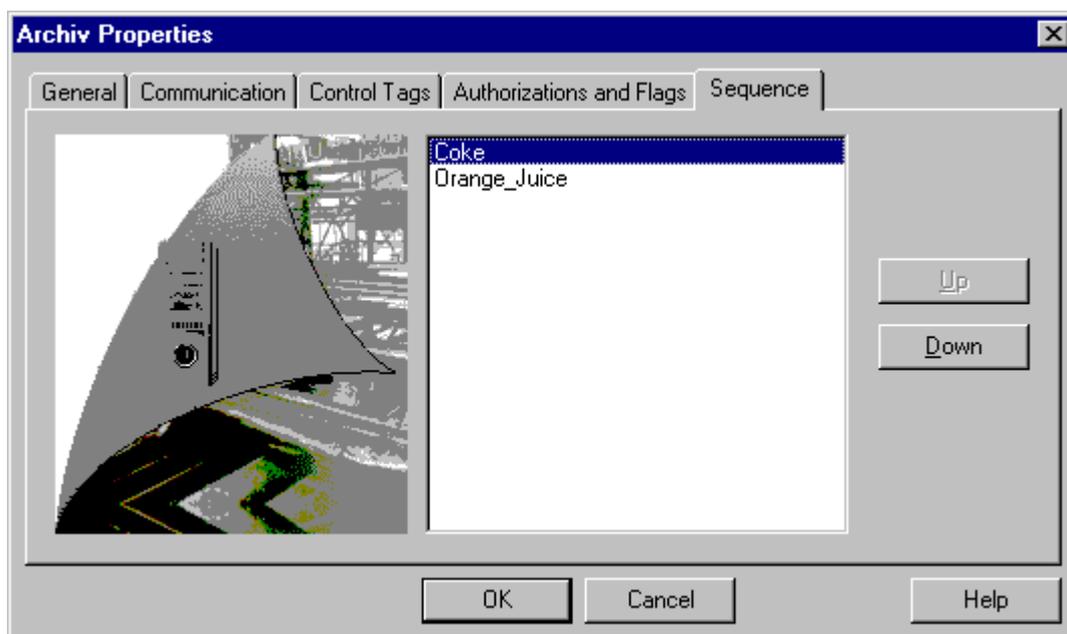


Откроется диалоговое окно Archive Properties (Свойства архива), в котором можно изменить свойства. Описание вкладок General (Общие), Communication (Связь), Flags (Флажки) и Select Authorization (Выбор разрешения) приведено в главе «Создание пользовательского архива». Дополнительная вкладка Order (Порядок) используется для настройки порядка пользовательских архивов.

2.3.2.11 Пользовательские архивы: вкладка Sequence (Последовательность)

Вкладка Sequence (Последовательность)

Вкладка Order (Порядок) используется для настройки порядка пользовательских архивов.



Выберите один или несколько пользовательских архивов и измените их положение в таблице с помощью клавиш Up (Вверх) и Down (Вниз). Подтвердите внесенные значения, нажав OK. В конце сохраните пользовательский архив, щелкнув кнопкой мыши на символе Save (Сохранить) или в меню Project – Save (Проект – Сохранить). Порядок следования пользовательских архивов отобразится в редакторе User Archives (Пользовательские архивы) в столбце Pos. (Поз.).

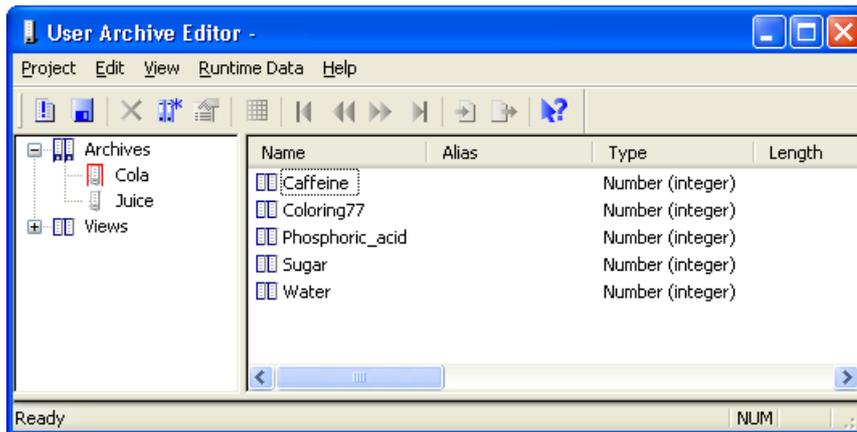
Примечание

Изменения в пользовательских архивах в базе данных принимаются после нажатия кнопки Save (Сохранить).

2.3.2.12 Свойства полей пользовательского архива

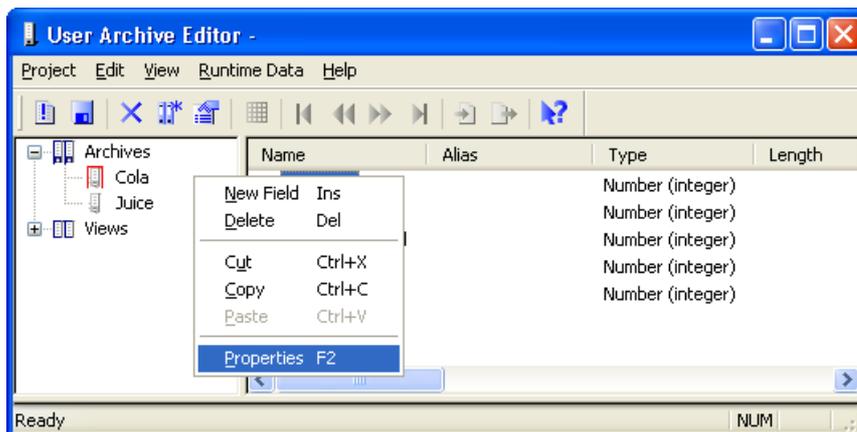
Редактирование свойств полей данных...

- Выберите окно навигации пользовательского архива Cola (сначала раскройте архив).
- В окне данных редактора User Archives (Пользовательские архивы) теперь можно видеть поля данных пользовательского архива Cola.



Редактирование полей данных пользовательского архива...

- Нажмите имя поля Water в окне данных редактора User Archives (Пользовательские архивы).
- В контекстном меню нажмите кнопку Properties (Свойства).



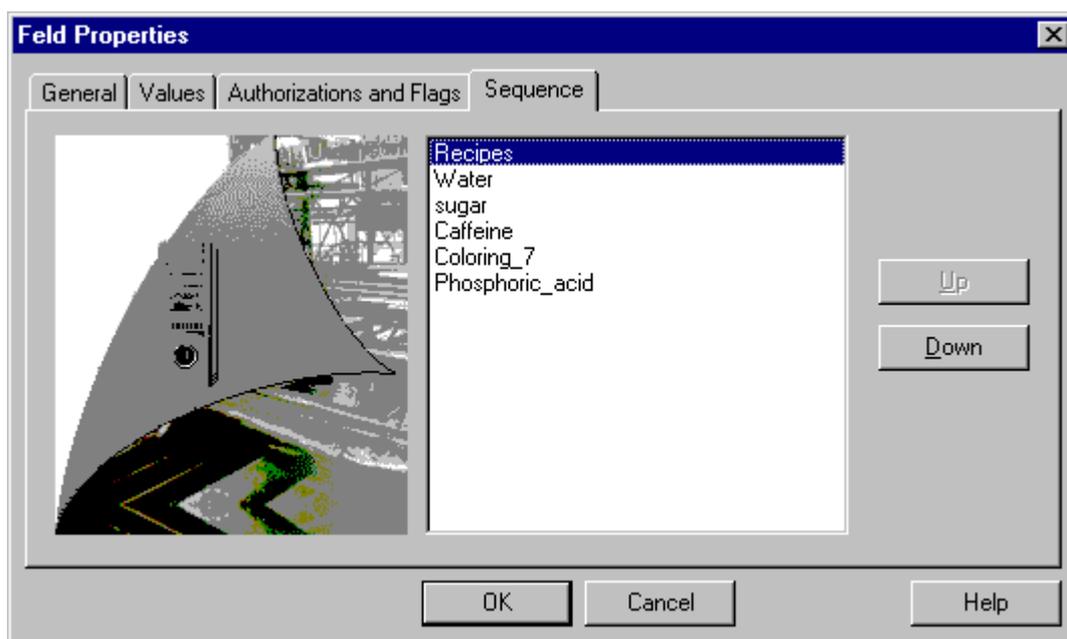
Откроется диалоговое окно Field Properties (Свойства поля), в котором можно изменить свойства поля данных.

Описание вкладок General (Общие), Values (Значения), Flags (Флажки) и Select Authorization (Выбор разрешения) приведено в главе «Определение полей пользовательского архива». Дополнительная вкладка Order (Порядок) используется для настройки последовательности полей пользовательского архива.

2.3.2.13 Поля пользовательского архива: вкладка Sequence (Последовательность)

Вкладка Sequence (Последовательность)

Дополнительная вкладка Order (Порядок) используется для настройки полей данных в определенном порядке. Установленный порядок влияет на отображение данных в окне таблицы редактора User Archives (Пользовательские архивы), в окне управления средой исполнения и при присвоении индексов для доступа через функции языка сценариев WinCC.



Выберите одно или несколько полей и измените их положение в таблице с помощью клавиш Up (Вверх) и Down (Вниз). Подтвердите внесенные значения, нажав OK. В конце сохраните пользовательский архив, щелкнув кнопкой мыши на символе Save (Сохранить) или в меню Project – Save (Проект – Сохранить). Порядок следования полей пользовательского архива отобразится в редакторе User Archives (Пользовательские архивы) в столбце Pos. (Поз.).

В примере пользовательский архив Cola обладает следующими свойствами.

Пользовательские архивы	Поля данных	Свойства
Cola	Вода	Имя: вода
		Заголовок: Wasser_aus_Brunnen_4
		Тип: Integer
		Длина:
		Точность:
		Мин. значение: 1000
		Макс. значение: 1200
		Начальное значение: 1100
		Тег п...:
		Право на чтение: 0
		Право на запись: 0
		Флажки: NN
		П...: 3
		Последний доступ: 03/05/98 12:54
	Сахар	Имя
		Заголовок
	
	Краситель 7	Имя
		Заголовок
	
	Кофеин	Имя
		Заголовок
	
	Фосфорная кислота	Имя
		Заголовок
	

В конце сохраните пользовательский архив.

Примечание

Изменения в пользовательских архивах в базе данных принимаются после нажатия кнопки Save (Сохранить).

2.3.2.14 Свойства управляющих тегов

Диалоговое окно Tag Properties (Свойства тегов)

С помощью диалогового окна свойств пользовательских архивов и поля пользовательского архива можно редактировать свойства управляющих тегов. Нажмите кнопку Edit (Правка) на соответствующей вкладке в панели инструментов. Откроется диалоговое окно Tag properties (Свойства тегов), в котором при необходимости можно регулировать и изменять свойства тегов.

Типы данных управляющих тегов	
Тип данных для @UA_Cola_ID	32-битовое число со знаком
Тип данных для @UA_Cola_Job	32-битовое число со знаком
Тип данных для @UA_Cola_Field	Текстовый тег, 8-битовый
Тип данных для @UA_Cola_Value	Текстовый тег, 8-битовый

Примечание

Запрещается изменять тип данных управляющих тегов.

С помощью четырех управляющих тегов можно инициировать пользовательский архив. Для этого необходимо задать соответствующие значения либо тегам ID (Идентификатор) и Job (Задание), либо тегам Job (Задание), Field (Поле) и Value (Значение).

Функционирование управляющих тегов	
ID	Идентификатор (соответствует номеру записи) пользовательского архива
Job	Выделяют три вида заданий: чтение, запись и удаление. Чтение = 6 Запись = 7 Удаление = 8 После выполнения задания в этом управляющем теге отображается код ошибки. Ошибка отсутствует = 0 Ошибки = -1
Array	Поле архива
Value	Значение поля архива

Дополнительные комбинации значений управляющих тегов ID (Идентификатор) и Job (Задание)

ID	Job = 6	Job = 7	Job = 8
-1	Добавление записи данных	-	Удаление записи данных с наименьшим идентификатором
-6	Чтение записи данных с наименьшим идентификатором	Запись данных с наименьшим идентификатором	Удаление записи данных с наименьшим идентификатором

-9	Чтение записи данных с наибольшим идентификатором	Запись данных с наибольшим идентификатором	Удаление записи данных с наибольшим идентификатором
----	---	--	---

Управляющие теги предоставляют два способа доступа к пользовательским архивам.

1. Путем ввода управляющих тегов ID (Идентификатор) и Job (Задание) можно записывать, читать или удалять требуемые значения в записи данных.
2. Вместо управляющего тега ID (Идентификатор) можно использовать управляющие теги Field (Поле) и Value (Значение) для поиска записи данных. С помощью управляющих тегов Job (Задание) можно записывать, читать или удалять выбранную этим способом запись. Этот тип выбора данных может использоваться, например, если записи данных необходимо удалить из таблицы и снова вставить в конец таблицы. Поле Value (Значение) должно быть уникальным, иначе используется первая запись, для которой подходит значение условия.

Примечание

Чтобы проверить функцию, необходимо всегда указывать или создавать все четыре управляющих тега пользовательского архива с помощью мастера.

При создании пользовательского архива рекомендуется использовать мастер для создания и сохранения управляющих тегов.

Для получения дополнительной информации о добавлении управляющих тегов см. в примере использования управляющих тегов.

Дополнительные источники информации

Пример использования управляющих тегов (стр. 51)

2.3.2.15 Пример использования управляющих тегов

Пример использования управляющих тегов

Чтобы поработать с примером, демонстрирующим управляющие теги, необходимо выполнить следующие действия.

В редакторе User Archive (Пользовательский архив)

1. Создайте пользовательский архив (в примере пользовательский архив Cola). Введите настройки, приведенные ниже во время создания пользовательского архива с помощью мастера. Если проект Cola уже был создан, его можно проверить и при необходимости изменить настройки в свойствах пользовательского архива.

Свойства пользовательского архива Cola	
Тип пользовательского архива	Unlimited (Размер неограничен)
Связь	через теги WinCC
Управляющие теги	Создание
Группа тегов	@UA_Cola
Тип данных для @UA_Cola_ID	32-битовое число со знаком
Тип данных для @UA_Cola_Job	32-битовое число со знаком
Тип данных для @UA_Cola_Field	Текстовый тег, 8-битовый
Тип данных для @UA_Cola_Value	Текстовый тег, 8-битовый

1. Создайте в пользовательском архиве поля данных Water (Вода), Sugar (Сахар), Dye stuff 7 (Краситель 7), Caffeine (Кофеин) и Phosphoric acid (Фосфорная кислота) (тип Integer).
2. Создайте поле данных Recipes (Рецепты) (тип String).

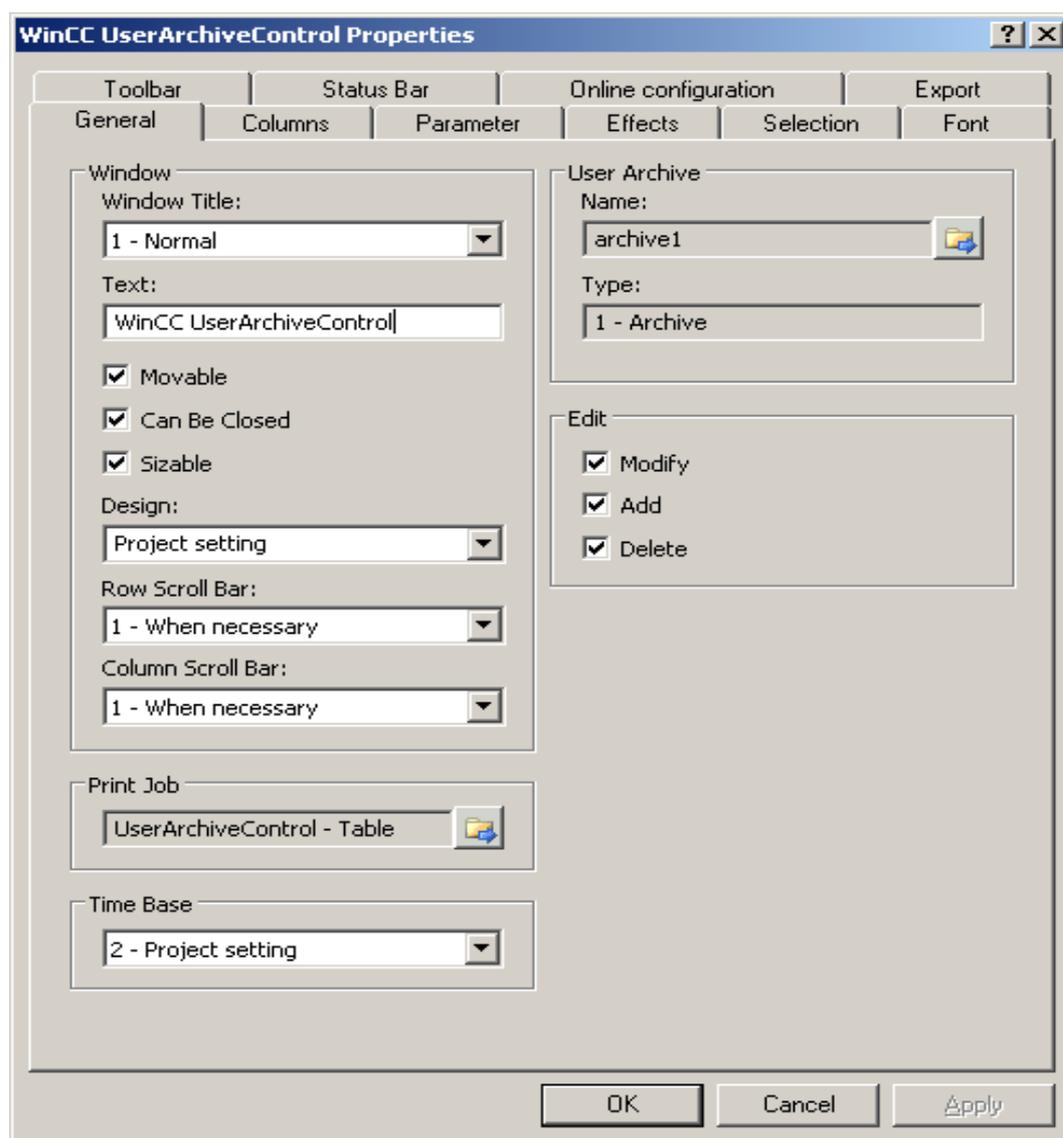
В графическом дизайнера

1. Откройте новый кадр и вставьте элемент управления WinCC UserArchiveControl.

Примечание

В версиях до WinCC V7 конфигурация отображения пользовательских архивов осуществлялась с помощью элемента управления User Archives Table Element (Табличный элемент пользовательских архивов).

Двойным щелчком мыши можно открыть диалоговое окно Properties of WinCC User ArchivesTable Element (Свойства элемента управления WinCC UserArchiveControl).



2. Измените следующие настройки.

- Выберите пользовательский архив Cola на вкладке General (Общие) в области User Archive (Пользовательский архив).
- Активируйте типы доступа Change (Изменение), Add (Добавление) и Delete (Удаление) в поле Edit (Редактирование).
- Другие настройки можно оставить без изменений. Если элемент UserArchiveControl уже создан, то можно проверить и в необходимых случаях изменить настройки в диалоговом окне свойств.

3. Создайте одно поле ввода-вывода для каждого из четырех управляющих тегов и выберите следующие настройки.

