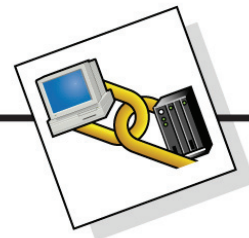




RSLinx Classic

For Rockwell Automation
Networks and Devices



Руководство по применению: Сбор данных ControlLogix в RSLinx Classic Переработано для RSLinx Classic 2.50.00 (CPR 7)

Rockwell Software
Версия документа: 2.50.00 (CPR 7)

ВВЕДЕНИЕ	4
СОВЕТЫ ПО ПРОЕКТИРОВАНИЮ ДЛЯ ОПТИМАЛЬНОЙ ПРОИЗВОДИТЕЛЬНОСТИ	4
<i>ОБЕСПЕЧЕНИЕ ДОСТАТОЧНОЙ ПОЛОСЫ ПРОПУСКАНИЯ СВЯЗИ ДЛЯ ПРОЦЕССОРА CONTROLLOGIX 4</i>	
<i>Непрерывная задача</i>	<i>4</i>
<i>Периодические задачи</i>	<i>5</i>
<i>Использование непрерывных и периодических задач</i>	<i>6</i>
<i>Конфигурирование процессоров ControlLogix в качестве концентраторов данных</i>	<i>7</i>
<i>МАКСИМАЛЬНОЕ ИСПОЛЬЗОВАНИЕ ПОЛОСЫ ПРОПУСКАНИЯ СВЯЗИ CONTROLLOGIX</i>	<i>7</i>
<i>Использование только одного RSLinx Classic для связи с процессором ControlLogix</i>	<i>7</i>
<i>Использование RSLinx Classic Gateway для передачи данных на несколько станций</i>	<i>7</i>
<i>Сбор данных из массивов или структур UDT</i>	<i>8</i>
<i>Отделение сбора данных от критических во времени управляющих программ</i>	<i>8</i>
<i>Не запрашивайте данные слишком часто</i>	<i>8</i>
<i>СОХРАНЕНИЕ ТЕГОВ НА СКАНЕ</i>	<i>8</i>
<i>ИСПОЛЬЗОВАНИЕ БОЛЕЕ БЫСТРОГО ПРОЦЕССОРА CONTROLLOGIX</i>	<i>9</i>
<i>УБЕДИТЕСЬ, ЧТО ВЕРСИИ RSLINX CLASSIC И CONTROLLOGIX НЕ УСТАРЕЛИ</i>	<i>9</i>
КОНФИГУРИРОВАНИЕ RSLINX CLASSIC	9
<i>ПАРАМЕТРЫ РАЗДЕЛА</i>	<i>11</i>
<i>Незапланированные (Unsolicited) сообщения</i>	<i>11</i>
<i>Тайм-аут коммуникации</i>	<i>11</i>
<i>Использование символов</i>	<i>12</i>
<i>RSLogix 5000 АВТОМАТИЧЕСКИ КОНФИГУРИРУЕТ РАЗДЕЛЫ В RSLINX CLASSIC</i>	<i>12</i>
ПОНИМАНИЕ МЕХАНИЗМА СБОРА ДАННЫХ CONTROLLOGIX	12
<i>ШАГ ЗА ШАГОМ</i>	<i>12</i>
<i>ОПТИМИЗАЦИЯ ЧТЕНИЯ</i>	<i>14</i>
<i>Оптимизация массива</i>	<i>14</i>
<i>Оптимизация Булева массива (Boolean Array)</i>	<i>15</i>
<i>Оптимизация UDT</i>	<i>15</i>
<i>Оптимизация первого уровня или Оптимизация Разбросанных данных</i>	<i>17</i>
<i>Сколько данных можно прочесть в одном оптимизированном пакете ControlLogix (ControlLogix Optimized Packet)?</i>	<i>17</i>
<i>ОПТИМИЗАЦИЯ ЗАПИСИ</i>	<i>17</i>
<i>Как использовать оптимизацию записи</i>	<i>17</i>
<i>Оптимизация массива при записи</i>	<i>18</i>
<i>Сколько тегов можно отправить в ControlLogix в одном пакете записи?</i>	<i>18</i>
<i>Тип данных</i>	<i>18</i>
<i>Максимальное число элементов, которые можно уместить в пакет записи</i>	<i>18</i>
<i>АДАПТИВНЫЙ ОПРОС (SMART POLLING) RSLINX CLASSIC</i>	<i>18</i>
<i>КАК RSLINX CLASSIC ИСПОЛЬЗУЕТ ПОДКЛЮЧЕНИЯ (CONNECTIONS) CONTROLLOGIX</i>	<i>18</i>
<i>ИСПОЛЬЗОВАНИЕ ПАМЯТИ В ПРОЦЕССОРЕ CONTROLLOGIX</i>	<i>20</i>
ЖУРНАЛ РЕГИСТРАЦИИ СОБЫТИЙ СВЯЗИ (COMMUNICATIONS EVENT LOG)	21
ПОЛУЧЕНИЕ ДАННЫХ RSLINX CLASSIC В ПРИЛОЖЕНИЯХ	22
<i>УТИЛИТА ИМПОРТА ТЕГОВ LOGIX5000 (LOGIX5000 TAG IMPORT UTILITY) – ИМПОРТ ТЕГОВ CONTROLLOGIX В RSVIEW И RSSQL</i>	<i>22</i>
<i>ПРОСМОТР И ВЫБОР ТЕГОВ CONTROLLOGIX</i>	<i>22</i>
НЕКОТОРЫЕ ОСОБЕННОСТИ RSLINX CLASSIC И СОВЕТЫ	25
<i>СОХРАНЕНИЕ/ВОССТАНОВЛЕНИЕ КОНФИГУРАЦИЙ RSLINX CLASSIC В/ИЗ ФАЙЛА</i>	<i>25</i>

<i>Мониторинг данных CONTROLLOGIX</i>	25
<i>Функция RSLINX CLASSIC "COPY DDE/OPC LINK" поддерживает просмотр тегов CONTROLLOGIX</i>	26
<i>Получение информации о модулях в RSLINX CLASSIC</i>	28
ЗАМЕЧАНИЯ ПО ПРИМЕНЕНИЮ	29
<i>Одновременное помещение тегов на скан</i>	29
<i>Изменение тегов CONTROLLOGIX, находящихся на скане</i>	29
<i>Резервные сервера RSLINX CLASSIC GATEWAY и команда NODESWITCH RSVIEW</i>	30
<i>Резервирование процессоров CONTROLLOGIX</i>	30
<i>Использование переменных типа STRING в CONTROLLOGIX</i>	30
ПОИСК НЕИСПРАВНОСТЕЙ	32
ПРОИЗВОДИТЕЛЬНОСТЬ	32
<i>Пропускная способность в зависимости от системных издержек (SYSTEM OVERHEAD TIME SLICE) для непрерывной задачи</i>	32
<i>Пропускная способность в зависимости от интервала перезапуска (PERIODIC RATE) периодической задачи</i>	33
<i>Пропускная способность в зависимости от системных издержек и интервала перезапуска для непрерывной и периодической задачи</i>	35
<i>Производительность в зависимости от типа процессора</i>	36

Введение

Этот документ содержит советы по проектированию, поиску неисправностей, замечания по применению и данные по производительности для осуществления сбора данных в RSLinx Classic с помощью процессоров ControlLogix.

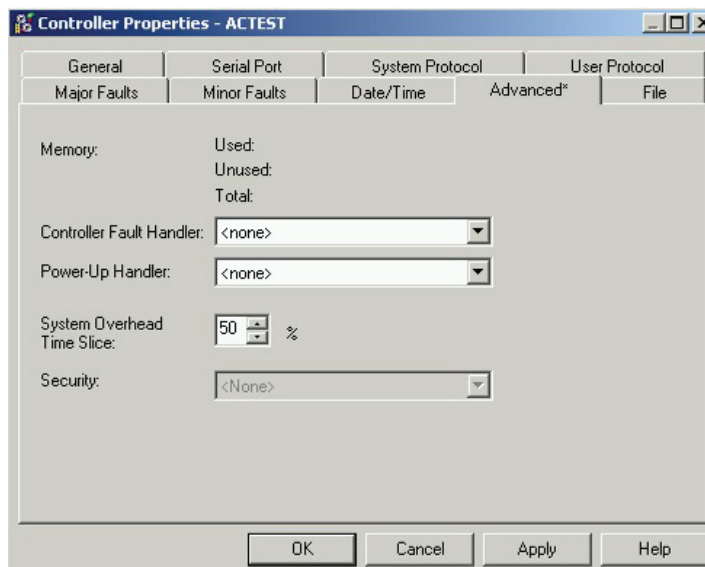
Советы по проектированию для оптимальной производительности

Обеспечение достаточной полосы пропускания связи для процессора ControlLogix

Полоса пропускания связи, доступная для сбора данных, определяется непосредственно через конфигурацию задач, выполняемых процессором ControlLogix. Необходимо делать выбор между сокращением времени на выполнение задачи и производительностью сбора данных. По умолчанию настройка процессора ControlLogix оптимизирована для быстрого выполнения задач. Вам необходимо будет найти соответствующий компромисс между временем выполнения задачи и производительностью сбора данных для вашего приложения. Процессор ControlLogix может выполнять два типа задач: непрерывные и периодические. Ниже описывается каждый из этих типов, а также то, как их конфигурации влияют на полосу пропускания связи.