Управляющие теги	Data Format (Формат данных)	Формат вывода
@UA_Cola_ID	десятичные	0999
@UA_Cola_Job	десятичные	s9
@UA_Cola_Field	строка	*
@UA_Cola_Value	строка	*

1. Для каждого тега выберите свойство объекта Update for changes (Обновить по изменению).
2. Создайте поле ввода-вывода для каждого настроенного поля данных (вода, сахар и др.) и соедините их с соответствующими тегам (например, поле ввода-вывода Water (Вода) с тегом процесса @UA_Cola_Water). Для каждого тега выберите свойство объекта Update for changes (Обновить по изменению).



Примечание

Дополнительную информацию о настройке полей ввода-вывода можно найти в документации графического дизайнера.

Создайте текстовое поле для каждого настроенного поля ввода-вывода для маркировки таким образом, чтобы можно было назначить отдельные поля в среде исполнения. Сохраните выполненные изменения и запустите среду исполнения WinCC. Теперь введите пять записей данных в окно таблицы. Введите данные в запись ID 2 в столбец рецептов Cola и в четвертую запись Cola Light. На приведенном ниже рисунке показан пример элемента WinCC User Archive Table Element (Элемент таблицы пользовательского архива WinCC) (версия до WinCC V7).

process tags

water	sugar	caffeine	coloring 7	acid
100	50	20	20	20

ID	Recipes	Water	sugar	Caffeine	Color_7	Phosphoric_acid
1	Calif Coke	90	10	10	10	15
2	Coke	80	30	15	20	20
3	Standard	100	50	20	20	20
4	CokeLight	100	20	30	20	20
5	CherryCoke	100	80	50	5	10

control tags

ID	JOB	FIELD	VALUE
0003	+0		

Процедура выполнения возможных действий.

1. Выберите запись данных с идентификатором и запишите значения записи данных.
 - Введите идентификатор 3 в поле ввода-вывода ID (Идентификатор) и запишите в поле ввода-вывода Job (Задание) значение 7 (запись).
 - Теперь значения записи данных 3 отображаются в полях ввода-вывода тегов процесса.
 - Если действие было выполнено успешно, код ошибки «0» отобразится в поле ввода-вывода Job (Задание). При возникновении ошибки отобразится код ошибки «-1».
 - Управляющие теги Field (Поле) и Value (Значение) не требуются.

Примечание

При вводе идентификатора «-1» и задания «6» происходит считывание текущего содержания тегов процессов в таблицу. Новые значения присоединяются в конец таблицы; идентификаторы записей увеличиваются. Другие комбинации значений управляющих тегов ID (Идентификатор) и Job (Задание) можно найти в главе «Свойства управляющих тегов».

1. Выберите запись данных с идентификатором и считайте значения записи данных.
 - Измените значения полей ввода-вывода тегов процесса и введите «5» в поле ID (Идентификатор). В поле ввода-вывода Job (Задание) введите «6» (чтение).
 - Измененные значения тегов процесса записываются в запись данных «5». Значения, ранее содержащиеся в записи, перезаписываются.
 - Управляющие теги Field (Поле) и Value (Значение) не требуются.
1. Выберите запись данных с управляющими тегами Field (Поле) и Value (Значение).
 - Введите в поле ввода-вывода Field (Поле) слово Recipe (Рецепт), а в поле ввода-вывода Value (Значение) запишите Cola Light. В поле ввода-вывода Job (Задание) введите «7» (запись).
 - Теперь запись данных Cola Light запишется и значения записи данных отобразятся в полях ввода-вывода тегов процессов.
 - Управляющий тег ID (Идентификатор) не требуется и, следовательно, для него необходимо установить значение «0».

Примечание

Для поля, которое соотносится с управляющими тегами Value (Значение), в диалоговом окне Authorizations and Flags (Разрешения и флажки) необходимо установить флажок Field must possess an unique value (Поле должно обладать уникальным значением). В противном случае уникальное назначение записи данных в поле будет невозможным.

Дополнительные источники информации

Создание поля ввода-вывода в графическом дизайнера

2.3.3 Изменение конфигурации пользовательских архивов

При желании изменить или увеличить существующий пользовательский архив имеющиеся данные в таблице базы данных могут быть потеряны. В особенности это касается случаев, когда меняется структура таблицы базы данных или свойства поля, т. к. условия целостности базы данных больше не могут выполняться.

Чтобы избежать потерю данных, рекомендуется следующая процедура.

1. Отключите среду исполнения, откройте пользовательский архив в редакторе пользовательских архивов и выполните необходимые изменения. Сохраните архив только после завершения всех изменений (кэш не сохраняйте).
2. После сохранения нажмите кнопку Edit Runtime Data (Редактировать данные среды исполнения). В открывшейся таблице можно посмотреть, доступны ли существующие данные в пользовательском архиве.
3. Если данные до сих пор присутствуют, архив можно использовать или выполнять дальнейшие изменения. При выполнении изменений после каждого процесса сохранения проверяйте, доступны ли данные.
4. Если после выполнения изменений данные отсутствуют, сначала отмените все произведенные изменения в архиве без сохранения. Затем выберите архив в редакторе пользовательских архивов, а в контекстном меню команду Restore RT data (Восстановить данные среды исполнения). Таким образом, предыдущие данные среды исполнения запишутся в таблицу. После этого сохраните архив и проверьте, записались ли данные в архив.

<u>N</u> ew Field	Ins
<u>D</u> elete	Del
<u>R</u> estore RT Data	Ctrl + Y
<u>P</u> roperties	F2

Примечание

После каждого сохранения проверяйте, доступны ли данные среды исполнения. Если данные были утеряны после сохранения первых изменений и затем было произведено второе сохранение, данные утрачиваются.

Если при запуске среды исполнения после выполнения изменений обнаруживается потеря данных, данные можно восстановить способом, описанным выше, с помощью команды Restore Rt data (Восстановить данные среды исполнения) в таблице. Данные можно восстановить, даже если редактор User Archives (Пользовательские архивы) закрыт или выполнен выход из WinCC. Важно отметить, что эта процедура возможна, только если изменения сохранялись один раз.

При выполнении комплексных изменений в пользовательском архиве рекомендуется сначала экспортировать существующие данные, чтобы предотвратить их потерю. Данные можно вновь записать в измененный архив, настроив таблицу экспортированных данных.

При изменении конфигурации пользовательских архивов на сервере клиенту без собственного проекта требуется обновление пакета сервера, чтобы правильно отображать данные в среде исполнения. На сервере можно сгенерировать обновление вручную или с помощью неявного обновления пакетов. Для неявного обновления пакетов выберите настройку Generate WinCC CS Server data immediately when changes are made (Генерирование данных сервера WinCC CS сразу после выполнения изменений) в редакторе сервера Server Data (Данные сервера) с помощью контекстного меню Implicit Update (Неявное обновление).

Дополнительные источники информации

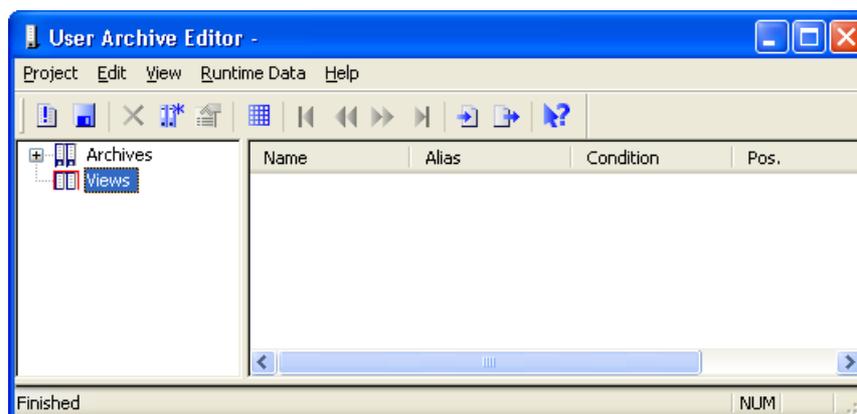
Команда меню Export (Экспортировать) (меню Runtime Data (Данные среды исполнения)) (стр. 25)

2.3.4 Конфигурация представлений**2.3.4.1 Создание новых представлений**

Параметр по умолчанию Put archive in cycle (Поместить архив в цикл) позволяет один за другим создавать множество представлений. Чтобы создать одно представление, можно отключить параметр в меню Edit – Options (Правка – Параметры).

Выполните следующие действия, чтобы создать новое представление.

1. В окне навигации щелкните левой кнопкой мыши на Views (Представления).

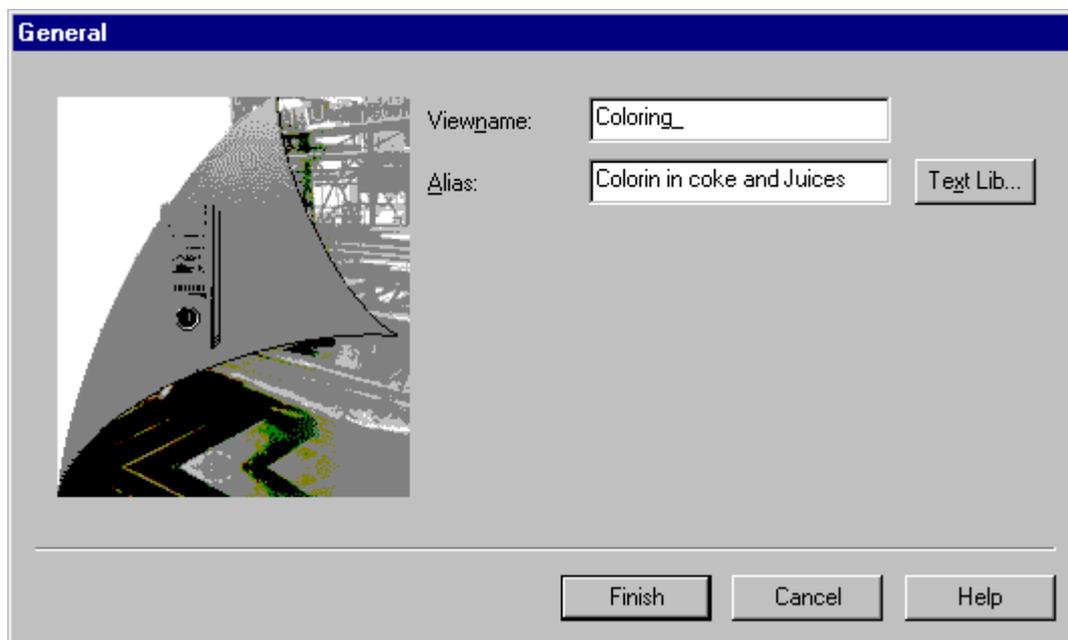


2. Затем щелкните в окне навигации или данных правой кнопкой мыши. Появится контекстное меню, показанное на изображении ниже.



3. Выберите параметр New View (Новое представление).

Запустится мастер для конфигурации представлений. В диалоговом окне General (Общие) можно создать новую форму.



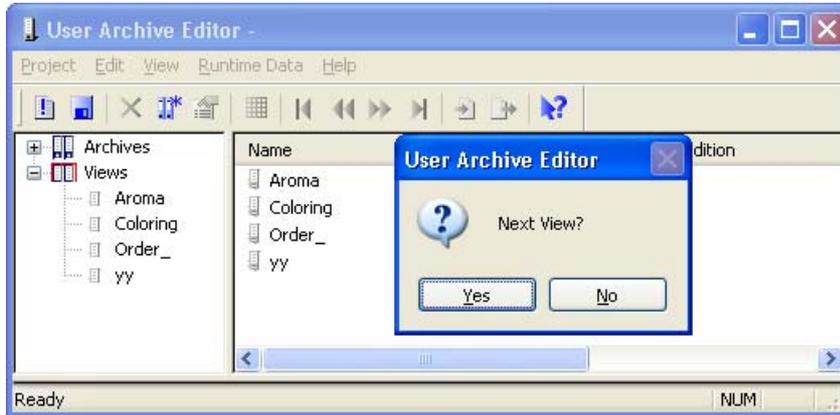
Введите имя формы, например Dyes. В поле Alias (Заголовок) можно задать форме второе имя, например Dye stuffs in Cola and Orange Juice (Красители в напитках Cola и апельсиновом соке), чтобы дать пользовательскому архиву комментарий или изменить язык с помощью текстовой библиотеки в среде исполнения. Ввод этих данных является необязательным.

Для получения дополнительной информации об изменении языка см. интерактивную справку.

Завершите создание формы, нажав кнопку Finish (Готово).

После создания формы появится вопрос Add columns? (Добавить столбцы?). При нажатии кнопки Yes (Да) откроется диалоговое окно General (Общие) для создания формы. Информацию о создании столбцов в форме см. в главе «Создание столбцов формы».

При активировании параметра Create forms in cycle (Создавать формы циклически) в меню Edit – Options (Правка – Параметры) после создания столбцов формы появится запрос Next form? (Следующая форма?). При нажатии кнопки Yes (Да) появится первоначальное диалоговое окно General (Общие) для ввода данных следующей формы.



После этого сохраните формы.

Примечание

Изменения форм в базе данных принимаются только после нажатия кнопки Save (Сохранить).

Дополнительные источники информации

Изменение языка (стр. 71)

2.3.4.2 Создание новых столбцов представления

Параметр по умолчанию Put columns of a view in cycle (Поместить столбцы представления в цикл) позволяет один за другим создавать множество столбцов. Чтобы создать один столбец, можно отключить параметр в меню Edit -Options (Правка – Параметры).

1. Разверните раздел Views (Представления) в окне навигации. Новое представление, например Dyes, отобразится в окне навигации.
2. В окне навигации щелкните левой кнопкой мыши на представлении Dyes. Отобразится следующее контекстное меню.



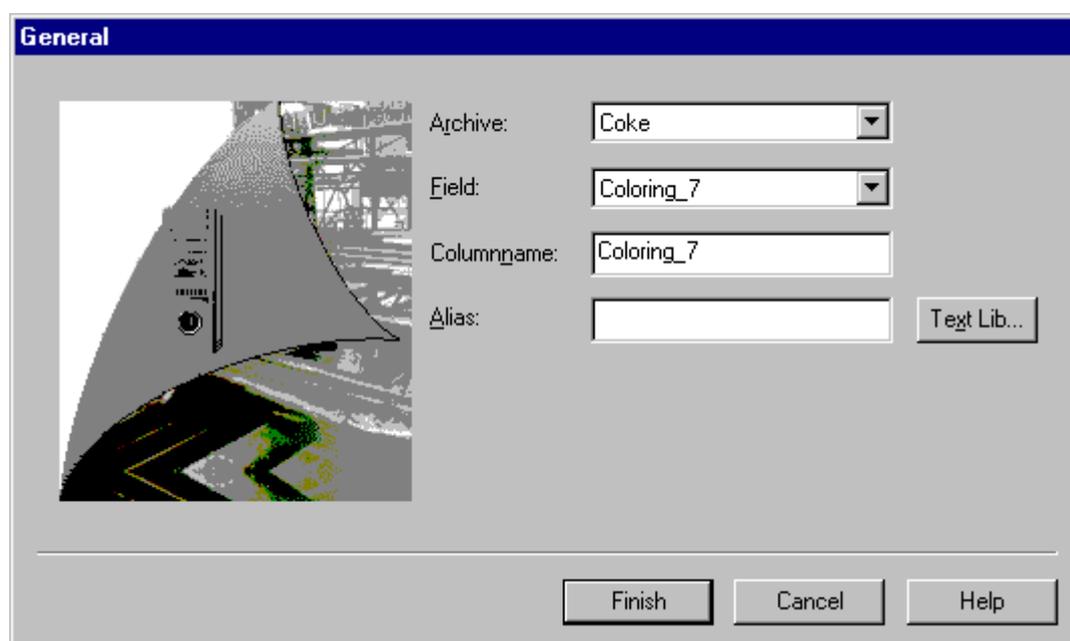
3. Нажмите кнопку New Column (Новый столбец).

Откроется диалоговое окно General (Общие).

2.3.4.3 Основные свойства столбцов представления

Основные свойства столбцов представления

В диалоговом окне General (Общие) из пользовательского архива можно выбрать поля, объявить их столбцами созданной пользователем формы и дать им имена.

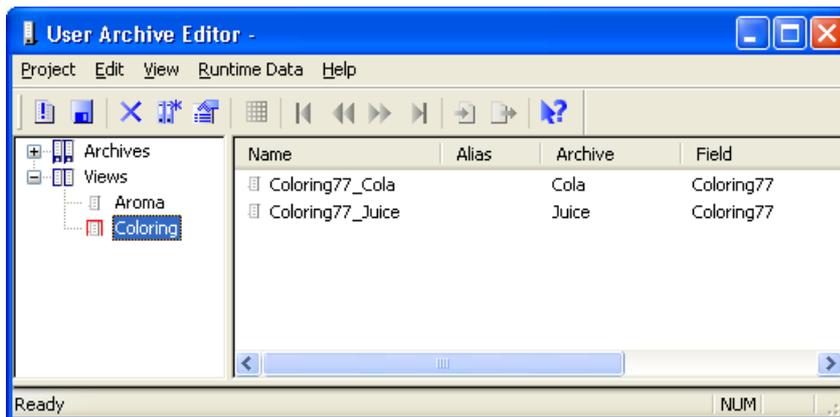


- В диалоге выбора Archive (Архив) можно выбрать один из настроенных пользовательских архивов. Например, не изменяйте настройку Cola.
- В диалоговом окне выбора Field (Поле) выберите одно из полей из пользовательского архива Cola. Например, не изменяйте настройку Dye 7. Перейдите в следующее поле с помощью клавиши TAB или щелчка кнопки мыши для дальнейшего ввода данных.
- При нажатии на поле Column Name (Имя столбца) принимается запись в поле Field (Поле). Имя колонки также можно выбирать произвольно, однако имя в рамках представления должно быть уникальным. К примеру, утвердите имя Dyes.

В поле Alias (Заголовок) можно присвоить колонке второе имя, чтобы дать колонке комментарий или изменить язык с помощью текстовой библиотеки в среде исполнения. Ввод этих данных является необязательным.

Для получения дополнительной информации об изменении языка см. интерактивную справку.

После нажатия кнопки Finish (Готово) выполняется установка настроенных данных.



Изготовитель напитков в примере создает форму Dyes, объединив поля данных Dye 7 и Dye 16 из пользовательского архива Cola и Juice.

После установки флажка Create columns of a view in cycle (Создавать столбцы представления циклически) в меню Edit – Options (Правка – Параметры) появится вопрос Next column? (Следующий столбец?). При выборе кнопки Yes (Да) появится первоначальное диалоговое окно General (Общие) для ввода данных в следующий столбец.

В конце сохраните поля представления.

Примечание

Изменения форм в базе данных принимаются только после нажатия кнопки Save (Сохранить).

Дополнительные источники информации

Изменение языка (стр. 71)

2.3.4.4 Свойства представлений

Чтобы отредактировать свойства формы, выполните следующие действия.