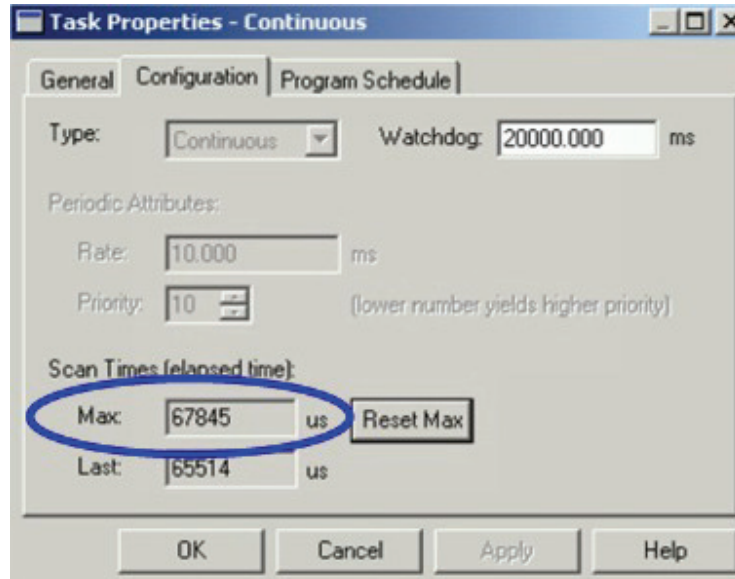
Непрерывная задача

Это тип задачи, заданный по умолчанию. Может существовать только одна непрерывная задача. Центральный процессор ControlLogix постоянно переключается между выполнением задачи и обслуживанием связи. Параметр *System Overhead Time Slice (Системные издержки)* задает, сколько времени отводится на обслуживание связи, а сколько – на выполнение задачи. Рассмотрим пример, где параметр *System Overhead Time Slice* был установлен на 10 %. В этом случае можно ожидать, что из каждых 10мс 9мс будут потрачены на выполнение задачи и 1мс - для обслуживания связи, включая сбор данных RSLinx Classic, движение, ввод/вывод и передачу сообщений. При увеличении этого значения в параметре *System Overhead Time Slice*, как правило, увеличивается время сканирования задачи. *System Overhead Time Slice* – это свойство процессора ControlLogix. Этот параметр можно задать в RSLogix в меню *Edit (Правка) -> Controller Properties (Свойства контроллера)*. Его также можно изменить с помощью тега OPC вместе с командой ControlLogix Set (Релейная логика).



В закладке "Advanced" (Дополнительно) диалогового окна Controller Properties (Свойства контроллера) программы RSLogix5000 находится параметр "System Overhead Time Slice" (Системные издержки).

Сбор данных ControlLogix в RSLinx Classic

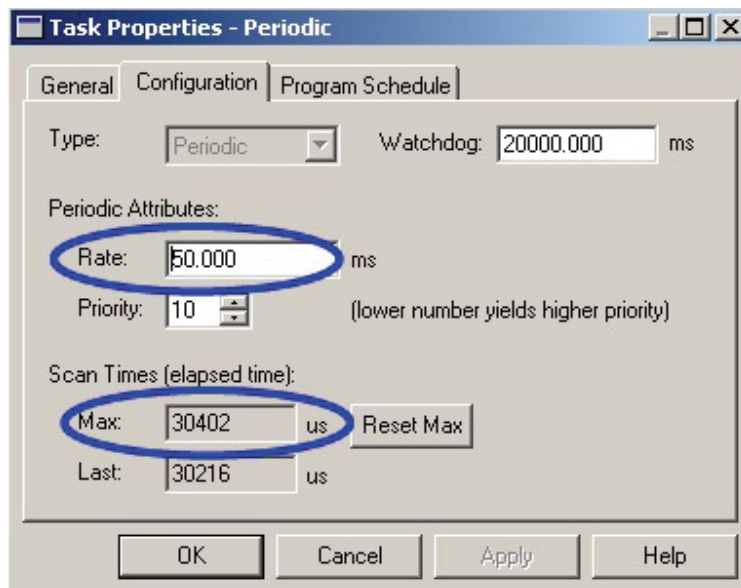


В закладке “Configuration” (Конфигурация) диалогового окна *Task Properties* (Свойства задачи) программы *RSLogix5000* для непрерывной задачи находится параметр *Max Scan Time* (Максимальное время сканирования задачи)

Периодические задачи

Выполнение периодической задачи происходит без прерывания для установления связи. Связь обслуживается только в промежутках между выполнением задач. Полосу пропускания связи можно увеличить за счет увеличения параметра *Periodic Rate* (Интервал Перезапуска) периодической задачи так, чтобы она запускалась реже. И наоборот, чем плотнее установлен график запуска периодической задачи, тем меньше полоса пропускания связи процессора *ControlLogix*. Например, если для запуска периодической задачи требуется 30мс, а по графику запуск должен осуществляться каждые 50мс, то тогда связь будет заблокирована на 30мс и затем она будет обслуживаться только в следующие 20мс. Движение, ввод/вывод и передача сообщений также должны обслуживаться в интервалах между выполнением периодических задач. Обратите внимание, что параметр *System Overhead Time Slice* не влияет на полосу пропускания связи в отношении периодической задачи.