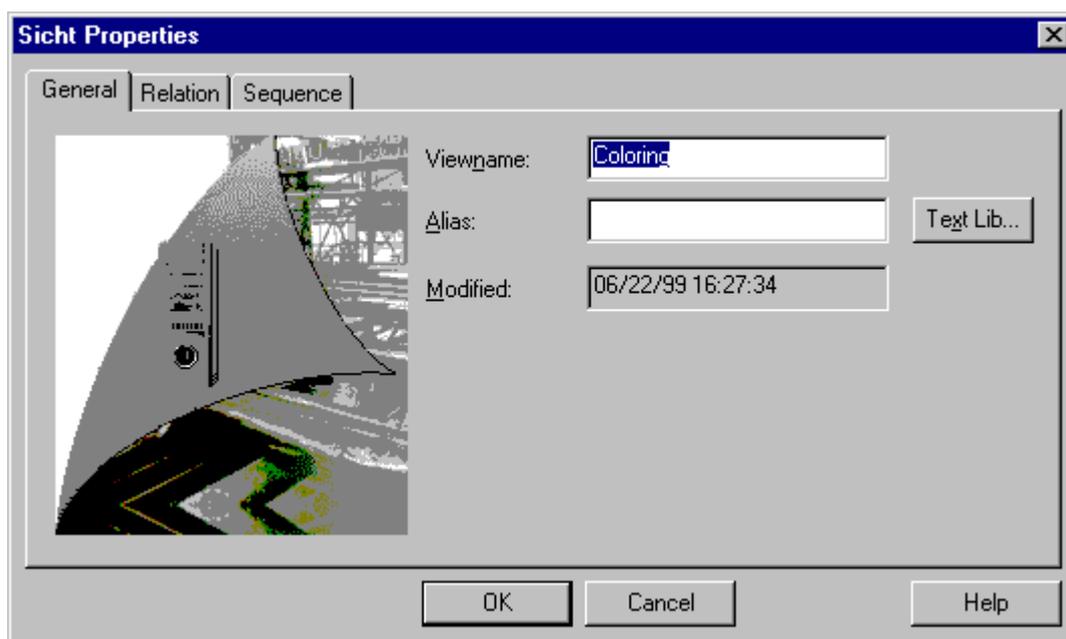
1. Щелкните правой кнопкой мыши на одном из представлений в редакторе User Archives (Пользовательские архивы).
2. В контекстном меню выберите Properties (Свойства).

Появится диалоговое окно Properties of the view (Свойства представления).

2.3.4.5 Основные свойства представлений

Основные свойства представлений

На вкладке General (Общие) будут отображены свойства выбранного представления.

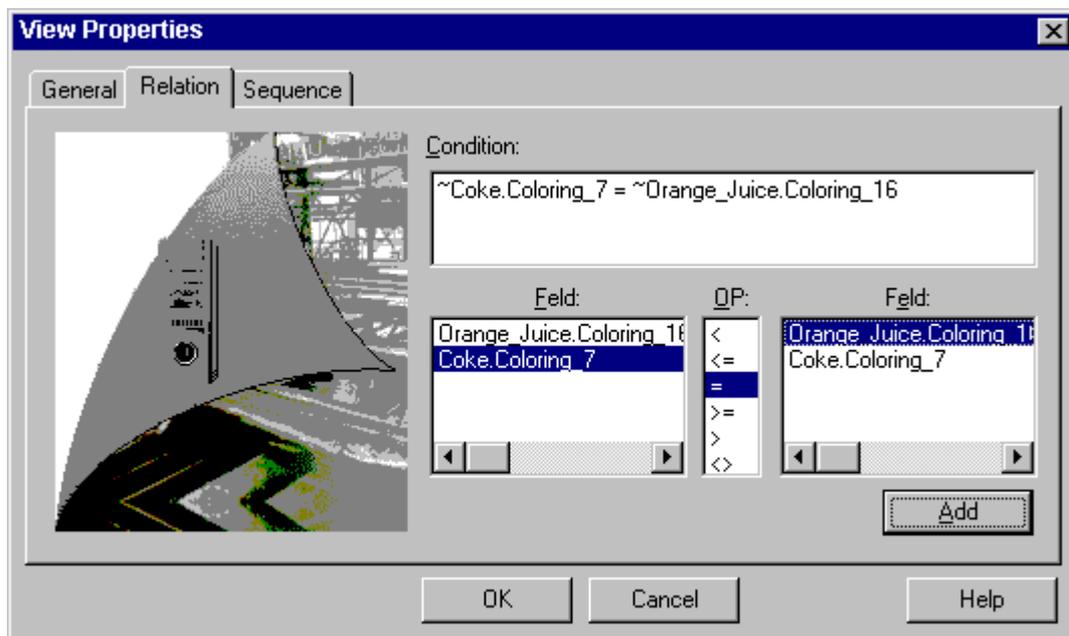


В поле View name (Имя представления) можно изменить имя представления, а в поле Alias (Заголовок) имя заголовка. В этом диалоговом окне можно также посмотреть дату и время последнего изменения.

2.3.4.6 Настройки связи между представлениями

Настройки связи между представлениями

На вкладке Relation (Связь) можно создать связь между несколькими архивами для вывода представления. Можно определить логические операции напрямую в языке SQL или задать их интерактивно с помощью предоставленных операторов связи. Убедитесь, что поля пользовательского архива, для которых необходимо установить связь, обладают одним типом тега.



Relation (Отношение)

В поле Relation (Связь) можно напрямую ввести операторы SQL. Подробную информацию о SQL-Sprache см. в приложении.

Условия

Условия можно ввести в поля выбора интерактивно. Для этого выберите поля в правом и левом списке Field (Поле) и задайте связь, выбрав соответствующую операцию в списке OP (Op.). Условие будет принято после нажатия кнопки Add (Добавить), после чего оно отобразится в поле Relation (Связь).

Принцип

Все поля, в которых отображается связь, соединены друг с другом в выбранном пользовательском архиве. При использовании связи выполняется фильтрация содержания полей, а результат отображается в среде исполнения в виде формы.

Данные представления также можно редактировать в среде исполнения, а измененные данные вставляются в исходный архив.

Примечание

В соединенных пользовательских архивах должен отображаться хотя бы один общий фактор или связь.

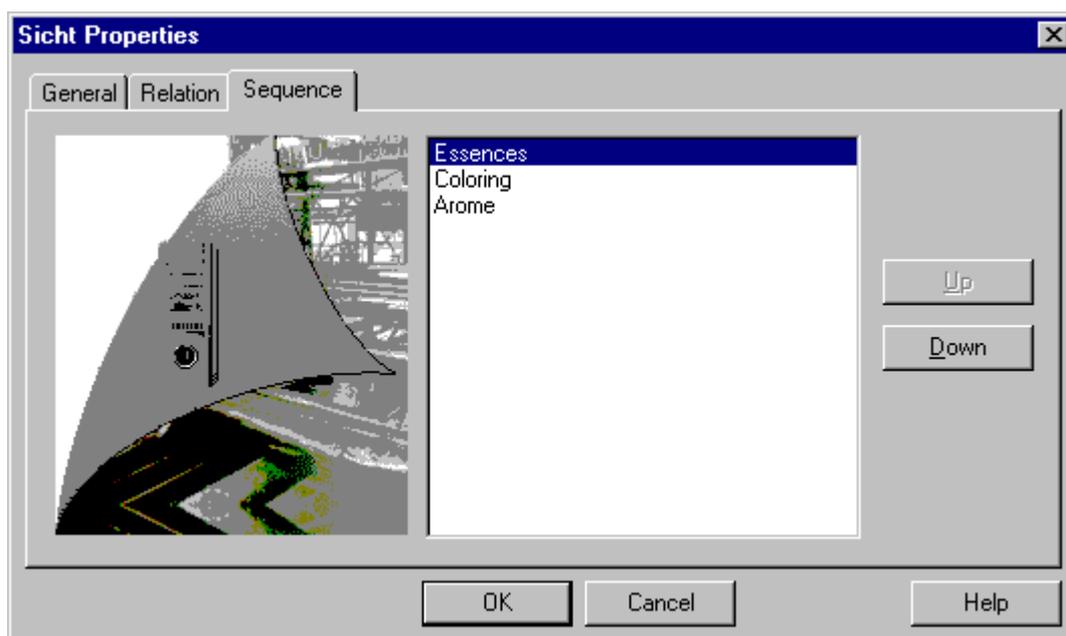
Дополнительные источники информации

Изменение языка (стр. 71)

2.3.4.7 Настройка последовательности представлений

Настройка последовательности представлений

На вкладке Sequence (Последовательность) задайте порядок следования форм.

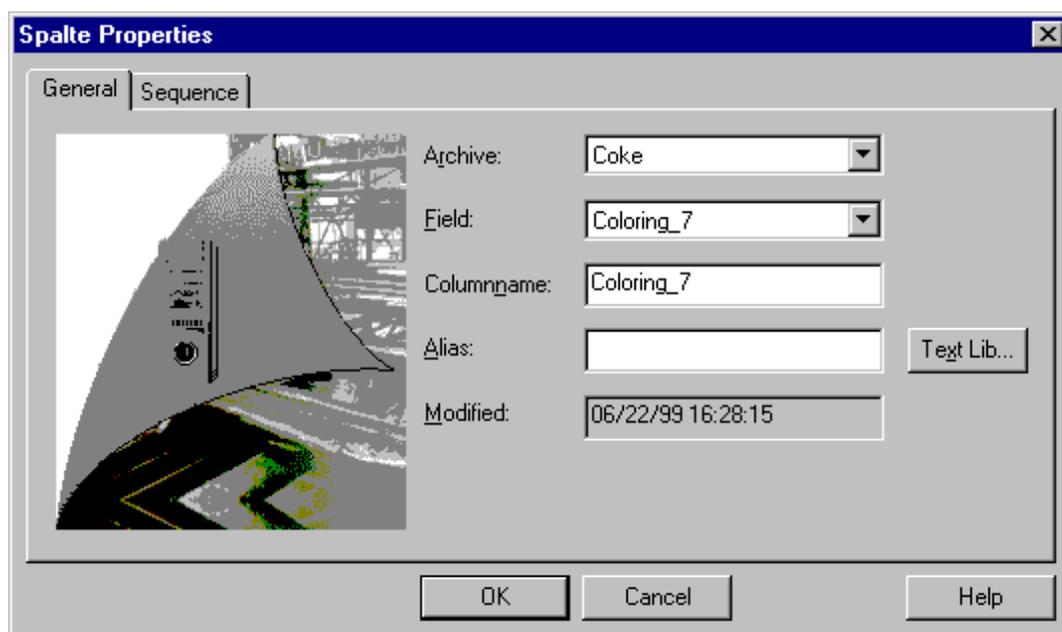


Выберите одно или несколько представлений и измените их положение в таблице с помощью клавиш Up (Вверх) и Down (Вниз). Подтвердите внесенные значения, нажав OK. В конце сохраните представления, щелкнув кнопкой мыши на символе Save (Сохранить) или в меню Project – Save (Проект – Сохранить). Порядок следования представлений отобразится в редакторе User Archives (Пользовательские архивы) в столбце Pos. (Поз.).

2.3.4.8 Свойства столбцов представления

Чтобы отредактировать свойства столбцов представления, выполните следующие действия.

1. Щелкните правой кнопкой мыши на одном из столбцов представления в редакторе User Archives (Пользовательские архивы).
2. В контекстном меню выберите Properties (Свойства). Появится диалоговое окно Properties of column (Свойства столбца).

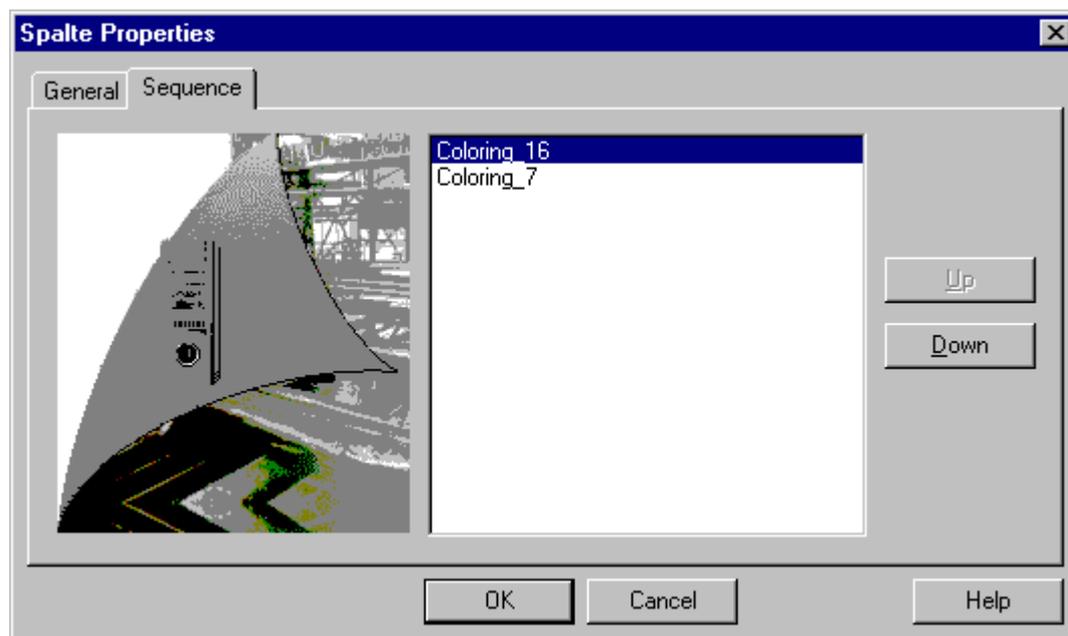


На вкладке General (Общие) содержатся такие же поля, как и при настройке нового столбца представления. Дату и время последнего изменения можно посмотреть в поле Changed (Измененный).

2.3.4.9 Настройка последовательности столбцов представления

Настройка последовательности столбцов представления

На вкладке Sequence (Последовательность) можно задать порядок следования столбцов представления.



Выберите один или несколько столбцов и измените их положение в таблице с помощью клавиш Up (Вверх) и Down (Вниз). Подтвердите внесенные значения, нажав OK. В конце сохраните представления, щелкнув кнопкой мыши на символе Save (Сохранить) или в меню Project – Save (Проект – Сохранить). Установленный порядок следования влияет на отображение столбцов в окне таблицы редактора User Archives (Пользовательские архивы) и в среде исполнения

2.3.4.10 Советы по конфигурации

- Связь между AS и пользовательским архивом ограничена одним соединением на пользовательский архив.
- Пока связь с системой автоматизации установлена, PLCID не может содержать больше 8 символов.
- Элементы, в которых содержатся специальные символы или зарезервированные слова, не принимаются в качестве имен полей и таблиц. Для получения более подробной информации см. также главу «Алфавитный список ключевых слов SQL».
- Сохранение изменений конфигурации в среде исполнения можно выполнить, только если в это время в любом из пользовательских архивов не выполняется синхронизация при резервировании.
- При резервировании WinCC на обоих серверах должна использоваться одинаковая структура пользовательских архивов, которые будут синхронизироваться. Конфигурация этих пользовательских архивов должна совпадать по свойствам и структуре полей/записей.
- Для связи пользовательских архивов через теги необработанных данных имена используемых тегов необработанных данных на обоих серверах также должны быть одинаковыми.
- Если после запуска редактора User Archive (Пользовательский архив) все поля панели инструментов недоступны (кроме восстановления), в пути проекта необходимо удалить файл UEditor.loc. Это также относится к элементу UserArchiveControl, который больше не работает.
- При получении сообщения об ошибке Error while connecting the data! (Сбой во время соединения с данными!) во время запуска среды исполнения или во время переключения элемента UserArchiveControl в среде исполнения это означает, что элемент UserArchiveControl не подключен к пользовательскому архиву или представлению. Проверьте, правильно ли было установлено соединение, была ли изменена конфигурация и существует ли выбранный пользовательский архив или представление.

Примечание

Во время конфигурации элемент UserArchiveControl подключается к выбранному пользовательскому архиву или форме и только после этого может получать доступ к этому пользовательскому архиву или форме. Для получения доступа необходимо разрешить пользовательский архив/форму (защита доступа). Элементу управления в редакторе User Administrator (Администратор пользователей) можно присвоить определенные разрешения.

Если отменить защиту доступа, необходимо повторно подключить элемент UserArchiveControl в графическом дизайнера к пользовательскому архиву, чтобы элемент UserArchiveControl смог определить отключение защиты доступа.

Защита доступа к архиву или полю запрашивается при открытии экрана элемента управления UserArchiveControl. Защиту доступа к управляющим тегами элементов защищенных архивов необходимо устанавливать отдельно с помощью свойств объекта, например, для изображения, поля ввода/вывода или кнопки.

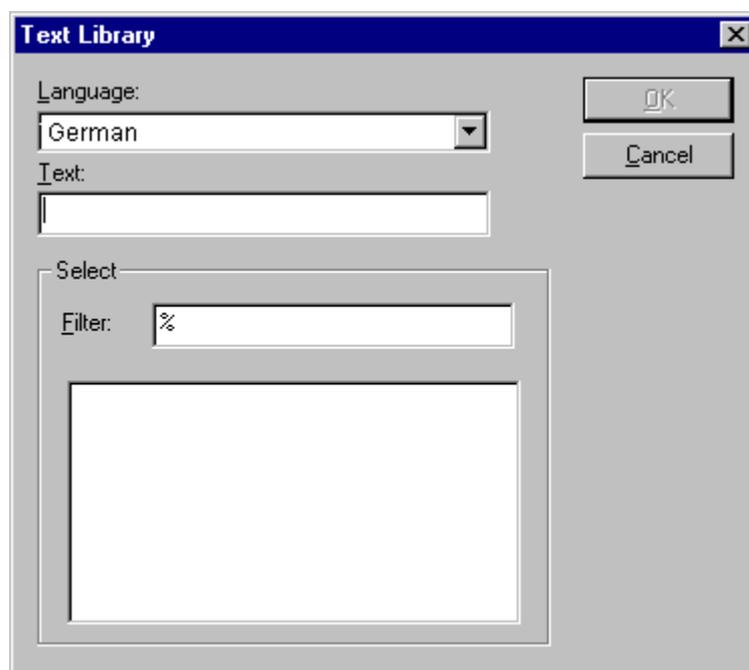
Дополнительные источники информации

Резервные пользовательские архивы

Алфавитный список ключевых слов SQL (стр. 82)

2.3.4.11 Изменение языка

Для пользовательских архивов, полей пользовательских архивов, представлений или полей представления язык можно изменить с помощью текстовой библиотеки. Для этого нажмите на кнопку TextLib... (Текстовая библи...) в соответствующем диалоговом окне. Откроется диалоговое окно текстовой библиотеки.



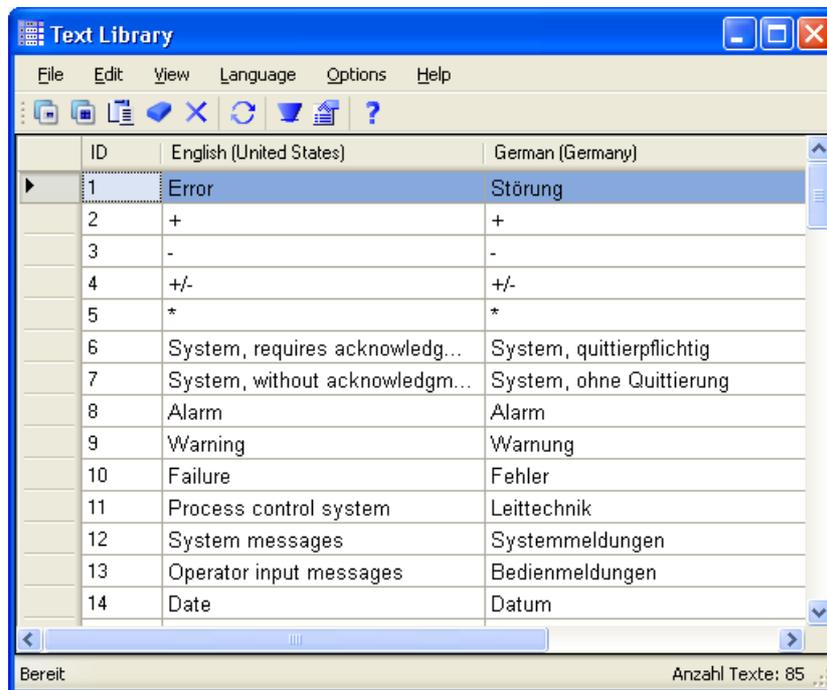
Процедура для текстов из текстовой библиотеки

1. Сначала в диалоговом окне выбора Language (Язык) выберите язык для конфигурирования.
2. Если требуемые языки уже были созданы в текстовой библиотеке, все существующие тексты можно загрузить в поле выбора, щелкнув кнопкой мыши. Они отобразятся в окне выбора, в котором можно выбрать необходимый элемент.
3. После этого выбранный элемент появится в поле Text (Текст).
4. Подтвердите выбор, нажав "OK".
5. В поле Alias (Заголовок) номер позиции этого слова указывается из текстовой библиотеки.

При изменении языка в среде исполнения появится элемент языка, выбранный в текстовой библиотеке.

Процедура для новых текстов

1. Сначала в диалоговом окне выбора Language (Язык) выберите язык для конфигурирования.
2. Введите текст или слово, язык которого необходимо поменять, в поле Text (Текст).
3. Подтвердите выбор, нажав «ОК».
4. В поле Alias (Заголовок) номер позиции этого слова указывается из текстовой библиотеки.
5. Откройте редактор Text library (Текстовая библиотека). Здесь можно вписать перевод текста, внесенного в редактор User Archives (Пользовательские архивы), в столбцы требуемого языка.



6. Завершив перевод, закройте текстовую библиотеку.

При изменении языка в среде исполнения появится элемент языка, выбранный в текстовой библиотеке.

Примечание

В проектах клиентов в текстовой библиотеке сервера и клиента должны использоваться те же идентификаторы текстов, что и в текстах пользовательских архивов. В противном случае текст неправильно отобразится в среде исполнения на клиентском компьютере.

В поле Filter (Фильтр) можно указать свойства фильтрации для текстов из текстовой библиотеки, например «a%» перечисляет все слова из текстовой библиотеки, которые начинаются на букву а. При выборе нового критерия фильтрации нужно снова нажать на окне выбора, чтобы обновить выбор текста.

2.3.5 Пример пользовательского архива

Пользовательский архив для изготовления напитков

Изготовитель напитков Sun Drink в примере производит колу Calif Cola и апельсиновый сок Sunny Juice. Чтобы сохранить рецепты с составом напитков, изготовитель применяет пользовательский архив WinCC. Если контейнер хранения системы наполнения пуст, данные рецепта отправляются по каналам связи из WinCC в AS. Затем AS может снова наполнить контейнер хранения с помощью данных рецепта. На приведенном ниже рисунке показан пример WinCC User Archive Table Element (Элемент таблицы пользовательского архива WinCC) (версия до WinCC V7).

The screenshot shows the WinCC-Runtime interface for a beverage production system. At the top, there are three ingredient containers labeled 'water', 'sugar', and 'caffeine', each with a red liquid level indicator. These are connected via pipes to a 'filing station' which includes a pump and a large green mixing tank containing a red liquid. Below the diagram is a data table with the following structure:

	ID	water	sugar	coloring7	caffeine
1	1	100	50	30	20
2	2	100	30	25	20
3	3	80	40	0	30
4	4	70	30	20	20
5	5	100	80	50	10

At the bottom of the interface, there are status indicators: 'Finished', 'Rec 1/5', 'Row 1', and 'Col 6'.

Пользовательские архивы используются следующим образом.

- **Пользовательские архивы:** один пользовательский архив предназначен для напитка, а другой для апельсинового сока.
- **Представления:** используйте представления, чтобы объединять данные полей двух архивов, в этом примере – краситель товарной группы.

Каждый пользовательский архив состоит из полей данных с редактируемыми свойствами. В примере содержимое напитка Cola находится в полях данных. Каждое поле данных обладает рядом свойств, таких как имя, имя заголовка, тип, длина, значение и т. д. Поля данных и свойства в редакторе User Archives (Пользовательские архивы) отображаются в виде строк и столбцов. Поэтому здесь говорится о рядах вместо полей данных и о столбцах вместо свойств. Структура пользовательского архива Cola выглядит следующим образом.

Пользовательский архив Cola	Свойства (столбцы)						
	Имя	Alias (Заголовок)	Тип	Length (Длина)	Мин. значение	Макс. значение	Start value (Начальное значение)
Поля данных (строки)							
Вода	Вода	Brun.5	Int	2	1000	1500	1000
Сахар	Сахар	Zmela	Int	2	120	140	130
Краситель 7	Краситель7	FS1007	Int	2	6	8	6
Кофеин	Кофеин	Кофеин	Int	2	2	3	2
Фосфорная кислота	Фосфорная кислота	PhosAc	Int	2	170	190	170

2.4 Обмен данными с SIMATIC S5/S7

2.4.1 Интерфейс фрейма сообщения SIMATIC

Обмен данными между пользовательскими архивами и ПЛК S5 и S7 можно произвести через теги необработанных данных или теги WinCC. Можно использовать все интерфейсы SIMATIC, кроме интерфейса программирования AS511.

Обмен данными с WinCC с помощью тегов необработанных данных можно выполнить, используя следующие команды:

- S7-400;
- S5-PLC-115U или более поздней версии.

Ниже будут рассмотрены следующие темы.

- Обмен данными с S5 и S7 с помощью тегов WinCC
- Обмен данными с S5 и S7 с помощью тегов необработанных данных

- Различия в форматах данных между WinCC и S5/S7.

2.4.2 Обмен данными с S5 и S7 с помощью тегов WinCC

Обмениваться данными с S5 и S7 с помощью тегов WinCC очень просто. Однако необходимо обратить внимание на то, что для типов данных пользовательских архивов могут использоваться только определенные типы данных тегов WinCC.

При использовании типов данных Integer, Double и String в редакторе User Archives (Пользовательские архивы) следующие типы данных должны применяться в управлении тегов менеджера тегов. Для типа данных Date/Time пользовательских архивов отсутствует соответствующий тип данных в управлении тегов.

Выбор в редакторе пользовательских архивов	Управление тегами/Тег WinCC
Число (int)	32-битовое число со знаком
Число (float)	32-битовое число с плавающей точкой IEEE 754
Число (double)	64-битовое число с плавающей точкой IEEE 754
Символьная строка (String)	Текстовый тег, 8-битовый шрифт
Дата/время	Отсутствует соответствующий тип данных

2.4.3 Обмен данными с S5 и S7 с помощью тегов необработанных данных

2.4.3.1 Обмен данными с S5 и S7 с помощью тегов необработанных данных

Далее будет описан процесс обмена данными между пользовательским архивом и системой автоматизации с помощью тегов необработанных данных WinCC. Для этого в AS используется функция BSEND/BRCV. Теги необработанных данных отправляются активно по инициативе AS. В фреймах сообщений содержатся один или несколько запросов к пользовательскому архиву WinCC. Это могут быть как запросы на запись, так и запросы на чтение. В качестве ответа на такие запросы WinCC отправляет данные и квитирование обработки.

Примечание

Т.к. AS является активным партнером для обмена данными, то функция пользовательского архива, выбираемая по инициативе пользователя в WinCC, должна быть запущена в AS, например запись/чтение архивных значений. Этот запуск можно выполнить с помощью, например внешних тегов WinCC в AS.

Параметр Job type (Тип задания), использующийся во время обмена данными в заголовке задания или квитирования, не может применяться для запуска функций AS, т. к. он имеет функциональность только в отношении пользовательских архивов.

Пользователь найдет информацию по следующим темам:

- Отправка заданий/данных в WinCC
- Отправка квитирования обработки /данных в SIMATIC S5 и S7

- Структура заголовка связи

2.4.3.2 Отправка заданий/данных в WinCC

Необработанные данные

Структура тега необработанных данных для отправки заданий и данных из SIMATIC S5 и S7 ПЛК в WinCC.

Блок данных в S5/S7
Заголовок телеграммы Telegram Header
Заголовок задания 1 Job Header1
Данные задания 1 Data of Job 1
Возможный заголовок задания 2
Возможные данные задания 2
Задание n

2.4.3.3 Отправка квитирования обработки /данных в SIMATIC S5/S7

Тег необработанных данных

Структура тега необработанных данных для отправки квитирований обработки и данных из WinCC в SIMATIC S5 и S7 ПЛК.

Тег необработанных данных для отправки в S5 и S7
Квитирование обработки
Заголовок квитирования
Данные квитирования

2.4.3.4 Структура заголовка связи

Структура блоков данных

Структура отдельных блоков фреймов сообщений (распределение в байтах).

Функция поля	Комментарий
длина фрейма сообщения в байтах LSB *)	Длина поля 4 байта
.	макс. длина 4091 байта
.	.(транспорт S5/ S7)
длина фрейма сообщения в байтах MSB **)	.

Тип передачи	1 из WinCC, 2 из ПЛК
Зарезервированный	
Число заданий во фрейме сообщения LSB *)	Длина поля 2 байтов
Число заданий во фрейме сообщения MSB**)	.
PLCID 1.character	Имя задается
.	в ASCII
.	Длина поля 8
.	байтов
.	.
.	.
.	.
PLCID 8.character	.

*) LSB = Least Significant Byte (LSB – младший значащий байт)

***) MSB = Most Significant Byte (MSB – старший значащий байт)

2.4.3.5 Заголовок задания

Структура заголовка задания

Структура отдельного заголовка задания (распределение в байтах).

Функция поля	Комментарий
Длина задания в байтах LSB	Длина поля 2 байтов
Длина задания в байтах MSB	.
Тип задания Job Type/Job Type	См. описание
Зарезервированный	
Номер поля LSB	Длина поля 2 байтов
Номер поля MSB	.
Номер записи LSB	Длина поля 4 байтов
.	.
.	.
Номер записи данных MSB	.
Критерий выбора LSB	Номер поля, в соответствии с которым производится выбор
Критерий выбора MSB	(не для 0) Длина поля 2 байта

2.4.3.6 Данные задания

Данные задания

Данные задания соответствуют записи данных (или адресованному полю).

Примечание

1. Текстовые поля не заканчиваются на \0 !!!
2. Числа должны передаваться в формате Intel (сначала LSB, потом MSB).
3. У поля Integer длина 4 байта, поля Float 4 байта, поля Double 8 байт.
4. Данные перемещаются по длине поля, выбранной в качестве критерия выбора, когда значение критерия выбора не равно 0.
5. При использовании критерия выбора начало области данных используется в качестве значения выбора в длине поля критерия выбора.

2.4.3.7 Заголовок квитирования

Заголовок квитирования

Структура отдельного заголовка квитирования (распределение в байтах).

Функция поля	Комментарий
Длина фрейма сообщения в байтах LSB	Длина поля 4 байтов
.	.
.	.
длина фрейма сообщения в байтах	.
Тип передачи	1 из WinCC, 2 из ПЛК
Зарезервированный	
Тип задания Job TypeJob Type	См. описание
ErrorCodeErrorCode	См. описание
Зарезервированный	
Зарезервированный	
Номер поля LSB	Длина поля 2 байтов
Номер поля MSB	.
Номер записи LSB	Длина поля 4 байтов
.	.
.	.
Номер записи данных MSB	.
PLCID 1.character	Имя задается
.	с помощью ASCII
.	Длина поля 8 байтов
.	.
.	.
.	.
.	.
PLCID 8.character	.

Данные квитирования

Квитирование содержит либо запись данных, либо адресованное поле (при запросе на чтение) или оно пустое (задание на запись, архивирование).

2.4.3.8 Описание типов заданий и кодов ошибок

Описание типов заданий

Тип	Описание
4	Проверить наличие пользовательского архива
5	Удалить все записи в пользовательском архиве
6	Прочитать набор данных
7	Внести запись данных
8	Удалить запись
9	Прочитать поле записи данных
10	Внести поле записи данных

Описание кодов ошибки

Group (Группа)	№	Описание	Возможные причины ошибки
General	0	Функция выполнена	--
archive	2	Данные недоступны	- ни один архив не конфигурировался с этим PLCID
Запись данных	101	Недопустимые данные	- Компоновка записи не совпадает, например число или тип полей - сбой при добавлении или обновлении записей, т. к. настроен тип архива limited (ограниченный) или для поля настроено минимальное или максимальное значение - неправильный критерий фильтрации
Запись данных	102	Данные недоступны	(только для типа задания 6) - отсутствие доступных данных - неправильный критерий фильтрации
Array	201	Недопустимые данные	(только для типа задания 10) - неправильный критерий фильтрации, т. к., например поле недоступно или для поля настроено минимальное или максимальное значение

Array	202	Данные недоступны	(только для типа задания 9) - неправильный критерий фильтрации или поле не обнаружено, т. к. не соответствует критерию фильтрации
General	254	Функция недоступна	--
General	255	Неизвестная ошибка	--

2.4.4 Различия в форматах данных между WinCC и S5/S7.

Формат данных в WinCC отличается от формата данных в ПЛК SIMATIC-S5/S7. Это необходимо учитывать во избежание нежелательных ошибок.

В WinCC применяются форматы данных Intel и Microsoft, где первыми хранятся младшие значащие байты, а старшие значащие байты последними. Такой формат данных очень распространен и известен как Intel Format (Формат Intel). Пример поможет прояснить Intel Format (Формат Intel).

Формат Intel

В формате Intel десятичное число 300 сохраняется следующим образом.

Бит	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
двоичные	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0

Шестнадцатеричный	0	1	2	C
-------------------	---	---	---	---

Десятичное число 300 в соответствии с форматом Intel соответствует шестнадцатеричному числу 12C ($1 \cdot 256 + 2 \cdot 16 + 12$).

Формат SIMATIC

В формате SIMATIC младшие значащие байты хранятся на самых значимых местах. В формате SIMATIC десятичное число 300 сохраняется следующим образом.

Бит	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
двоичные	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1

Шестнадцатеричный	2	C	0	1
-------------------	---	---	---	---

Десятичное число 300 в соответствии с форматом SIMATIC соответствует шестнадцатеричному 2C01. Если 2C01 будет неправильно интерпретировано в соответствии с форматом Intel, то получится десятичное число 11265, что является существенным отклонением.

Для ПЛК SIMATIC доступен функциональный блок, который может выполнять соответствующие преобразования данных. Их необходимо запускать всегда до и после передачи данных между S5/S7 и WinCC. Блоки функций можно загрузить в Интернете с веб-сайта технической поддержки Siemens:

<http://support.automation.siemens.com/>

Введите выражение для поиска: Funktionsbausteine ANSI_S5

Сжатый файл ANSI_S5.EXE доступен для загрузки. В файле ANSI_S5.EXE содержится функциональный блок «IEEE:GP».

Активная отправка описана в справочных руководствах ПЛК или КП (коммуникационные процессоры).

2.5 Приложение

2.5.1 Приложение

Разделы в приложении

Это приложение состоит из следующих разделов.

- Инструкции SQL для настройки критериев сортировки и фильтрации пользовательских архивов.
- Алфавитный список ключевых слов SQL, которые не должны использоваться в именах архивов или полей в пользовательских архивах.
- Типовое применение, особенно производительность при записи и чтении тегов и открытии экрана.
- На что обратить внимание

2.5.2 Язык SQL

SQL (Structured Query Language – язык структурированных запросов) – это распространенный язык баз данных. В функциях языка сценариев WinCC язык SQL используется для заданий баз данных. Подробную информацию см. в специализированных руководствах.

Для некоторых стандартных функций, а также некоторых функций в редакторе User Archives (Пользовательские архивы) необходимо определить условия на языке базы данных SQL для спецификации редактируемых записей данных. Ниже приводятся несколько примеров представления инструкции SQL.

- FieldA > '1992-12-31 23:45:12.12'. Оператор выбирает все записи данных, в которых значение в столбце FieldA больше введенного. FieldA представляет собой тип данных DB_TYP_TIME.
- FieldB like 'Cauldron%'. Используется, например для выбора записей данных, у которых в столбце FieldB есть значение Cauldron1, Cauldron4, Cauldron12. FieldB представляет собой тип данных DB_TYP_CHAR.
- FieldC > 100. В соответствии с условием этой формы выбираются все записи, у которых в столбце FieldC значение больше 100. FieldC представляет собой тип данных DB_TYP_INTEGER
- BETWEEN FieldC = 20 AND FieldC = 200. Оператор выбирает все записи данных, значение которых в столбце FieldC находится в диапазоне от 20 до 200. FieldC представляет собой тип данных DB_TYP_INTEGER
- FieldD. Сортировка происходит по столбцу FieldD
- FieldE desc. Сортировка происходит по столбцу FieldE в обратном алфавитном порядке (по убыванию).

2.5.3 Алфавитный список ключевых слов SQL

Ключевые слова SQL

Имена архивов, представлений и полей могут состоять только из букв, цифр и символов подчеркивания «_». Число символов не должно превышать 25. Первым символом всегда должна быть буква.

В именах архивов, представлений и полей нельзя использовать следующие термины:

- Archives (архивы);
- View (представление);
- Field (поле);
- ViewCol (столбец представления);
- все ключевые слова SQL.

Ключевые (или зарезервированные) слова языка SQL базы данных нельзя использовать в именах архивов, представлений и полей в пользовательских архивах. Ниже приводятся ключевые слова SQL.

Ключевые слова, использующиеся в языке SQL			
add	all	alter	and
any	as	asc	begin
between	binary	break	by
call	cascade	cast	char
char_convert	character	check	checkpoint
close	comment	commit	connect
constraint	continue	convert	create
cross	current	cursor	date
dba	dbspace	deallocate	dec

Ключевые слова, используемые в языке SQL			
decimal	declare	default	delete
desc	distinct	do	double
drop	else	elseif	encrypted
end	endif	escape	exception
exec	execute	exists	fetch
first	float	for	foreign
from	full	goto	grant
group	having	holdlock	identified
if	in	index	inner
inout	insert	instead	int
integer	into	is	isolation
join	key	left	like
lock	long	match	membership
message	mode	modify	named
natural	noholdlock	not	null
numeric	of	Off	on
open	option	Options	or
order	others	out	outer
passthrough	precision	prepare	primary
Print	privileges	proc	procedure
raiserror	readtext	Real	reference
references	release	remote	rename
resource	restrict	return	revoke
right	rollback	save	savepoint
schedule	select	set	share
smallint	some	sqlcode	sqlstate
start	stop	subtrans	subtransaction
synchronize	syntax_error	table	temporary
then	time	tinyint	to
tran	trigger	truncate	tsequal
union	unique	unknown	update
user	using	validate	values
varbinary	varchar	variable	varying
view	when	where	while
with	work	writetext	

2.5.4 Количественная структура

Среда для тестирования

Измерения, описанные ниже, были выполнены в следующей среде для тестирования:

- Процессор: Pentium III 600/256 МБ
- Соединение: S7 Protocol Suite, каналный блок MPI
- Среда проекта:
 - Объем данных: Три пользовательских архива, у архива 1 – 100 полей и 3000 записей данных, у архива 2 – 200 полей и 1500 записей данных и у архива 3 – 500 полей и 500 записей данных.
 - При измерениях в среде исполнения считывается или записывается первая и последняя строка, инициирование команд записи/чтения в объекте таблицы осуществляется кнопками панели инструментов.
 - Редактор закрыт, С-функции не выполняются, не происходит обмен кадрами.

Теги WinCC и теги необработанных данных

В этих измерениях использовались теги WinCC. Как показывают измерения, время доступа возрастает в больших пользовательских архивах.