И для большего числа периодических задач связь в любом случае обслуживается только в интервалах между выполнением периодических задач. Увеличение числа периодических задач делает планирование более сложным и уменьшает полосу пропускания связи.



В закладке “Task Properties (Свойства задачи) ->Configuration (Конфигурация)” программы RSLogix5000 для периодической задачи находятся параметры Periodic Rate (Интервал Перезапуска) и Scan Time (Время сканирования) задачи.

Использование непрерывных и периодических задач

Если непрерывные задачи и периодическая задача выполняются одновременно, полоса пропускания связи будет ограничена. Связь будет обслуживаться только в то время, когда выполняется непрерывная задача в пропорции, определенной параметром System Overhead Time Slice. Непрерывная задача может запускаться только в интервалах между выполнением периодических задач.

При наличии непрерывных и периодических задач параметры *System Overhead Time Slice* и *Periodic Rate* должны быть увеличены, чтобы полоса пропускания связи была больше. Если увеличивается только один из этих параметров, то время сканирования задачи может значительно увеличиться при незначительном увеличении полосы пропускания связи. В первую очередь необходимо устранить любые ненужные задачи путем объединения вашего кода PLC в меньшее число задач. Затем для периодической задачи максимально увеличьте параметр *Periodic Rate* таким образом, чтобы задача выполнялась только по мере необходимости. Наконец, увеличьте параметр *System Overhead Time Slice*.

Конфигурирование процессоров ControlLogix в качестве концентраторов данных

Если процессор ControlLogix предназначен в первую очередь для работы в качестве концентратора данных, то необходимо использовать только одну задачу, чтобы максимизировать полосу пропускания связи. При наличии только одной задачи в процессоре ControlLogix полоса пропускания максимально увеличивается.

Максимальное использование полосы пропускания связи ControlLogix

Использование только одного RSLinx Classic для связи с процессором ControlLogix

Мы настоятельно рекомендуем использовать только один RSLinx Classic, устанавливающий связь с любым процессором ControlLogix, или два, если требуется резервирование серверов данных. RSLinx Classic разработан для оптимизации связи, поэтому использование нескольких RSLinx Classic на один контроллер будет неэффективным, а также увеличит затраты ресурсов процессора и памяти ControlLogix, необходимых для установления связи. Эти ресурсы часто становятся критическим звеном при осуществлении сбора данных.

Вполне нормально, если станция RSLinx Classic устанавливает связь с несколькими процессорами ControlLogix. Нет никакого физического ограничения на то, с каким количеством процессоров ControlLogix может связываться один RSLinx, но есть практический предел, определяемый ресурсами центрального процессора самой рабочей станции.

Использование RSLinx Classic Gateway для передачи данных на несколько станций

Во многих приложениях вам, возможно, понадобится передавать данные ControlLogix на несколько различных человеко-машинных интерфейсов (HMI), пакетных обработок, или других пунктов сбора данных. Необходимо увеличить использование RSLinx Classic Gateway, чтобы осуществлять сбор данных на нескольких станциях без необходимости для каждой станции устанавливать связь с контроллером ControlLogix. Каждая станция будет соединяться с RSLinx Classic Gateway через Remote OPC. RSLinx Classic Gateway оптимизирует несколько запросов станций сбора данных для ControlLogix, за счет чего сокращаются затраты на осуществление связи. Например, если несколько станций запрашивают один и тот же тег, то RSLinx Classic Gateway запросит его только однажды у ControlLogix и передаст его каждой из станций сбора данных. Никакого физического ограничения на то, сколько Remote OPC-клиентов может соединяться с RSLinx Classic Gateway, не существует.

Правильное усиление мощности RSLinx Classic Gateway

Это очень важно. RSLinx Classic Gateway с удаленным RSView или другими удаленными приложениями сбора данных используется не так, как с RSLogix. Для сбора данных используется Remote OPC (Удаленный OPC). Это можно осуществить за счет конфигурирования разделов DDE/OPC RSLinx Classic Gateway и подключения станции сбора данных с помощью Remote OPC. Обратите внимание, что RSLinx Classic не будет использоваться даже на станциях сбора данных.