Для больших пользовательских архивов рекомендуется использовать теги необработанных данных. Теги необработанных данных передают данные в пакетах и предоставляют более быстрый доступ даже к большим архивам.

Примечание

На один пользовательский архив можно создать до 500 полей.

2.5.5 Производительность чтения и записи тегов

Производительность

В приведенных здесь данных производительности было измерено поведение пользовательских архивов при записи и чтении тегов.

Не забывайте, что производительность зависит от типового приложения.

Число полей	Число записей	Время записи в теги в секундах	Время чтения из тегов в секундах
100	1	1	2
100	10	1	2-3
100	50	1	3-4
100	100	1	3-9
100	1000	1-2	>3 (в зависимости от соединения)
200	1	2	3
200	10	2	4
200	50	1-2	>4
200	100	1-2	>4
200	1000	2-3	>4
500	1	3	4
500	10	3	7
500	50	3-4	Прибл. 15
500	100	4	>15
500	500	4	>15

Время зависит от размера соответствующей таблицы.

2.5.6 Характеристика времени открытия экрана

Время открытия экрана

В таблице ниже приводятся данные одного из измерений производительности времени открытия экрана. В редакторе User Archives (Пользовательские архивы) окно таблицы не активно.

Не забывайте, что время открытия экрана зависит от типового приложения.

Число полей	Число записей	Время открытия экрана в секундах
100	1	1
100	10	2
100	100	3
100	500	3
100	1000	3
100	2000	3
100	3000	3
200	1	1
200	10	2
200	100	4
200	500	4
200	1000	4
200	1500	>4
500	1	3
500	10	4
500	100	>4
500	500	>4

Сценарии пользовательских архивов

3.1 Стандартные функции языка сценариев WinCC

Описание стандартных функций пользовательского архива делится на следующие разделы.

- Общая информация о программировании сценариев.
- Функции для редактирования и отображения пользовательских архивов
- Дескрипторы стандартных функций.
- Практический пример сценария.
- Ссылка стандартных функций для пользовательских архивов
- Подробное описание функций пользовательских архивов можно найти в интерактивной справке по пользовательским архивам WinCC.

WinCC предоставляет несколько стандартных функций, позволяющих пользователю гибко применять пользовательские архивы.

Эти стандартные функции имеют единообразные правила наименования. Все стандартные функции пользовательских архивов начинаются с «ua», например uaConnect, uaArchiveOpen, uaArchiveGetFields и др. Функции среды исполнения пользовательских архивов всегда начинаются с «uaArchive».

Функции делятся на функции конфигурации и функции среды исполнения. Дескрипторы, которые необходимы для функций конфигурации и среды исполнения, возвращаются ранее вызванными функциями uaQueryConfiguration, uaConnect и uaOpen.

Примечание

"ua" в функциях пользовательских архивов всегда пишется в нижнем регистре.

В сценарии необходимо обеспечить актуальность данных.

Если в сценарии был открыт пользовательский архив и с помощью редактора или элемента управления пользовательскими архивами была добавлена или удалена запись, в сценарии не предусматривается уведомление о произведенных изменениях. Уведомление появляется только после повторного запроса.

Дополнительные источники информации

Пример сценария (стр. 93)

Функции для редактирования и отображения пользовательских архивов (стр. 88)

Дескрипторы для конфигурации пользовательских архивов (стр. 90)

Описание стандартных функций языка сценариев WinCC (стр. 101)

3.2 Конфигурация макросов

Выполните следующую процедуру, чтобы осуществить конфигурацию макроса.

1. Запустите графический редактор и создайте кадр системы.
2. Щелкните правой кнопкой мыши на объекте, для которого необходимо добавить макрос (например, кнопка).
3. Выберите свойства
4. Выберите элемент на вкладке Properties (Свойства) или Events (События) и дважды щелкните на необходимом действии (например, для конфигурации макроса по щелчку мыши Press Left (Щелкнуть левой кнопкой) выберите Events/Mouse/Press left (События/Мышь/Щелкнуть левой кнопкой)). В следующем диалоговом окне можно напрямую ввести код C и затем выполнить компиляцию.
5. 5. Нажмите ОК, что завершить конфигурацию макроса.

3.3 Функции для редактирования и отображения пользовательских архивов

Конфигурация со стандартными функциями

Функция `uaQueryConfiguration` предоставляет дескриптор (UAHCONFIG) для функций конфигурации. С помощью этого дескриптора можно вызывать такие функции конфигурации, как `uaSetArchive`, `uaAddArchive`, `uaSetField`, `uaAddField` и др. Функция `uaReleaseConfiguration` завершает конфигурацию.

Установка соединения с пользовательскими архивами

Для получения доступа в среде исполнения необходимо вызвать стандартную функцию `uaConnect`, чтобы установить соединение с компонентом пользовательского архива. Функция `uaConnect` генерирует дескриптор UAHCONNECT, с помощью которого можно открыть пользовательские архивы и представления. Функция `uaDisconnect` прерывает соединение с пользовательскими архивами.

Открытие функций среды исполнения

Сконфигурированный пользовательский архив необходим для работы в среде исполнения. Функции `uaQueryArchive` и `uaQueryArchiveByName` предоставляют дескриптор для функций среды исполнения. Функции среды исполнения пользовательского архива можно использовать после открытия пользовательского архива с помощью функции `uaArchiveOpen`.

Функции для работы в среде исполнения

С помощью функций `uaArchiveNext`, `uaArchivePrevious`, `uaArchiveFirst` и `uaArchiveLast` можно перемещать указатель. Уникальное назначение для записи данных пользовательского архива создается с помощью дескриптора `hArchive`. С этим назначением можно выполнять косвенную адресацию, которая требуется для диалоговых окон.

Функция `uaArchiveUpdate` хранит временную запись данных в пользовательском архиве и переписывает запись, на которую в настоящее время направлен указатель. Эту запись нужно предварительно прочитать с помощью функций `uaArchiveNext`, `uaArchivePrevious`, `uaArchiveFirst` или `uaArchiveLast`.

Завершение соединения с пользовательскими архивами

Функция `uaArchiveClose` закрывает пользовательский архив. Функция `uaReleaseArchive` прерывает соединение с текущим пользовательским архивом, а функция `uaDisconnect` завершает соединение с компонентом пользовательского архива.

Примечание

Соединения с пользовательскими архивами, установленные в сценарии, должны завершиться также в этом сценарии.

Функции, устанавливающие соединение	Функции, завершающие соединение
<code>UaQueryConfiguration</code>	<code>uaReleaseConfiguration</code>
<code>uaConnect</code>	<code>uaDisconnect</code>
<code>uaQueryArchive</code>	<code>uaReleaseArchive</code>
<code>uaQueryArchiveByName</code>	<code>uaReleaseArchive</code>
<code>uaArchiveOpen</code>	<code>uaArchiveClose</code>

Для пользовательских архивов существует два вида вызовов API.

1. С помощью префикса «ua» (строчные буквы) для сценариев (Global Script и программирование макросов).
2. С помощью префикса «UA» (прописные буквы) для программ, работающих вне WinCC.

Если вызовы пользовательских архивов используются в мастере динамики, перед ними должен стоять префикс «UA» (прописные буквы).

3.4 Дескрипторы для конфигурации пользовательских архивов

Дескриптор UAHCONFIG

Функция `uaQueryConfiguration` генерирует дескриптор UAHCONFIG, который необходим для конфигурации пользовательских архивов. Это значит, что сначала необходимо вызвать функцию `uaQueryConfiguration`, чтобы получить дескриптор UAHCONFIG. Вместе с этим дескриптором можно вызывать функции, перечисленные ниже. И, наконец, чтобы завершить конфигурацию, необходимо вызвать `uaReleaseConfiguration`.

Дескрипторы для конфигурации пользовательских архивов	
<code>UaQueryConfiguration</code>	---> дескриптор UAHCONFIG
	требуется функциям...
	<code>uaAddArchive</code>
	<code>uaAddField</code>
	<code>uaGetArchive</code>
	<code>uaGetField</code>
	<code>uaGetNumArchives</code>
	<code>uaGetNumFields</code>
	<code>uaReleaseConfiguration</code>
	<code>uaRemoveAllArchives</code>
	<code>uaRemoveAllFields</code>
	<code>uaRemoveArchive</code>
	<code>uaRemoveField</code>
	<code>uaSetArchive</code>
	<code>uaSetField</code>

3.5 Дескрипторы функций среды исполнения

Дескриптор UAHCONNECT

Функцией пользовательских архивов `uaConnect` генерируется дескриптор UAHCONNECT, который необходим для открытия и закрытия пользовательских архивов и представлений. Это значит, что сначала должна вызываться функция `uaConnect`, чтобы получить дескриптор UAHCONNECT. Вместе с этим дескриптором также можно вызывать функции, перечисленные ниже, для открытия и закрытия архивов и представлений. И, наконец, чтобы завершить конфигурацию, необходимо вызвать `uaDisconnect`.

Функции `uaQueryArchive` и `uaQueryArchiveByName` генерируют дескриптор UAHARCHIVE. Этот дескриптор требуется для функции `uaArchiveOpen`, которая открывает пользовательский архив для работы в среде исполнения. Для завершения соединения необходимо вызвать функции `uaRelease` и `uaArchiveClose`.

Примечание

Сортировка и фильтрация пользовательских архивов

Функции uaArchiveSetSort и uaArchiveSetFilter можно использовать для пользовательского архива, не открывая пользовательский архив с помощью uaArchiveOpen.

Дескрипторы функций среды исполнения		
uaConnect	-> Дескриптор UAHCONNECT	
	требуется функциям...	
	uaDisconnect	
	uaQueryArchive	--> дескриптор UAHARCHIVE
	uaQueryArchiveByName	--> дескриптор UAHARCHIVE
		требуется функциям...
		uaArchiveOpen
		Необходим для:
		uaArchiveClose
		uaArchiveDelete
		uaArchiveExport
		uaArchiveGetCount
		uaArchiveGetFieldLength
		uaArchiveGetFields
		uaArchiveGetFieldType
		uaArchiveGetFieldValueDate
		uaArchiveGetFieldValueDouble
		uaArchiveGetFieldValueFloat
		uaArchiveGetFieldValueLong
		uaArchiveGetFieldValueString
		uaArchiveGetFieldName
		uaArchiveGetFilter
		uaArchiveGetID
		uaArchiveGetName
		uaArchiveGetSort
		uaArchiveImport
		uaArchiveInsert
		uaArchiveMoveFirst
		uaArchiveMoveLast
		uaArchiveMoveNext
		uaArchiveMovePrevious
		uaArchiveReadTagValues
		uaArchiveReadTagValuesByName

Дескрипторы функций среды исполнения		
		uaArchiveRequery
		uaArchiveSetFieldValueDate
		uaArchiveSetFieldValueDouble
		uaArchiveSetFieldValueFloat
		uaArchiveSetFieldValueLong
		uaArchiveSetFieldValueString
		uaArchiveSetFilter*
		uaArchiveSetSort*
		uaArchiveUpdate
		uaArchiveWriteTagValues
		uaArchiveWriteTagValuesByName
		uaReleaseArchive
*: Функции можно вызывать только перед uaArchiveOpen.		

3.6 Пример сценария

	ID	Date	Recipes	Water	Sugar	Coloring 7	Caffeine	Ph
1	1	12:30:15	Calif Coke	90	10	10	10	15
2	2		Coke	80	30	15	20	20
3	3		Standard	100	50	20	20	20
4	4		Coke Light	100	20	30	20	20
5	5		CherryCo	100	80	50	5	10
6	6	30.04.99	KidsCoke	70	30	40	20	30

Global Script - Diagnose		
Data of Field 5: Field Type = Integer Field Value = 0 Field Length = 4		
Data of Field 6: Field Type = Integer Field Value = 0 Field Length = 4		
Data of Field 7: Field Type = Integer Field Value = 0 Field Length = 4		
Data of Field 8: Field Type = String Field Length = 10		
Data of Field 9: Field Type = Date & Time Jahr 1999 Monat 04 Tag 07 Field Length = 16		
		07.05.99 15:04:03

В следующем разделе содержится пример с двумя стандартными функциями для чтения и записи пользовательского архива в среде исполнения. Функция `UAReadFromArchive` считывает пользовательский архив Cola и отображает считанные данные в окне Global Script Diagnostics Control. Функция `UAWriteToArchive` записывает пользовательский архив и отображает состояния и сообщения. Окно диагностики создается путем размещения OLE Control (из Object Palette (Палитра объектов) -> Smart Objects (Интеллектуальные объекты)) в графический дизайнер и последующем выборе WinCC Global Script – Diagnostics Control из диалогового окна Insert OLE Control (OCX).

В графическом дизайнерах создайте для проекта новый экран. В экране создайте кнопки Read Archive (Читать архив) и Write Archive (Записать архив) и назначьте их стандартным функциям, описанным ниже. Выполните следующие действия.

3.6 Пример сценария

1. Откройте окно выбора Object Palette (Палитра объектов) и выберите Button (Кнопка) в Windows Objects (Объекты Windows).
2. Поместите кнопку в рабочую зону графического дизайнера и, удерживая кнопку мыши нажатой, измените размер.
3. Щелкните правой кнопкой мыши на только что созданной кнопке и в контекстном меню выберите Properties (Свойства). На вкладке Properties (Свойства) можно задать название кнопки и цвет. На кнопке может быть написано, например Read Archive (Читать архив) и Write Archive (Записать архив).
4. На вкладке Events (События) назначьте действие мыши, выбрав Mouse (Мышь) и дважды щелкнув на Mouse Action (Действие мыши). Отобразится редактор сценариев. Введите стандартную функцию сценария UAReadFromArchive, перечисленную ниже.

```

#include "apdefap.h"

void UAReadFromArchive()
{
    UAHCONNECT hConnect = 0;
    UAHARCHIVE hArchive = 0;
    LONG IndexArchive;
    LONG FieldLength;
    LONG FieldType,
    LONG NumberOfFields;
    LONG Index;
    long IntValue;
    double DoubleValue;
    char ArchiveName[255], StringField[255];
    SYSTEMTIME SysDate;

    //***** Подключение к компоненту пользовательского архива *****
    if (uaConnect( &hConnect ) == FALSE )
    {
        printf("uaConnect error: %d\n", uaGetLastError() );
        return;
    }
    if (hConnect == NULL)
    {
        printf("Handle UAHCONNECT equals 0\n" );
        return;
    }

    //***** Подключение к архиву через имя архива *****
    if (uaQueryArchiveByName( hConnect, "Cola", &hArchive ) == FALSE )
    {

```

printf("uaQueryArchive Error: %d\n", uaGetLastError());
goto finish;
}
//***** Открытие архива *****
if (uaArchiveOpen(hArchive) == FALSE)
{
printf("uaArchive Open Error\n");
goto finish;
}
//***** Переход к первой записи *****
if (uaArchiveMoveFirst(hArchive) == FALSE)
{
printf("uaArchiveMoveFirst Error = %u\n");
goto finish;
}
//***** Получение числа полей *****
NumberOfFields = uaArchiveGetFields(hArchive);
printf("Number of Fields = %u\n", NumberOfFields);
//***** Считывание и отображение полей данных *****
for (Index = 1; Index < NumberOfFields; Index++)
{
printf("Data of Field %u: \n", Index);
FieldType = uaArchiveGetFieldType(hArchive, Index);
switch (FieldType)
{
case UA_FIELDTYPE_INTEGER :
printf("Field Type = Integer\n");
if (uaArchiveGetFieldValueLong (hArchive, Index, &IntValue) == TRUE)
printf("Field Value = %u\n", IntValue);
else
printf("Error callinguaArchiveGetFieldValueLong: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_DOUBLE :
printf("Field Type = Double\n");
if (uaArchiveGetFieldValueDouble (
hArchive, Index, &DoubleValue) == TRUE)

3.6 Пример сценария

```

printf("Field Value = %g\n", DoubleValue );
    else
printf("Error calling uaArchiveGetFieldValueDouble: %d\n", uaGetLastError() );

break;

case UA_FIELDTYPE_STRING :
printf("Field Type = String\n");
if (uaArchiveGetFieldValueString ( hArchive, Index, StringField, 20 ) == TRUE )
printf("Field Value = %s\n", StringField );
    else
printf("Error callinguaArchiveGetFieldValueString: %d\n", uaGetLastError() );

break;

case UA_FIELDTYPE_DATETIME :
printf("Field Type = Date & Time\n");
if (uaArchiveGetFieldValueDate ( hArchive, Index, &SysDate ) == TRUE )
printf("%d.%d.%d\n ",SysDate.wDay, SysDate.wMonth, SysDate.wYear );
    else
printf("Error calling uaArchiveGetFieldValueLong: %d\n", uaGetLastError() );

break;

case -1 :
default:
printf("Error executing uaArchiveGetFieldType\n");
}

//***** Считывание и отображение длины полей данных *****
FieldLength = uaArchiveGetFieldLength( hArchive, Index );
if ( FieldLength != -1 )
printf("Field Length = %u\n", FieldLength );
    else
printf("Error executing uaArchiveGetFieldLength\n");
}

//***** Закрытие всех дескрипторов и соединений *****
finish:;
//***** Закрытие архива *****
if( NULL != hArchive )
{
if (uaArchiveClose ( hArchive ) == FALSE )

```

```

{
printf("error on closing archive\n" );
}
}
//***** Разрыв соединения с архивом *****
if( NULL != hArchive )
{
if (uaReleaseArchive ( hArchive ) == FALSE )
{
printf("error on releasing archive\n" );
}
hArchive = 0;
}
//***** Отключение от компонента пользовательского архива *****
if( NULL != hConnect )
{
if (uaDisconnect ( hConnect ) == FALSE )
{
printf("error on disconnection\n" );
}
hConnect = 0;
}
}

```

Создайте вторую кнопку для описания пользовательского архива. Следуйте процедуре создания первой кнопки. Выберите стандартную функцию `UAWriteToArchive` и введите следующий сценарий.

```

void UAWriteToArchive()
{
UAHCONNECT hConnect = 0;
UAHARCHIVE hArchive = 0;
LONG IndexArchive;
LONG FieldLength;
LONG FieldType,
LONG NumberOfFields;
LONG Index;
long IntValue;
double DoubleValue;
char StringField[255];
SYSTEMTIME SysDate;
}

```

3.6 Пример сценария

```

//***** Подключение к компоненту пользовательского архива *****
if (uaConnect( &hConnect ) == FALSE )
{
printf("uaConnect error: %d\n", uaGetLastError() );
return;
}
if (hConnect == NULL)
{
printf("Handle UAHCONNECT equals NULL\n" );
return;
}

//***** Подключение к архиву через имя архива *****
if (uaQueryArchiveByName( hConnect, "Cola", &hArchive ) == FALSE )
{
printf("uaQueryArchive Error: %d\n", uaGetLastError() );
goto finish;
}

//***** Открытие архивов *****
if ( uaArchiveOpen( hArchive ) == FALSE )
{
printf("uaArchive Open Error\n" );
goto finish;
}

//***** Получение числа полей *****
NumberOfFields = uaArchiveGetFields( hArchive );
printf("Number of Fields = %u\n", NumberOfFields );

//***** Считывание последней записи данных *****
if (uaArchiveMoveLast( hArchive ) == TRUE )
printf("Number of Fields = %u\n", NumberOfFields );
else
{
printf("uaArchiveMoveLast Error: %d\n", uaGetLastError() );
goto finish;
}

//***** Запись в поля данных *****
IntValue = 32;
DoubleValue = 64;
    
```

strcpy(StringField, "Text12");
GetSystemTime(&SysDate);
for (Index = 1; Index < NumberOfFields; Index++)
{
printf("Data of Field %u: \n", Index);
FieldType = uaArchiveGetFieldType(hArchive, Index);
switch (FieldType)
{
case UA_FIELDTYPE_INTEGER :
printf("Field Type = Integer\n");
if (uaArchiveSetFieldValueLong (hArchive, Index, IntValue) == TRUE)
printf("Field Value = %u\n", IntValue);
else
printf("Error calling uaArchiveSetFieldValueLong: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_DOUBLE :
printf("Field Type = Double\n");if (uaArchiveSetFieldValueDouble (hArchive, Index, DoubleValue)
== TRUE)
printf("Field Value = %g\n", DoubleValue);
else
printf("Error calling uaArchiveSetFieldValueDouble: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_STRING :
printf("Field Type = String\n");
if (uaArchiveSetFieldValueString (hArchive, Index, StringField) == TRUE)
printf("Field Value = %s\n", StringField);
else
printf("Error calling uaArchiveSetFieldValueString: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_DATETIME :
printf("Field Type = Date & Time\n");
if (uaArchiveSetFieldValueDate (hArchive, Index, &SysDate) == TRUE)
printf("%d.%d.%d\n ",SysDate.wDay, SysDate.wMonth, SysDate.wYear);
else

3.6 Пример сценария

```

printf("Error calling uaArchiveGetFieldValueLong: %d\n", uaGetLastError() );
break;
case -1 :
default:
printf("Error executing uaArchiveGetFieldType\n");
}

FieldLength = uaArchiveGetFieldLength( hArchive, Index );
if ( FieldLength != -1 )
printf("Field Length = %u\n", FieldLength );
else
printf("Error executing uaArchiveGetFieldLength\n");
}

//***** Обновление архива *****
if (uaArchiveUpdate(hArchive) == FALSE )
{
printf("uaArchiveUpdate Error:\n" );
}

//***** Закрытие всех дескрипторов и соединений *****
finish;;

//***** Закрытие архива *****
if( NULL != hArchive )
{
if (uaArchiveClose ( hArchive ) == FALSE )
{
printf("error on closing archive\n" );
}
}

//***** Разрыв соединения с архивом *****
if( NULL != hArchive )
{
if (uaReleaseArchive ( hArchive ) == FALSE )
{
printf("error on releasing archive\n" );
}
}
hArchive = 0;
}

//***** Отключение от компонента пользовательского архива *****
if( NULL != hConnect )
{

```

```
if (uaDisconnect ( hConnect ) == FALSE )
{
printf("error on disconnection\n" );
}
hConnect = 0;
}
}
```

Затем можно закрыть диалоговое окно и запустить среду исполнения. Действия макросов можно контролировать в Global Script Diagnostics Window (Окно Global Script Diagnostics).