Напротив, RSLogix должен иметь местный RSLinx Classic с драйвером TCP, который ссылается на драйвер RSLinx Classic Gateway. Не используйте эту установку для сбора данных, поскольку она пропускает этап оптимизации для RSLinx Classic и приводит к большим затратам на установление связи в ControlLogix.

Устранение точки отказа с помощью резервного RSLinx Classic Gateway

Это легко осуществить, если у вас имеется резервный сервер RSLinx Classic Gateway. При отказе RSLinx Classic Gateway все станции HMI могут передать функцию сбора данных на резервный сервер RSLinx Classic Gateway. Например, RSView имеет простую команду переключения узла, с помощью которой можно переключить сбор данных от одного сервера RSLinx Classic Gateway на другой. Смотрите для примера раздел данного документа под названием Замечания по применению (Application Note).

Совет: Затраты на аппаратное обеспечение можно также снизить, если сделать одну из станций HMI сервером или резервным сервером RSLinx Classic Gateway.

Сбор данных из массивов или структур UDT

Для сбора данных, содержащихся в массивах или структурах UDT (User Defined (Data)Types – Типы (данных), определяемые пользователем), которые можно оптимизировать, будет использоваться меньше ресурсов и памяти центрального процессора ControlLogix. Оптимизация массива происходит автоматически, как только приложение сбора данных одновременно помещает элементы одномерного массива на скан. Внутри процессор ControlLogix воспримет набор этих элементов массива как один тег, что позволит использовать меньше ресурсов и памяти центрального процессора.

Оптимизация UDT осуществляется за счет того, что RSLinx Classic захватывает всю структуру UDT, если любой из элементов структуры помещается на скан, и в RSLinx Classic разрешена оптимизация UDT. Для получения дополнительной информации смотрите далее в этом документе раздел [Оптимизация UDT](#).

Отделение сбора данных от критических во времени управляющих программ

Зачастую процессор ControlLogix выполняет критическую во времени программу, которой не хватает времени для установления связи. Если требуется увеличение полосы пропускания связи, и нет возможности увеличить время выполнения задачи, то попробуйте максимально объединить сбор данных и ввод/вывод в тех процессорах ControlLogix, которые не выполняют критические во времени задачи.

Не запрашивайте данные слишком часто

Иногда вы можете добиться более высокой производительности, если не будете запрашивать данные слишком часто. Например, если вы осуществляете сбор данных каждые 10мс в приложении HMI, и обновления происходят слишком медленно, то попробуйте увеличить интервал между сбором данных до 250-500мс и более. В любом случае экраны HMI не могут обновляться каждые 10мс, и вы сможете уменьшить нагрузку на процессор ControlLogix и вашу станцию сбора данных.

Сохранение тегов на скане

Когда приложение сбора данных помещает и снимает теги со скана, RSLinx Classic должен взаимодействовать с процессором ControlLogix и создавать оптимизированные пакеты ControlLogix (ControlLogix Optimized Packets). Этим и объясняется, почему иногда для получения данных при переключении между экранами HMI или при запуске проекта HMI необходимо время. Для многих приложений при запуске проекта лучше поместить все теги на скан. Это позволит добиться лучшей оптимизации при небольшой задержке во время переключения между экранами. Если на скане будут находиться все теги, то понадобится больше памяти процессора ControlLogix, но при этом связь будет более стабильной. Вам понадобится больше сетевого трафика, однако во многих случаях это не является критическим параметром. Если же ваш человеко-машинный интерфейс (HMI) постоянно помещает и снимает теги со скана, то связь зачастую может прерываться, и вы не сможете добиться лучшей оптимизации использования RSLinx Classic и процессора ControlLogix. Возможно, вам придется попробовать несколько

вариантов конкретных установок, чтобы определить, какие из них лучше всего подходят для вашего приложения.

Использование более быстрого процессора ControlLogix

Производительность сбора данных может увеличиться на 150 % при использовании ControlLogix 5563 вместо 5555. Кроме того, 5563 выполняет задачи приблизительно вдвое быстрее, чем 5555 или 5550. Это позволяет также увеличить полосу пропускания связи. Производительность сбора данных увеличивается на 30%-40% при использовании ControlLogix 5555 вместо ControlLogix 5550.

Убедитесь, что версии RSLinx Classic и ControlLogix не устарели

RSLinx Classic и контроллеры на базе ControlLogix непрерывно совершенствуются, предлагая улучшенные характеристики и производительность сбора данных. Обратитесь к таблице ниже, чтобы убедиться, что вы используете последние версии продуктов.

Модернизация	Минимальная допустимая версия RSLinx Classic	Минимальная допустимая версия ControlLogix Firmware
First Level optimization	RSLinx 2.1*	4.41*
Array Optimization	RSLinx 2.20	4.41
UDT Optimization	RSLinx 2.20 SP1	4.41
String Support	RSLinx 2.30	8.x
Write Optimization (Значительно расширена загрузка пакетов и рецептов)	RSLinx 2.30.02	10.x
Различные усовершенствования и доработки	RSLinx 2.40 and +	---

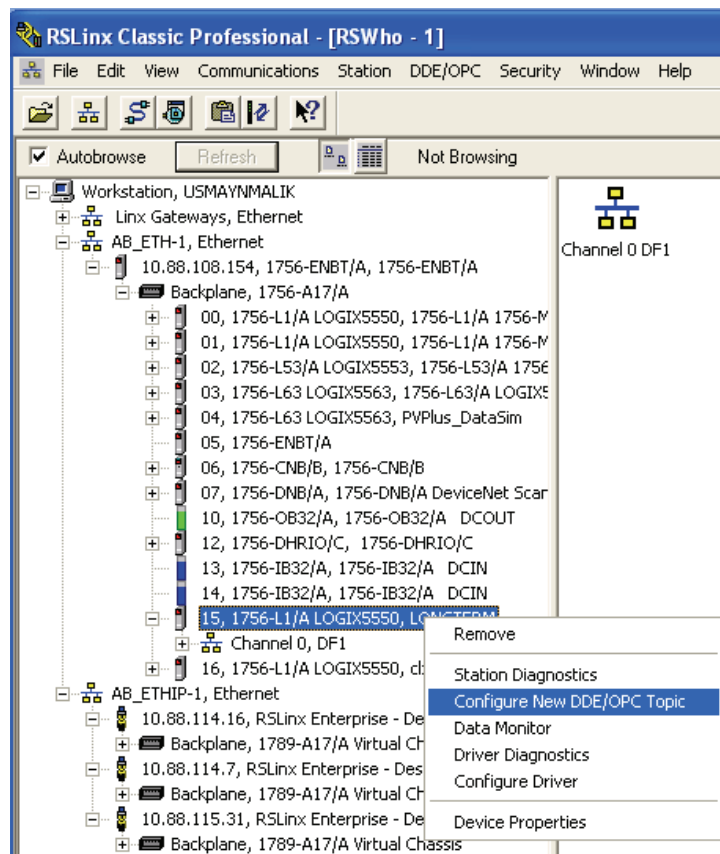
* Вы должны иметь как минимум версии RSLinx 2.1 и ControlLogix Firmware 4.41 для осуществления любого типа сбора данных.

Конфигурирование RSLinx Classic

Конфигурирование RSLinx Classic для связи с приложениями сбора данных подразумевает создание Раздела (Topic). Настройки RSLinx Classic будут одинаковыми как для DDE, так и для OPC. Для получения наивысшей производительности мы рекомендуем использовать в любом процессоре ControlLogix не более одного RSLinx Classic для сбора данных. Связь будет менее надежной, если вы будете создавать разделы на нескольких станциях с RSLinx Classic для одного и того же процессора ControlLogix. Если несколько станций сбора данных предназначены для сбора данных, необходимо использовать RSLinx Classic Gateway. В этом случае вы должны будете сконфигурировать раздел для станции с RSLinx Classic Gateway. Приложения OPC могут удаленно соединяться с RSLinx Classic Gateway и собирать данные, используя его раздел.