3.7 Приложение

3.7.1 Описание стандартных функций языка сценариев WinCC

Язык сценариев WinCC

В следующем разделе содержится подробное описание функций языка сценариев WinCC.

Функции делятся на четыре группы.

- Стандартные функции для пользовательских архивов
- Функции для конфигурации пользовательских архивов
- Основные функции среды исполнения
- Функции среды исполнения для архивов

Эти функции приводятся по группам в алфавитном порядке.

При создании записи данных не выполняется проверки на завершенность и правильность. В частности, поле не должно быть пустым.

Дополнительные источники информации

Стандартные функции языка сценариев WinCC (стр. 87)

3.7.2 Стандартные функции для пользовательских архивов

3.7.2.1 uaGetLastError

Описание:

Функциями языка сценариев WinCC возвращается значение BOOL, где TRUE соответствует безошибочной обработке данных. При возвращении FALSE ошибку последней функции можно прочитать с помощью uaGetLastError() и uaGetLastHResult().

Если функция uaGetLastError() вызывается после запуска нескольких функций, uaGetLastError() вернет ошибку, возникшую последней. Чтобы знать, в какой функции возникла ошибка, uaGetLastError() и uaGetLastHResult() необходимо вызывать всякий раз, когда функция возвращает FALSE.

Пример

```
if ( uaArchiveGetFieldValueLong ( hArchive, Index, &IntValue ) ==
TRUE )
printf( "Field Value = %u\n", IntValue );
else
printf("Error calling uaArchiveGetFieldValueLong: %d / %08lx\n",
uaGetLastError(), uaGetLastHResult());
```

Если функции не возвращают значение (VOID), то в любом случае необходимо отправить запрос через uaGetLastError().

Пример

```
uaArchiveGetFilter(hArchive, pszFilter, cMaxLen);
INT nUAError = uaGetLastError ( );
if ( UA_ERROR_SUCCESS != nUAError)
{
printf( "Filter = [%s]\n", pszFilter );
}
else
{
printf("Error calling uaArchiveGetFilter: %d, hr=0x%08lX\n",
nUAError, uaGetLastHResult());
}

INT uaGetLastError()
```

Возвращаемое значение:

Состояние ошибки последней выполняемой функции. Функция `uaGetLastError()` может возвращать следующие ошибки:

UA_ERROR_SUCCESS
UA_ERROR_GENERIC
UA_ERROR_CONNECT_FAILED
UA_ERROR_OPEN_FAILED
UA_ERROR_CLOSE_FAILED
UA_ERROR_REQUERY_FAILED
UA_ERROR_MOVE_FAILED
UA_ERROR_INSERT_FAILED
UA_ERROR_UPDATE_FAILED
UA_ERROR_DELETE_FAILED
UA_ERROR_IMPORT_FAILED
UA_ERROR_EXPORT_FAILED
UA_ERROR_READ_FAILED
UA_ERROR_WRITE_FAILED
UA_ERROR_GET_FAILED
UA_ERROR_SET_FAILED
UA_ERROR_INVALID_NAME
UA_ERROR_INVALID_TYPE
UA_ERROR_INVALID_NUMRECS
UA_ERROR_INVALID_COMMTYPE
UA_ERROR_INVALID_LENGTH
UA_ERROR_INVALID_PRECISION
UA_ERROR_NULL_POINTER
UA_ERROR_INVALID_POINTER
UA_ERROR_INVALID_HANDLE
UA_ERROR_INVALID_INDEX
UA_ERROR_SERVER_UNKNOWN

Эти константы ошибок, а также предустановки пользовательского архивов, находятся в `CCUACAPI.H`.

3.7.2.2 uaGetLastHResult

Описание:

Считывает ошибку COM, возникшую последней. Эта функция служит в основном для диагностики несовместимости при применении COM или для обнаружения ошибок регистрации и связи.

Данная функция в основном используется как приложение к функции UAGetLastError, если функция пользовательского архива (например, uaConnect) выдает сигнал об ошибке с FALSE.

```
LONG uaGetLastHResult()
```

Возвращаемое значение:

Ошибка COM, возникшая последней

3.7.3 Функции для конфигурации пользовательских архивов

3.7.3.1 Функции для конфигурации пользовательских архивов

Функции для конфигурации

Эти функции служат для конфигурации пользовательских архивов.

Функция	Описание
uaAddArchive	Добавляет новый пользовательский архив
uaAddField	Добавляет новое поле
uaGetArchive	Считывает конфигурацию архива
uaGetField	Считывает конфигурацию поля
uaGetNumArchives	Обнаруживает число сконфигурированных архивов
uaGetNumFields	Обнаруживает число полей
UaQueryConfiguration	Устанавливает соединение с конфигурацией пользовательского архива
uaReleaseConfiguration	Завершает соединение с конфигурацией
uaRemoveAllArchives	Удаляет все архивы
uaRemoveAllFields	Удаляет все поля
uaRemoveArchive	Удаляет архив
uaRemoveField	Удаляет поле
uaSetArchive	Записывает конфигурацию архива
uaSetField	Устанавливает конфигурацию поля

3.7.3.2 uaAddArchive

Описание:

Добавляет новый пользовательский архив. Это соответствует конфигурации нового пользовательского архива с помощью редактора пользовательских архивов.

```
LONG uaAddArchive (  
    UAHCONFIG hConfig,  
    UACONFIGARCHIVE* pArchive )
```

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации для пользовательского архива. Этот дескриптор генерируется `uaQueryConfiguration`.

UACONFIGARCHIVE* pArchive

Указатель на буфер для хранения конфигурации пользовательского архива.

Возвращаемое значение:

Индекс нового пользовательского архива. Ошибка соответствует "-1".

3.7.3.3 uaAddField

Описание:

Добавляет новое поле данных.

```
LONG uaAddField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGFIELD* pField )
```

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется `uaQueryConfiguration`.

long lArchive,

Индекс архива (от 0 до (`uaGetNumArchives()-1`))

UACONFIGFIELD* pArchive

Указатель на буфер конфигурации поля.

Возвращаемое значение:

Индекс нового поля. Значение "-1" указывает на ошибку.

Дополнительные источники информации

Структура программирования дескриптора uaCONFIGIELDA (стр. 113)

Структура программирования дескриптора uaCONFIGIELDW (стр. 113)

3.7.3.4 uaGetArchive

Описание:

Считывает конфигурацию пользовательского архива.

```
BOOL uaGetArchive (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGARCHIVE* pArchive )
```

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

long lArchive,

Индекс архива (от 0 до (uaGetNumArchives()-1))

UACONFIGARCHIVE* pArchive

Указатель на буфер для получения конфигурации пользовательского архива.

Возвращаемое значение:

TRUE (Истина): доступ к пользовательскому архиву был получен

FALSE (Ложь): Error (Ошибка)

3.7.3.5 uaGetField

Описание:

Считывает конфигурацию поля.

```
BOOL uaGetField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    long lField,  
    UACONFIGFIELD* pField )
```

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

long lArchive,

Индекс архива (от 0 до (uaGetNumArchives()-1))

long lField,

Номер поля, где при lField = 0 выполняется адресация первого поля.

UACONFIGFIELD* pArchive

Указатель на буфер для получения конфигурации поля.

Возвращаемое значение:

TRUE (Истина): доступ к пользовательскому архиву был получен

FALSE (Ложь): Error (Ошибка)

3.7.3.6 uaGetNumArchives**Описание:**

Считывает число сконфигурированных на данный момент пользовательских архивов.

LONG uaGetNumArchives (

UAHCONFIG hConfig)

Параметры:

UAHCONFIG hConfig

Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

Возвращаемое значение:

число сконфигурированных на данный момент пользовательских архивов. При возникновении ошибки вернется -1.

3.7.3.7 uaGetNumFields**Описание:**

Предоставляет число сконфигурированных полей. Поля ID (Идентификатор), Last User (Последний пользователь) и Last Access (Последний доступ) не включены. В вызовах конфигурации указываются индексы от 0 до uaGetNumFields() -1.

LONG uaGetNumFields (

UAHCONFIG hConfig,

long lArchive)

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

long lArchive,

Индекс архива (от 0 до (uaGetNumArchives()-1))

Возвращаемое значение:

число настроенных полей. При возникновении ошибки вернется -1.

3.7.3.8 UaQueryConfiguration

Описание:

Устанавливает соединение с пользовательским архивом для конфигурации.

```
BOOL uaQueryConfiguration (  
UAHCONFIG* phConfig )
```

Параметры:

UAHCONFIG* phConfig,

Указатель на дескриптор архива.

Возвращаемое значение:

TRUE (Истина): доступ к пользовательскому архиву был получен

FALSE (Ложь): Error (Ошибка)

3.7.3.9 uaReleaseConfiguration

Описание:

Прерывает соединение с пользовательскими архивами (конфигурация).

```
BOOL uaReleaseConfiguration (  
UAHCONFIG hConfig,  
BOOL bSave )
```

Параметры:

UAHCONFIG hConfig

Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

BOOL bSave

Сохраняет изменения, выполненные в конфигурации до завершения соединения с пользовательским архивом.

TRUE = сохранение изменений, FALSE = отказ от внесенных изменений

Внимание! Сохранить изменения (bSave = TRUE) можно только при выключенной среде исполнения! Можно проверить, включена ли среда исполнения, запросив ualsActive().

Возвращаемое значение:

TRUE (Истина): успешное отключение соединения

FALSE (Ложь): Error (Ошибка)

3.7.3.10 uaRemoveAllArchives**Описание:**

Удаляет все пользовательские архивы, не используемые на данный момент в представлениях.

BOOL uaRemoveAllArchives
(UAHCONFIG hConfig)

Параметры:

UAHCONFIG hConfig

Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

Возвращаемое значение:

TRUE (Истина): успешное удаление

FALSE (Ложь): Error (Ошибка)

Comment (Комментарий)

После этого можно отправить запрос с помощью uaGetNumArchives(), чтобы узнать, были ли удалены все архивы.

3.7.3.11 uaRemoveAllFields

Описание:

Удаляет все поля.
BOOL uaRemoveAllFields (
UAHCONFIG hConfig,
long lArchive)

Параметры:

UAHCONFIG hConfig,
Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.
long lArchive,
Индекс архива (от 0 до (uaGetNumArchives()-1))

Возвращаемое значение:

TRUE (Истина): успешное удаление полей
FALSE (Ложь): Error (Ошибка)

3.7.3.12 uaRemoveArchive

Описание:

uaRemoveArchive удаляет сконфигурированный пользовательский архив целиком.
BOOL uaRemoveArchive (
UAHCONFIG hConfig,
long lArchive)

Параметры:

UAHCONFIG hConfig,
Дескриптор конфигурации пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.
long lArchive,
Индекс архива (от 0 до (uaGetNumArchives()-1))

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно удален
FALSE (Ложь): Error (Ошибка)

3.7.3.13 uaRemoveField

Описание:

Удаляет поле.

```
BOOL uaRemoveField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    long lField )
```

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации для пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

long lArchive,

Индекс архива (от 0 до (uaGetNumArchives()-1))

long lField,

Номер поля, где при lField = 0 выполняется адресация первого поля.

Возвращаемое значение:

TRUE (Истина): поле было успешно удалено

FALSE (Ложь): Error (Ошибка)

3.7.3.14 uaSetArchive

Описание:

Определяет конфигурацию пользовательского архива.

```
BOOL uaSetArchive (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGARCHIVE* pArchive  
    )
```

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации для пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

long lArchive,

Индекс архива (от 0 до (uaGetNumArchives()-1))

UACONFIGARCHIVE* pArchive

Указатель на буфер pArchive с конфигурацией пользовательского архива.

Возвращаемое значение:

TRUE (Истина): доступ к пользовательскому архиву был получен

FALSE (Ложь): Error (Ошибка)

3.7.3.15 uaSetField

Описание:

Устанавливает конфигурацию поля.

```
BOOL uaSetField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    long lField,  
    UACONFIGFIELD* pField )
```

Параметры:

UAHCONFIG hConfig,

Дескриптор конфигурации для пользовательского архива. Этот дескриптор генерируется uaQueryConfiguration.

long lArchive,

Индекс архива (от 0 до (uaGetNumArchives()-1))

long lField,

Номер поля, где при lField = 0 выполняется адресация первого поля.

UACONFIGFIELD* pField

Указатель на буфер настройки поля.

Возвращаемое значение:

TRUE (Истина): доступ к пользовательскому архиву был получен

FALSE (Ложь): Error (Ошибка)

3.7.3.16 Структура программирования дескриптора uaCONFIGIELDA

Структура uaCONFIGIELDA

```
typedef struct tagUACONFIGIELDA
{
    LONG lArchiveId; // Уникальный идентификатор архива
    LONG lArchiveId; // Уникальный идентификатор архива
    LONG lPosition; // Положение (последовательность) архивов
    CHAR szName[UA_MAXLEN_NAME+1]; // Имя архива не больше 20 символов
    CHAR szAlias[UA_MAXLEN_ALIAS+1]; // Имя заголовка не больше 50 символов
    LONG lType; //Тип архива
    LONG lLength; /* Максимальное число символов, если поле данных типа STRING; в
    противном случае не используется */
    LONG lPrecision; // Используется внутри; не нужно вносить
    CHAR szMinValue[UA_MAXLEN_VALUE+1]; /* Минимум для полей данных, кроме
    типа STRING или DATE; в противном случае не используется */
    CHAR szMaxValue[UA_MAXLEN_VALUE+1]; /* Максимум для полей данных, кроме
    типа STRING или DATE; в противном случае не используется */
    CHAR szStartValue[UA_MAXLEN_VALUE+1]; // Начальное значение
    CHAR szDMVarName[UA_MAXLEN_DMVARNAME+1]; /* Тег из менеджера данных
    (используется для архивов со связью через теги WinCC) */
    DWORD dwReadRight; // Права доступа для чтения
    DWORD dwWriteRight; // Права доступа для записи
    DWORD dwFlags; // Последний доступ
} UACONFIGIELDA;
```

Дополнительные источники информации

uaAddField (стр. 105)

3.7.3.17 Структура программирования дескриптора uaCONFIGIELDW

Структура uaCONFIGIELDW

```
typedef struct tagUACONFIGIELDA
{
    LONG lArchiveId; // Уникальный идентификатор архива
    LONG lArchiveId; // Уникальный идентификатор архива
    LONG lPosition; // Положение (последовательность) архивов
    WCHAR wszName[UA_MAXLEN_NAME+1]; // Имя архива не больше 20 символов
```

```

WCHAR wszAlias[UA_MAXLEN_ALIAS+1]; // Имя заголовка не больше 50 символов
LONG lType; //Тип архива
LONG lLength; /* Максимальное число символов, если поле данных типа STRING; в
противном случае не используется */
LONG lPrecision; // Используется внутри; не нужно вносить
WCHAR wszMinValue[UA_MAXLEN_VALUE+1]; /* Минимум для полей данных, кроме
типа STRING или DATE, в противном случае не используется */
WCHAR wszMaxValue[UA_MAXLEN_VALUE+1]; /* Максимум для полей данных, кроме
типа STRING или DATE; в противном случае не используется */
WCHAR wszStartValue[UA_MAXLEN_VALUE+1]; // Начальное значение
WCHAR wszDMVarName[UA_MAXLEN_DMVARNAME+1]; /* Тег из менеджера данных
(используется в архивах для связи через теги WinCC) */
DWORD dwReadRight; // Права доступа для чтения
DWORD dwWriteRight; // Права доступа для записи
DWORD dwFlags; // Последний доступ
} UA_CONFIGFIELD

```

Дополнительные источники информации

uaAddField (стр. 105)

3.7.3.18 Структура программирования дескриптора uaAddArchive

Структура uaAddArchive

```

typedef struct tagUAADDARCHIVE
{
LONG lArchiveId; // Уникальный идентификатор архива
LONG lPosition; // Положение (последовательность) архивов
CHAR szName[UA_MAXLEN_NAME+1]; // Имя архива не больше 20 символов
CHAR szAlias[UA_MAXLEN_ALIAS+1]; // Имя заголовка не больше 50 символов
LONG lType;UA_ARCHIVETYPE_UNLIMITED // Неограниченный тип архива
UA_ARCHIVETYPE_LIMITED // Ограниченный тип архива
LONG lNumRecs; // макс. число наборов данных
LONG lCommType;
UA_COMMTYPE_NONE // отсутствие связи
UA_COMMTYPE_RAW // Связь через необработанные данные
UA_COMMTYPE_DIRECT // Связь через теги DM
CHAR szPLCID[UA_MAXLEN_PLCID+1]; // PLCID тега необработанных данных
CHAR szDMVarName[UA_MAXLEN_DMVARNAME+1]; // Имя тега необработанных
данных

```

```

CHAR szIDVar[UA_MAXLEN_DMVARNAME+1]; // Идентификатор управляющего тега
CHAR szJobVar[UA_MAXLEN_DMVARNAME+1]; // Задание управляющего тега
CHAR szFieldVar[UA_MAXLEN_DMVARNAME+1]; // Поле управляющего тега
CHAR szValueVar[UA_MAXLEN_DMVARNAME+1]; // Значение управляющего тега
DWORD dwReadRight; // Права доступа для чтения
DWORD dwWriteRight; // Права доступа для записи
DWORD dwFlags; UA_ARCHIVEFLAG_ACCESS // Последний доступ
UA_ARCHIVEFLAG_USER // Последний пользователь
} UAADDARCHIVE;

```

Примечание**LONG lArchiveId; // Уникальный идентификатор архива**

При значении: 0, автоматически используется уникальный идентификатор, который возвращается AddArchive. Этот возвращаемый идентификатор затем применяется для AddField (когда -1 возвращается как идентификатор, то создать архив невозможно).