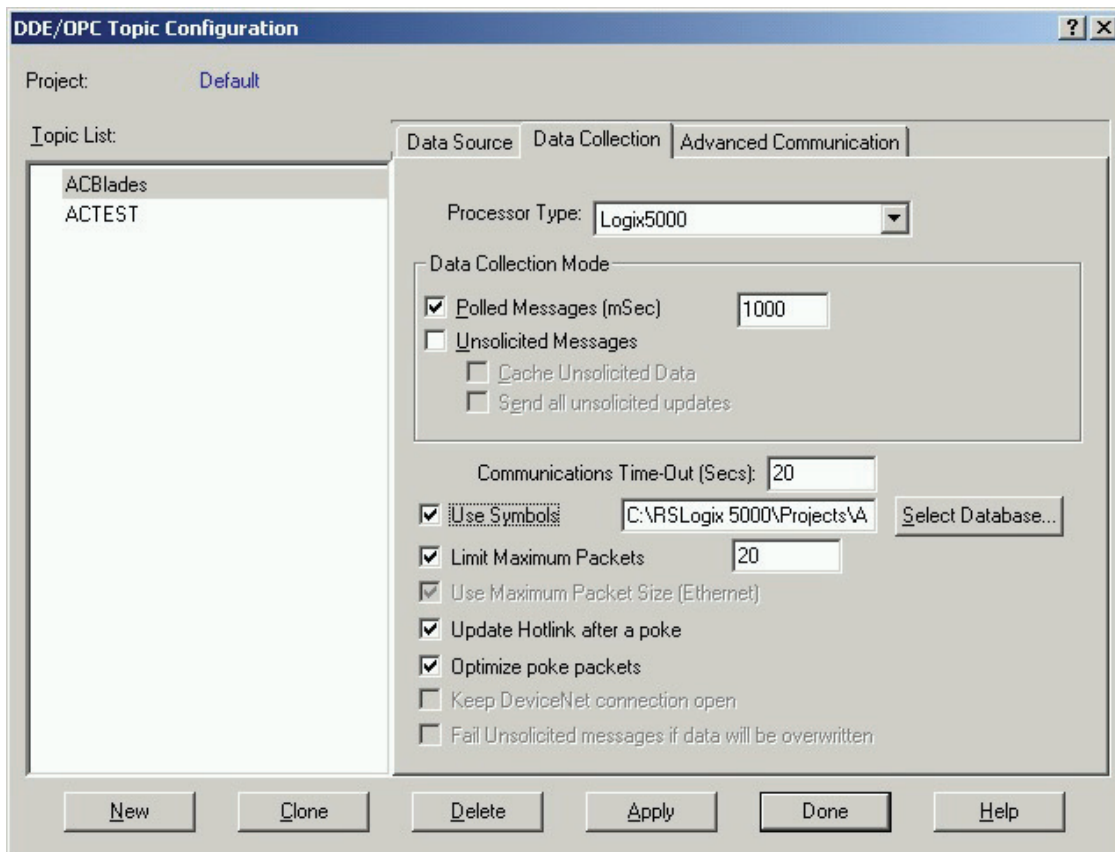
Ниже представлен самый простой способ создания раздела. При этом предполагается, что драйвер уже сконфигурирован для соответствующей сети. В этом примере будет использоваться сеть Ethernet.

Шаг 1: В окне RSWho щелкните правой кнопкой мыши на процессоре ControlLogix, для которого вы хотите сконфигурировать раздел. Выберите "Configure New DDE/OPC Topic". Откроется диалоговое окно DDE/OPC Topic Configuration.



Для конфигурации раздела щелкните правой кнопкой мыши на ControlLogix в RSWho.

Шаг 2: Создание раздела с именем управляющей программы. На снимке экрана ниже показана закладка Data Collection (Сбор данных) диалогового окна. В ней содержатся некоторые дополнительные параметры, которые будут описаны ниже.



Диалоговое окно RSLinx Classic DDE/OPC Topic Configuration (Конфигурирование раздела RSLinx Classic DDE/OPC). Здесь представлены свойства раздела ControlLogix.

Параметры раздела

Раздел для процессора ControlLogix должен быть сконфигурирован как для опрашиваемых (Polled), так и для незапланированных (Unsolicited) сообщений. Передача опрашиваемых сообщений используется для того, чтобы RSLinx Classic запрашивал у процессора ControlLogix требуемые элементы данных. Интервал между опросами определяет, как часто элементы данных будут считываться из контроллера. В соответствии с этим интервалом будут обновляться DDE-клиенты. OPC-клиенты будут обновляться согласно их собственным заданным интервалам.

Незапланированные (Unsolicited) сообщения

Вы должны сконфигурировать отдельный раздел для незапланированных сообщений, если вы хотите, чтобы процессор ControlLogix отправлял данные в RSLinx Classic через инструкцию передачи сообщений (MSG). Незапланированные данные, отправленные в RSLinx Classic, передаются любому клиенту, который запрашивает эти элементы данных.

Тайм-аут коммуникации

Это количество секунд, в течение которых RSLinx Classic будет ждать, пока процессор не подтвердит его запрос. По истечении этого времени RSLinx Classic выдаст ошибку тайм-аута коммуникации. Вы можете увеличить параметр Communications Time-Out (Тайм-аут коммуникации) для раздела, если вы запрашиваете большое количество данных и получаете слишком много ошибок тайм-аута коммуникации в Журнале регистрации событий связи RSLinx Classic (RSLinx Classic Communication Event Log). Сообщение об ошибке тайм-аута

коммуникации для ControlLogix выглядит так: “CIP Messaging Error: a message timed out waiting for a response.”

Использование символов

В данном случае вы привязываете .acd файл RSLogix 5000 к разделу (Topic) для просмотра в режиме Offline. Это позволит сконфигурировать ваши приложения сбора данных в режиме Offline, так как список доступных тегов привязан к разделу. Вы должны иметь RSLogix5000 версии 10 или более поздней версии, чтобы воспользоваться данной возможностью.

Оптимизация пакетов записи (Poke Packets) – Разрешает/Запрещает оптимизацию записи. Эта опция разрешена по умолчанию при создании нового раздела в RSLinx версии 2.30.02 и более поздних версий. Если вы хотите установить новую версию RSLinx Classic или собираетесь отправлять загруженные пакеты/рецепты или другие данные в процессор ControlLogix, убедитесь, что эта опция разрешена. Эта опция доступна только для процессоров ControlLogix версии прошивки 10.x (или более поздней) и RSLinx версии 2.30.02 (или более поздней).

RSLogix 5000 автоматически конфигурирует разделы в RSLinx Classic

При сохранении проекта .acd RSLogix 5000 (версия 10 и более поздние версии) автоматически конфигурирует раздел в RSLinx Classic. Это позволит связать .acd файл с разделом RSLinx Classic. Это также дает возможность приложениям, распознающим браузер OPC, получить список доступных тегов процессора ControlLogix в режиме Offline. Это понадобится при создании проекта RSView, если у вас нет доступа к процессору ControlLogix, который будет работать с проектом.

Понимание механизма сбора данных ControlLogix

Сбор данных с помощью процессоров ControlLogix сильно отличается от сбора данных с помощью контроллеров PLC-5 и SLC. В этих ранних контроллерах теги контроллеров структурированы в таблицы данных, так что RSLinx Classic заранее известны типы данных тегов и то, как они структурированы. Гибкость тегов ControlLogix и типа данных освобождает пользователей от необходимости использовать ограничительные определения данных и позволяет им осмысленно моделировать свои собственные данные. Эта гибкость приводит к некоторым системным издержкам, так как RSLinx Classic изначально ничего неизвестно о типах данных и о том, какие теги доступны.

Когда приложение сбора данных первоначально запрашивает данные из ControlLogix, то RSLinx Classic выгружает список всех доступных тегов и их типов данных. RSLinx Classic будет хранить эту информацию в памяти. Каждый раз, когда приложение OPC или DDE запрашивает данные, RSLinx Classic определяет, как оптимально запросить данные для ControlLogix, помещая до 486 байтов данных в каждый запрос. Каждый из этих запросов из RSLinx Classic в ControlLogix – это так называемый *Оптимизированный Пакет ControlLogix (ControlLogix Optimized Packet)*. Размер этого пакета ограничен размером пакета сети ControlNet, которая является по существу сетью, которую использует задняя шина шасси 1756.

Шаг за шагом

Считывание данных ControlLogix:

1. Приложение сбора данных помещает теги на скан через интерфейсы OPC и DDE RSLinx Classic.
2. RSLinx Classic выгружает список всех доступных тегов ControlLogix и их типов данных. RSLinx Classic хранит этот список в памяти.
3. RSLinx Classic создает оптимизированные пакеты ControlLogix. Каждый оптимизированный пакет может содержать до 488 байтов данных.
4. Как только создаются пакеты, данные могут передаваться из RSLinx Classic в приложения сбора данных через OPC или DDE. Чтобы поместить несколько тысяч тегов на скан, может потребоваться несколько секунд.

Примечание: RSLinx 2.30 SP1 и ControlLogix версии прошивки 10.x и более поздней версии используют более быстрый метод помещения и снятия тегов со скана.

Теги, снятые/помещенные на скан: (Как правило, при переключении экранов HMI)

5. При снятии тегов со скана RSLinx Classic удаляет эти теги из их соответствующих оптимизированных пакетов ControlLogix (ControlLogix Optimized Packets). RSLinx Classic вновь оптимизирует любые теги, находящиеся на скане, после удаления и создает как можно меньше пакетов. Все данные из ControlLogix не будут обновляться до тех пор, пока не завершится процесс вторичной оптимизации. ControlLogix не может отправлять пакеты данных во время изменения содержимого этих пакетов.
6. При помещении новых данных на скан RSLinx Classic формирует новые оптимизированные пакеты. Если теги уже находятся на скане, RSLinx Classic просто получает данные из существующих оптимизированных пакетов ControlLogix. Это как раз тот случай, когда использование RSLinx Classic Gateway для обслуживания данных на нескольких станциях сбора данных намного эффективнее использования нескольких независимых станций RSLinx Classic. Процессор ControlLogix создает и обслуживает один набор оптимизированных пакетов ControlLogix для RSLinx Classic Gateway. При использовании множества станций RSLinx Classic процессор ControlLogix должен был бы создать и обслужить набор оптимизированных пакетов ControlLogix для каждой станции RSLinx Classic, даже в том случае, если каждая станция осуществляет сбор одних и тех же данных!

Добавление, удаление или изменение тегов в процессоре ControlLogix во время сбора данных:

Как правило, станции сбора данных могут работать во время загрузки новых программ и/или внесения изменений в процессорах ControlLogix в режиме Online. Если Вы используете более раннюю версию RSLinx Classic, чем 2.30, то при любых изменениях тегов процессора ControlLogix необходимо будет остановить работу всех приложений сбора данных до начала внесения изменений и затем перезапустить их после завершения обновления процессора ControlLogix. RSLinx