3.7.3.19 Структура программирования дескриптора uaCONFIGARCHIVEA**Структура uaCONFIGARCHIVE**

```

typedef struct tagUACONFIGARCHIVE
{
LONG lArchiveId; // Уникальный идентификатор архива
LONG lPosition; // Положение (последовательность) архивов
CHAR szName[UA_MAXLEN_NAME+1]; // Имя архива не больше 20 символов
CHAR szAlias[UA_MAXLEN_ALIAS+1]; // Имя заголовка не больше 50 символов
LONG lType; UA_ARCHIVE_TYPE_UNLIMITED // Неограниченный тип архива
UA_ARCHIVE_TYPE_LIMITED // Ограниченный тип архива
LONG lNumRecs; // макс. число наборов данных
LONG lCommType;
UA_COMMTYPE_NONE // отсутствие связи
UA_COMMTYPE_RAW // Связь через необработанные данные
UA_COMMTYPE_DIRECT // Связь через теги DM
CHAR szPLCID[UA_MAXLEN_PLCID+1]; // PLCID тега необработанных данных
CHAR szDMVarName[UA_MAXLEN_DMVARNAME+1]; // Имя тега необработанных
данных
CHAR szIDVar[UA_MAXLEN_DMVARNAME+1]; // Идентификатор управляющего тега
CHAR szJobVar[UA_MAXLEN_DMVARNAME+1]; // Задание управляющего тега

```

3.7 Приложение

```

CHAR szFieldVar[UA_MAXLEN_DMVARNAME+1]; // Поле управляющего тега
CHAR szValueVar[UA_MAXLEN_DMVARNAME+1]; // Значение управляющего тега
DWORD dwReadRight; // Права доступа для чтения
DWORD dwWriteRight; // Права доступа для записи
DWORD dwFlags; UA_ARCHIVEFLAG_ACCESS // Последний доступ
UA_ARCHIVEFLAG_USER // Последний пользователь
} UA_CONFIGARCHIVE;
    
```

Примечание

LONG IArchiveId; //Уникальный идентификатор архива

При значении: 0, автоматически используется уникальный идентификатор, который возвращается ConfigArchive. Этот возвращаемый идентификатор затем применяется для ConfigField (когда -1 возвращается как идентификатор, то создать архив невозможно).

3.7.4 Основные функции среды исполнения

3.7.4.1 Основные функции среды исполнения

Функции пользовательских архивов

С помощью этих функций открываются и закрываются пользовательские архивы и представления для работы в среде исполнения.

Функция	Описание
uaConnect	Установка соединения с пользовательскими архивами. Это соединение действует для всех пользовательских архивов в среде исполнения.
uaDisconnect	Если соединение с пользовательскими архивами (среда исполнения) существует, оно будет прервано.
uaGetLocalEvents	Считывает локальные события.
uaIsActive	Определяет, запущена ли среда исполнения.
uaOpenArchives	Определяет число открытых пользовательских архивов.
uaOpenViews	Определяет число открытых представлений.
uaQueryArchive	Устанавливает соединение с пользовательским архивом.
uaQueryArchiveByName	Устанавливает соединение с пользовательским архивом с помощью имени архива.
uaReleaseArchive	Завершает соединение с пользовательским архивом.
uaSetLocalEvents	Устанавливает локальные события.

uaUsers	Обнаруживает число установленных соединений или активных пользователей.
---------	---

3.7.4.2 uaConnect

Описание:

Устанавливает соединение с пользовательскими архивами (среда исполнения).

```
BOOL uaConnect (  
    UAHCONNECT* phConnect )
```

Параметры:

```
UAHCONNECT* phConnect
```

Указатель на дескриптор только что подключенного пользовательского архива.

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно подключен

FALSE (Ложь): Error (Ошибка)

3.7.4.3 uaDisconnect

Описание:

Если соединение с пользовательскими архивами (среда исполнения) существует, оно будет прервано.

```
BOOL uaDisconnect (  
    UAHCONNECT hConnect )
```

Параметры:

```
UAHCONNECT hConnect
```

Дескриптор подключенного пользовательского архива (среда исполнения). Этот дескриптор генерируется функцией uaConnect.

Возвращаемое значение:

TRUE (Истина): успешное отключение пользовательского архива

FALSE (Ложь): Error (Ошибка)

3.7.4.4 uaGetLocalEvents

Описание:

Считывает локальные события
BOOL uaGetLocalEvents
(UAHCONNECT hConnect)

Параметры:

UAHCONNECT hConnect
Дескриптор пользовательского архива. Этот дескриптор генерируется функцией uaConnect.

Возвращаемое значение:

локальное событие (bLocalEvents)

3.7.4.5 ualsActive

Описание:

Проверяет, включена ли среда исполнения.
BOOL uaIsActive (
UAHCONNECT hConnect)

Параметры:

UAHCONNECT hConnect
Дескриптор пользовательского архива. Этот дескриптор генерируется функцией uaConnect.

Возвращаемое значение:

TRUE (Истина): один или несколько пользовательских архивов открыто в среде исполнения
FALSE (Ложь): в среде исполнения отсутствует пользовательский архив

3.7.4.6 uaOpenArchives

Описание:

Запросы о количестве открытых пользовательских архивов в среде исполнения.
LONG uaOpenArchives (
UAHCONNECT hConnect)

Параметры:`UAHCONNECT hConnect`

Дескриптор пользовательского архива. Этот дескриптор генерируется функцией `uaConnect`.

Возвращаемое значение:

число открытых на данный момент пользовательских архивов.

3.7.4.7 uaOpenViews**Описание:**

Запросы о количестве открытых представлений пользовательского архива в среде исполнения.

```
LONG uaOpenViews (  
UAHCONNECT hConnect )
```

Параметры:`UAHCONNECT hConnect`

Дескриптор представления. Этот дескриптор генерируется функцией `uaConnect`.

Возвращаемое значение:

число представлений, открытых на данный момент.

3.7.4.8 uaQueryArchive**Описание:**

Устанавливает соединение с пользовательским архивом для работы среды исполнения. `UaQueryArchive` создает дескриптор `UAHARCHIVE`.

```
BOOL uaQueryArchive (  
UAHCONNECT hConnect,  
LONG lArchive,  
UAHARCHIVE* phArchive )
```

Параметры:`UAHCONNECT hConnect`

Дескриптор подключенного пользовательского архива (среда исполнения). Этот дескриптор генерируется функцией `uaConnect`.

`LONG lArchive`

Индекс архива (0... `uaGetNumArchives()` -1)

UAHARCHIVE* phArchive

Указатель на дескриптор архива.

Возвращаемое значение:

TRUE (Истина): успешное создание дескриптора для пользовательского архива

FALSE (Ложь): Error (Ошибка)

Комментарий:

При использовании функций пользовательских архивов в проекте клиента, который подключен к резервированной паре серверов, соединение пользовательских архивов нельзя автоматически переключить на новый мастер при смене мастера. В этом случае все вызовы пользовательских архивов отображают сообщение об ошибке LastError UA_ERROR_SERVER_UNKNOWN = 1004. В результате в программе пользователя необходимо заново запустить uaQueryArchive() или uaQueryArchiveByName() и uaArchiveOpen().

3.7.4.9 uaQueryArchiveByName

Описание:

Устанавливает соединение с пользовательским архивом для работы в среде исполнения с помощью имени архива. UaQueryArchiveByName создает дескриптор UAHARCHIVE для пользовательского архива.

```
BOOL uaQueryArchiveByName (
    UAHCONNECT hConnect,
    LPCSTR pszName,
    UAHARCHIVE* phArchive )
```

Параметры:

UAHCONNECT hConnect

Дескриптор подключенного пользовательского архива (среда исполнения). Этот дескриптор генерируется функцией uaConnect.

LPCSTR pszName

Имя пользовательского архива. При использовании проекта клиента можно добавить префикс сервера "::" в качестве разделителя имени архива, если используется сервер, не установленный по умолчанию .

UAHARCHIVE* phArchive

Указатель на дескриптор архива.

Возвращаемое значение:

TRUE (Истина): создание дескриптора пользовательского архива выполнено успешно

FALSE (Ложь): Error (Ошибка)

Комментарий:

При использовании функций пользовательских архивов в проекте клиента, который подключен к резервированной паре серверов, соединение пользовательских архивов нельзя автоматически переключить на новый мастер при смене мастера. В этом случае все вызовы пользовательских архивов отображают сообщение об ошибке `LastError UA_ERROR_SERVER_UNKNOWN = 1004`. В результате в программе пользователя необходимо заново запустить `uaQueryArchive()` или `uaQueryArchiveByName()` и `uaArchiveOpen()`.

3.7.4.10 uaReleaseArchive

Описание:

Завершает соединение с текущим пользовательским архивом

```
BOOL uaReleaseArchive (  
    UAHARCHIVE hArchive )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

Возвращаемое значение:

TRUE (Истина): отключение пользовательского архива успешно завершено.

FALSE (Ложь): Error (Ошибка)

Comment (Комментарий)

При успешном отключении дескриптор hArchive необходимо установить на ZERO, так чтобы ошибка UA_ERROR_INVALID_HANDLE обязательно появлялась при использовании недействительного дескриптора без отключения функции в интерфейсах COM в течение долгого времени.

3.7.4.11 uaSetLocalEvents

Описание:

Устанавливает локальные события.

```
void uaSetLocalEvents (  
    UAHCONNECT hConnect  
    BOOL bLocalEvents )
```

Параметры:

UAHCONNECT hConnect

Дескриптор пользовательского архива. Этот дескриптор генерируется функцией uaConnect.

BOOL bLocalEvents

Локальное событие

3.7.4.12 uaUsers

Описание:

Функция среды исполнения uaUsers возвращает количество всех пользователей, соединенных с пользовательским архивом с помощью функции uaConnect. Обратите внимание, что здесь учитываются внутренние вызовы WinCC пользовательского архива помимо вызовов, инициированных пользователем (например, из сценариев и т. д.).

```
LONG uaUsers (
UAHCONNECT hConnect )
```

Параметры:

UAHCONNECT hConnect

Дескриптор пользовательского архива. Этот дескриптор генерируется функцией uaConnect.

Возвращаемое значение:

Число установленных связей или пользователей.

3.7.5 Функции среды исполнения для архивов

3.7.5.1 Функции среды исполнения для архивов

Функции пользовательских архивов

Эти функции используются для работы с пользовательскими архивами и представлениями в среде исполнения.

Функция	Описание
uaArchiveClose	Завершает соединение с текущим пользовательским архивом
uaArchiveDelete	Удаляет запись из текущего пользовательского архива
uaArchiveExport	Экспортирует текущий пользовательский архив
uaArchiveGetCount	Считывает число записей данных.
uaArchiveGetFieldLength	Считывает длину текущего поля
uaArchiveGetFieldName	Считывает имя текущего поля
uaArchiveGetFields	Считывает число полей
uaArchiveGetFieldType	Считывает тип текущего поля
uaArchiveGetFieldValueDate	Считывает дату и время и перемещает их в текущее поле данных
uaArchiveGetFieldValueDouble	Считывает значение Double текущего поля данных

uaArchiveGetFieldValueFloat	Считывает значение Float текущего поля данных
uaArchiveGetFieldValueLong	Считывает значение Long Int текущего поля данных
uaArchiveGetFieldValueString	Считывает значение String текущего поля данных
uaArchiveGetFilter	Считывает фильтр текущего поля данных
uaArchiveGetID	Считывает идентификатор текущего поля данных
uaArchiveGetName	Считывает имя текущего поля данных
uaArchiveGetSort	Считывает сортировку текущего поля данных
uaArchiveImport	Импортирует пользовательский архив
uaArchiveInsert	Вставляет новую запись данных в пользовательский архив
uaArchiveMoveFirst	Осуществляет переход к первой записи данных
uaArchiveMoveLast	Осуществляет переход к последней записи данных
uaArchiveMoveNext	Осуществляет переход к следующей записи данных
uaArchiveMovePrevious	Осуществляет переход к предыдущей записи данных
uaArchiveOpen	Устанавливает соединение с текущим пользовательским архивом
uaArchiveReadTagValues	Считывает значения тегов
uaArchiveReadTagValuesByName	Считывает значения тегов по имени
uaArchiveRequery	Новый запрос
uaArchiveSetFieldValueDate	Записывает текущее поле данных
uaArchiveSetFieldValueDouble	Записывает значение Double текущего поля данных
uaArchiveSetFieldValueFloat	Записывает значение Float текущего поля данных
uaArchiveSetFieldValueLong	Записывает значение Long Int текущего поля данных
uaArchiveSetFieldValueString	Записывает значение String текущего поля данных
uaArchiveSetFilter	Устанавливает фильтр
uaArchiveSetSort	Устанавливает критерий сортировки
uaArchiveUpdate	Обновляет открытый пользовательский архив
uaArchiveWriteTagValues	Заносит значения текущей записи данных в тег
uaArchiveWriteTagValuesByName	Заносит значения текущей записи данных в тег по имени

3.7.5.2 uaArchiveClose

Описание:

Завершает соединение с текущим пользовательским архивом

```
BOOL uaArchiveClose (
UAHARCHIVE hArchive )
```

Параметры:

```
UAHARCHIVE hArchive
```

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно закрыт

FALSE (Ложь): Error (Ошибка)

3.7.5.3 uaArchiveDelete**Удаление архивов**

Стандартные функции языка сценария `WinCC!AL(UAEditorScriptAllgemein,3,"", "")`

Ссылки стандартных функций для пользовательских архивов `!AL(UAEditorReferenzenScript,3,"", "")`

Описание:

Удаляет данные из пользовательского архива. Настроенный пользовательский архив, однако, сохраняется.

```
BOOL uaArchiveDelete (  
    UAHARCHIVE hArchive,  
    LPCSTR pszWhere )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

LPCSTR pszWhere

В этой строке содержится выражение выборки SQL. С его помощью определяются записи данных для удаления. Выражение идентично SQL-запросу "DELETE FROM <архив> WHERE pszWhere".

Внимание! Если эта строка пустая, весь пользовательский архив будет удален.

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно удален

FALSE (Ложь): Error (Ошибка)

3.7.5.4 uaArchiveExport

Описание:

Экспортирует текущий пользовательский архив в другой архив в формате CSV.

```
BOOL uaArchiveExport (  
    UAHARCHIVE hArchive,  
    LPCSTR pszDestination,  
    LONG lType,  
    LONG lOptions )
```

Параметр:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPCSTR pszDestination

Имя файла целевого архива. При вызове функции на клиентах указание пути относится к компьютеру сервера.

LONG lType

Формат данных целевого архива. Возможно два формата:

UA_FILETYPE_DEFAULT = 0: формат файла по умолчанию = CSV

UA_FILETYPE_CSV = 1: формат файла CSV

LONG lOptions

Options (Варианты)

Зарезервировано для последующих расширений. Должен быть 0.

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно экспортирован

FALSE (Ложь): Error (Ошибка)

3.7.5.5 uaArchiveGetCount**Описание:**

Считывает число записей данных.

```
LONG uaArchiveGetCount(  
    UAHARCHIVE hArchive,  
    LONG * plCount )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG plCount

Указатель на тег, где должны храниться записи данных.

Возвращаемое значение:

Число записей данных.

0 = Архив пуст или произошла ошибка. Необходимо создать запрос с помощью uaGetLastError().

3.7.5.6 uaArchiveGetFieldLength

Описание:

Считывает длину поля текущей записи данных.

```
LONG uaArchiveGetFieldLength(  
    UAHARCHIVE hArchive,  
    LONG lField )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где при lField = 1 выполняется адресация первого поля.

Возвращаемое значение:

Длина текущего поля.

3.7.5.7 uaArchiveGetFieldName

Описание:

Считывает имя поля в текущей записи данных.

```
VOID uaArchiveGetFieldName (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LPCSTR pszName,  
    LONG cMaxLen )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где при lField = 1 выполняется адресация первого поля.

LPCSTR pszName

Имя поля

LONG cMaxLen

Максимальная длина

3.7.5.8 uaArchiveGetFields

Описание:

Считывает число настроенных полей данных. Сюда также относятся поля ID (Идентификатор), Last User (Последний пользователь) и Last Access (Последний доступ). При вызовах в среде исполнения индексы полей нумеруются от 1 до N. Поле ID (Идентификатор) имеет индекс 0. Поля Last User (Последний пользователь) и Last Access (Последний доступ) добавляются в конец настроенных полей.

```
LONG uaArchiveGetFields (  
    UAHARCHIVE hArchive )
```

Параметры:

```
UAHARCHIVE hArchive
```

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

Возвращаемое значение:

число настроенных полей.

3.7.5.9 uaArchiveGetFieldType

Описание:

Считывает тип поля в текущей записи данных.

```
LONG uaArchiveGetFieldType (  
    UAHARCHIVE hArchive,  
    LONG lField )
```

Параметры:

```
UAHARCHIVE hArchive
```

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

```
LONG lField
```

Номер поля, где при `lField = 1` выполняется адресация первого поля.

Возвращаемое значение:

Тип текущего поля.

Символическими определениями типов полей являются следующие:

```
UA_FIELDTYPE_INTEGER
```

```
UA_FIELDTYPE_DOUBLE
```

```
UA_FIELDTYPE_STRING
```

UA_FIELDTYPE_DATETIME

3.7.5.10 uaArchiveGetFieldValueDate

Описание:

Считывает дату и время поля в текущей записи данных.

```
BOOL uaArchiveGetFieldValueDate (  
UAHARCHIVE hArchive,  
LONG lField,  
LPSYSTEMTIME pstDateTime )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где при lField = 1 выполняется адресация первого поля.

LPSYSTEMTIME pstDateTime

Указатель на тег типа SYSTEMTIME.

Возвращаемое значение:

TRUE (Истина): успешное считывание даты и времени

FALSE (Ложь): Error (Ошибка)

3.7.5.11 uaArchiveGetFieldValueDouble

Описание:

Считывает двойное значение поля в текущей записи данных.

```
BOOL uaArchiveGetFieldValueDouble (  
UAHARCHIVE hArchive,  
LONG lField,  
double* pdValue )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где при lField = 1 выполняется адресация первого поля.

double* pdValue

Указатель на тег текущего содержимого поля.

Возвращаемое значение:

TRUE (Истина): успешное считывание значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.12 uaArchiveGetFieldValueFloat**Описание:**

Считывает плавающее значение поля в текущей записи данных.

```
BOOL uaArchiveGetFieldValueFloat (
```

```
UAHARCHIVE hArchive,
```

```
LONG lField,
```

```
FLOAT* pfValue )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где при lField = 1 выполняется адресация первого поля.

FLOAT* pfValue

Указатель на плавающий тег текущего содержимого поля.

Возвращаемое значение:

TRUE (Истина): успешное считывание значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.13 uaArchiveGetFieldValueLong

Описание:

Считывает длинное целое значение поля в текущей записи данных.

```
BOOL uaArchiveGetFieldValueLong (  
UAHARCHIVE hArchive,  
LONG lField,  
LONG* pdValue )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где при lField = 1 выполняется адресация первого поля.

LONG* pdValue

Указатель на длинный тег текущего содержимого поля.

Возвращаемое значение:

TRUE (Истина): успешное считывание значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.14 uaArchiveGetFieldValueString

Описание:

Считывает строку поля в текущей записи данных.

```
BOOL uaArchiveGetFieldValueString (  
UAHARCHIVE hArchive,  
LONG lField,  
LPSTR pszString,  
LONG cMaxLen )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где при lField = 1 выполняется адресация первого поля.

LPCSTR pszString

Значение поля в качестве строки.

LONG cMaxLen

Максимальная длина строки.

Возвращаемое значение:

TRUE (Истина): успешное считывание значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.15 uaArchiveGetFilter**Описание:**

Считывает фильтр текущей записи данных. Дополнительную информацию см. в приложении "Операторы SQL".

```
VOID uaArchiveGetFilter (
    UAHARCHIVE hArchive,
    LPSTR pszFilter,
    LONG cMaxLen )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPSTR pszFilter

Считать фильтр.

LONG cMaxLen

Максимальная длина

3.7.5.16 uaArchiveGetID**Описание:**

uaArchiveGetID считывает идентификатор пользовательского архива.

```
LONG uaArchiveGetID (
    UAHARCHIVE hArchive )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

Возвращаемое значение:

Идентификатор пользовательского архива

3.7.5.17 uaArchiveGetName

Описание:

Считывает имя пользовательского архива.

```
VOID uaArchiveGetName (
    UAHARCHIVE hArchive,
    LPSTR pszName,
    LONG cMaxLen )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPSTR pszName

Указатель на буфер имени пользовательского архива.

LONG cMaxLen

Максимальная длина

Пример

```
char Filling [40];
uaArchiveGetName( hArchive, Filling, 39 );
```

3.7.5.18 uaArchiveGetSort

Описание:

uaArchiveGetSort считывает сортировку пользовательского архива. Дополнительную информацию см. в приложении "Операторы SQL".

```
VOID uaArchiveGetSort (
    UAHARCHIVE hArchive,
    LPSTR pszSort,
    LONG cMaxLen )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPCSTR pszSort

Сортировка...

LONG cMaxLen

Максимальная длина

3.7.5.19 uaArchiveImport

Описание:

uaArchiveImport импортирует пользовательский архив с помощью формата данных CSV. Структура целевого архива должна быть идентична импортируемому архиву в формате CSV.

```
BOOL uaArchiveImport (
    UAHARCHIVE hArchive,
    LPCSTR pszSource,
    LONG lType,
    LONG lOptions )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPCSTR pszSource

Имя файла исходного архива.

LONG lType

Формат данных исходного архива. Возможно два формата:

UA_FILETYPE_DEFAULT = 0: формат файла по умолчанию = CSV

UA_FILETYPE_CSV = 1: формат файла CSV

LONG lOptions

Зарезервировано для последующих расширений. Должен быть 0.

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно импортирован

FALSE (Ложь): Error (Ошибка)

3.7.5.20 uaArchiveInsert

Описание:

Вставляет локальный буфер записи данных в текущую базу данных. Чтобы данные новой записи можно было применить, поля локального буфера записи данных должны быть записаны с помощью функций uaArchiveSetFieldValue... до вызова uaArchiveInsert. Внутренний столбец ID (Идентификатор) должен быть записан в данные с помощью функции uaArchiveSetFieldValueLong.

```
BOOL uaArchiveInsert (
    UAHARCHIVE hArchive )
```

Параметры:

UARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

Возвращаемое значение:

TRUE (Истина): успешное добавление записи данных

3.7.5.21 uaArchiveMoveFirst

Описание:

Переход к первой записи данных.

```
BOOL uaArchiveMoveFirst (  
UARCHIVE hArchive )
```

Параметры:

UARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

Возвращаемое значение:

TRUE (Истина): перемещение в пользовательский архив прошло успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.22 uaArchiveMoveLast

Описание:

Переход к последней записи данных.

```
BOOL uaArchiveMoveLast (  
UARCHIVE hArchive )
```

Параметры:

UARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

Возвращаемое значение:

TRUE (Истина): перемещение в пользовательский архив прошло успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.23 uaArchiveMoveNext

Описание:

Переход к следующей записи данных.

```
BOOL uaArchiveMoveNext (  
UAHARCHIVE hArchive )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

Возвращаемое значение:

TRUE (Истина): перемещение в пользовательский архив прошло успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.24 uaArchiveMovePrevious

Описание:

Переход к предыдущей записи данных.

```
BOOL uaArchiveMovePrevious (  
UAHARCHIVE hArchive )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

Возвращаемое значение:

TRUE (Истина): перемещение в пользовательский архив прошло успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.25 uaArchiveOpen

Описание:

uaArchiveOpen открывает существующий пользовательский архив. uaArchiveOpen всегда необходимо вызывать при выполнении операций считывания или записи с этим пользовательским архивом. uaArchiveOpen необходимо вызывать, например, до вызова функций uaArchiveMoveFirst, uaArchiveMoveLast, uaArchiveMoveNext, uaArchiveMovePrevious, uaArchiveDelete, uaArchiveUpdate, uaArchiveInsert.

Примечание

Сортировка и фильтрация пользовательских архивов

Функции `uaArchiveSetSort` и `uaArchiveSetFilter` для пользовательского архива можно использовать, не открывая пользовательский архив с помощью `uaArchiveOpen`.

```
BOOL uaArchiveOpen (  
UAHARCHIVE hArchive )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно открыт

FALSE (Ложь): Error (Ошибка)

3.7.5.26 `uaArchiveReadTagValues`

Описание:

Считывает текущее значение из поля тега.

```
BOOL uaArchiveReadTagValues (  
UAHARCHIVE hArchive,  
LONG* pnFields,  
LONG cFields,  
LONG lOptions )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

LONG* pnFields

Зарезервировано для дальнейшего применения (NULL).

LONG cFields

Число переданных индексов поля (размер массива pnFields).

Зарезервировано для дальнейшего применения (0).

LONG lOptions

Зарезервировано для дальнейшего применения (0).

Для всех других значений lOptions данные добавляются в место, где находится указатель.

Возвращаемое значение:

TRUE (Истина): считывание в пользовательском архиве прошло успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.27 uaArchiveReadTagValuesByName**Описание:**

Считывает значения тегов в текущих данных.

```
BOOL uaArchiveReadTagValuesByName (
    UAHARCHIVE hArchive,
    LPCSTR pszFields,
    LONG lOptions )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPCSTR pszFields

Зарезервировано для дальнейшего применения (NULL).

LONG lOptions

Зарезервировано для дальнейшего применения (0).

Возвращаемое значение:

TRUE (Истина): считывание в пользовательском архиве прошло успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.28 uaArchiveRequery**Описание:**

После вызова функции uaArchiveSetFilter и uaArchiveSetSort необходимо перезагрузить архив с помощью функции uaArchiveRequery.

Примечание**Сортировка и фильтрация пользовательских архивов**

Функции uaArchiveSetSort и uaArchiveSetFilter для пользовательского архива можно использовать, не открывая пользовательский архив с помощью uaArchiveOpen. В этом случае не вызывайте функцию uaArchiveRequery.

Функция `uaArchiveRequery` вызывается в следующих случаях.

- При создании записей с помощью объекта таблиц пользовательского архива.
- При создании записей в редакторе User Archives (Пользовательские архивы), которые необходимо передать в поля таблицы.

```
BOOL uaArchiveRequery(  
UANARCHIVE hArchive )
```

Параметры:

`UANARCHIVE hArchive`

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

Возвращаемое значение:

TRUE (Истина): повторный запрос выполнен успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.29 `uaArchiveSetFieldValueDate`

Описание:

Заносит дату и время в поле текущей записи данных.

```
BOOL uaArchiveSetFieldValueDate (  
UANARCHIVE hArchive,  
LONG lField,  
LPSYSTEMTIME pstDateTime )
```

Параметры:

`UANARCHIVE hArchive`

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

`LONG lField`

Номер поля, где `lField = 1` адресуется первому настроенному полю. При `lField = 0` выполняется адресация к полю ID (Идентификатор).

`LPSYSTEMTIME pstDateTime`

Дата и время

Возвращаемое значение:

TRUE (Истина): успешная запись даты и времени

FALSE (Ложь): Error (Ошибка)

3.7.5.30 uaArchiveSetFieldValueDouble

Описание:

Заносит значение Double в поле текущей записи данных.

```
BOOL uaArchiveSetFieldValueDouble (  
UAHARCHIVE hArchive,  
LONG lField,  
double dValue )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где lField = 1 адресуется первому настроенному полю. При lField = 0 выполняется адресация к полю ID (Идентификатор).

double dValue

Значение поля

Возвращаемое значение:

TRUE (Истина): успешная запись значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.31 uaArchiveSetFieldValueFloat

Описание:

Заносит значение Float в поле текущей записи данных.

```
BOOL uaArchiveSetFieldValueFloat (  
UAHARCHIVE hArchive,  
LONG lField,  
float fValue )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где lField = 1 адресуется первому настроенному полю. При lField = 0 выполняется адресация к полю ID (Идентификатор).

float fValue

Значение поля

Возвращаемое значение:

TRUE (Истина): успешная запись значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.32 uaArchiveSetFieldValueLong

Описание:

Заносит значение Long Integer в поле текущей записи данных.

```
BOOL uaArchiveSetFieldValueLong (  
UAHARCHIVE hArchive,  
LONG lField,  
LONG dValue )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где lField = 1 адресуется первому настроенному полю. При lField = 0 выполняется адресация к полю ID (Идентификатор).

LONG dValue

Значение поля

Возвращаемое значение:

TRUE (Истина): успешная запись значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.33 uaArchiveSetFieldValueString

Описание:

Заносит String в поле текущей записи данных.

```
BOOL uaArchiveSetFieldValueString (  
UAHARCHIVE hArchive,  
LONG lField,  
LPCSTR pszString )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG lField

Номер поля, где lField = 1 адресуется первому настроенному полю. При lField = 0 выполняется адресация к полю ID (Идентификатор).

LPCSTR pszString

Значение поля

Возвращаемое значение:

TRUE (Истина): успешная запись значения поля

FALSE (Ложь): Error (Ошибка)

3.7.5.34 uaArchiveSetFilter**Описание:**

Устанавливает фильтр. Эту функцию можно вызывать, не открывая архив с помощью uaArchiveOpen.

Примечание

При открытии пользовательского архива с помощью uaArchiveOpen необходимо перезагрузить пользовательский архив после фильтрации с помощью функции uaArchiveRequery. Дополнительную информацию см. в приложении "Операторы SQL".

```
VOID uaArchiveSetFilter (  
    UAHARCHIVE hArchive,  
    LPSTR pszFilter )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPSTR pszFilter

Необходимо установить фильтр.

3.7.5.35 uaArchiveSetSort**Описание:**

Устанавливает сортировку пользовательского архива. Эту функцию можно вызывать, не открывая архив с помощью uaArchiveOpen.

Примечание

При открытии пользовательского архива с помощью `uaArchiveOpen` необходимо перезагрузить пользовательский архив после сортировки с помощью функции `uaArchiveRequery`. Дополнительную информацию см. в приложении "Операторы SQL".

```
BOOL uaArchiveSetSort (
    UAHARCHIVE hArchive,
    LPSTR pszSort )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

LPCSTR pszSort

Сортировка...

Возвращаемое значение:

TRUE (Истина): установка сортировки выполнена успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.36 uaArchiveUpdate

Описание:

Обновляет открытый пользовательский архив. Все изменения данных пользовательского архива включаются в базу данных. Конфигурация пользовательского архива остается неизменной.

```
BOOL uaArchiveUpdate (
    UAHARCHIVE hArchive )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется `uaQueryArchive` или `uaQueryArchiveByName`.

Возвращаемое значение:

TRUE (Истина): пользовательский архив был успешно обновлен

FALSE (Ложь): ошибка Update_failed = 106

Данная ошибка возникает при нарушении целостности, например:

Флажок Field must contain a Value (Поле должно содержать значение) установлен, но значение не было занесено в это поле.

3.7.5.37 uaArchiveWriteTagValues

Описание:

Заносит значения текущей записи данных в теги.

```
BOOL uaArchiveWriteTagValues (  
    UAHARCHIVE hArchive,  
    LONG* pnFields,  
    LONG cFields,  
    LONG lOptions )
```

Параметры:

UAHARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LONG* pnFields

Зарезервировано для дальнейшего применения (NULL).

LONG cFields

Зарезервировано для дальнейшего применения (0).

LONG lOptions

Зарезервировано для дальнейшего применения (0).

Возвращаемое значение:

TRUE (Истина): считывание в пользовательском архиве прошло успешно

FALSE (Ложь): Error (Ошибка)

3.7.5.38 uaArchiveWriteTagValuesByName

Описание:

Заносит значения текущей записи данных в теги. Доступ основан на именах пользовательского архива и поля.

```
BOOL uaArchiveWriteTagValuesByName (  
    UAHARCHIVE hArchive,  
    LPCSTR pszFields,  
    LONG lOptions )
```

Параметры:

UARCHIVE hArchive

Дескриптор пользовательского архива. Этот дескриптор генерируется uaQueryArchive или uaQueryArchiveByName.

LPCSTR pszFields

Зарезервировано для дальнейшего применения (NULL).

LONG lOptions

Зарезервировано для дальнейшего применения (0).

Возвращаемое значение:

TRUE (Истина): считывание в пользовательском архиве прошло успешно

FALSE (Ложь): Error (Ошибка)

Индекс

A

Access rights, 33, 39
Appendix, 74
Archive fields
 configuring, 42
 Configuring, 30, 36, 38, 39
 Create, 36
Archive fields:Configuring, 30, 36, 38, 39, 42
Archive fields:Creating, 36

C

CC_StartHTMLHelp, 7
Changing language, 64
Communication, 31, 67
 via Raw data tags, 69, 70, 71
Communication:via raw data tags, 68, 71, 72
Communication:via WinCC tags, 68
Configuring: from Archive Fields, 30, 36, 38, 39, 42
Configuration:of a User Archive, 28
configuring
 from archive fields, 42
 of user archives, 28, 40, 63
 of views, 57, 61
Configuring
 from archive fields, 30, 36, 38, 39
 of a User Archive, 28
 of user archives, 31, 32, 33, 37
 of views, 53, 55, 56, 58, 59, 60, 62
Configuring:of User Archives, 28, 32, 33, 37, 40, 63
Configuring:views, 53, 55, 56, 57, 58, 59, 60, 61, 62
Control tags, 32
 Example:, 47
 Features, 45
Control Tags:Properties, 45
Create actions, 80

D

Data Window, 9
Description of the User Archives Functions, 93

H

Handles of the Standard Functions (User Archives), 80, 82

I

Introduction to User Archives, 7

M

Menu commands, 15
Modification Configuration of User Archives, 52

N

Navigation window, 9

P

Performance to read and write tags, 77
Properties
 of a User Archive, 37
Properties:of a User Archive, 37

Q

Quantity framework, 76

R

Raw data tags, 68, 69, 70, 71, 72
Recipe, 11, 12
Redundancy and User Archive, 63
Relations, 59

S

Scripts, 85
 Editing User Archives, 80

- SIMATIC S5, 67, 68, 73
- SIMATIC S7, 67, 68, 73
- SQL, 59, 74
- Standard functions, 11
- Standard functions (User Archives), 11, 79
 - Editing User Archives, 80
 - for configuring user archives, 96
 - Handles, 80, 82
 - Runtime functions, 108, 114
- Standard functions (User Archives):Example of use, 85
- Structure for Programming the "uaAddArchive" Handle, 106

- T**
- Table window, 9, 27

- U**
- uaAddArchive, 97
- uaAddField, 97
- uaArchiveClose, 115
- uaArchiveDelete, 116
- uaArchiveExport, 116
- uaArchiveGetCount, 117
- uaArchiveGetFieldLength, 118
- uaArchiveGetFieldName, 118
- uaArchiveGetFields, 119
- uaArchiveGetFieldType, 119
- uaArchiveGetFieldValueDate, 120
- uaArchiveGetFieldValueDouble, 120
- uaArchiveGetFieldValueFloat, 121
- uaArchiveGetFieldValueLong, 122
- uaArchiveGetFieldValueString, 122
- uaArchiveGetFilter, 123
- uaArchiveGetID, 123
- uaArchiveGetName, 124
- uaArchiveGetSort, 124
- uaArchiveImport, 125
- uaArchiveInsert, 125
- uaArchiveMoveFirst, 126
- uaArchiveMoveLast, 126
- uaArchiveMoveNext, 127
- uaArchiveMovePrevious, 127
- uaArchiveOpen, 127
- uaArchiveReadTagValues, 128
- uaArchiveReadTagValuesByName, 129
- uaArchiveRequery, 129
- uaArchiveSetFieldValueDate, 130
- uaArchiveSetFieldValueDouble, 131
- uaArchiveSetFieldValueFloat, 131
- uaArchiveSetFieldValueLong, 132
- uaArchiveSetFieldValueString, 132
- uaArchiveSetFilter, 133
- uaArchiveSetSort, 133
- uaArchiveUpdate, 134
- uaArchiveWriteTagValues, 135
- uaArchiveWriteTagValuesByName, 135
- uaConfigArchive "Structure for Programming the Handles", 107
- uaConnect, 109
- uaDisconnect, 109
- uaGetArchive, 98
- uaGetField, 98
- uaGetLastError, 93
- uaGetLastHResult, 96
- uaGetLocalEvents, 109
- uaGetNumArchives, 99
- uaGetNumFields, 99
- ualsActive, 110
- uaOpenArchives, 110
- uaOpenViews, 111
- uaQueryArchive, 111
- uaQueryArchiveByName, 112
- UaQueryConfiguration, 100
- uaReleaseArchive, 113
- uaReleaseConfiguration, 100
- uaRemoveAllArchives, 101
- uaRemoveAllFields, 102
- uaRemoveArchive, 102
- uaRemoveField, 103
- uaSetArchive, 103
- uaSetField, 104
- uaSetLocalEvents, 113
- uaUsers, 114
- User archive:Create actions, 80
- User archives, 7
 - Access rights, 33
 - Application Options, 11
 - Archive fields, 36
 - Change Configuration, 52
 - Communication, 31, 67
 - configuring, 28, 40, 63
 - Configuring, 28, 31, 32, 33, 37
 - Control tags, 32
 - Create, 28, 31, 32, 33, 37
 - Editing With Scripts, 80
 - Editor, 9
 - Performance Features, 13
 - Properties, 37
 - Quantity framework, 13
 - Standard functions, 79
 - Views, 53
- User Archives Table Element, 10
- User archives: Using WinCC Redundancy, 63

User archives:Access rights, 33
User Archives:Application areas, 11
User archives:Archive fields, 36
User Archives:Configuring, 28, 32, 33, 37, 40
User Archives:Configuring, 63
User archives:control tags, 32
User archives:Creating, 28, 32, 33, 37
User Archives:Editor, 9
User Archives:Performance markers, 13
User archives:Properties, 37
User archives:Views, 53
User Archives:Volumes, 13

V

Views, 11, 12
 configuring, 57, 61
 Configuring, 53, 55, 56, 58, 59, 60, 62
 Create, 53
 Relations, 59
Views:Configuring, 53, 55, 56, 57, 58, 59, 60, 61, 62
Views:Creating, 53
Views:Relations, 59

W

WinCC Tags, 68
WinCC User Archives Element, 10
WinCC User Archives Table Element, 10

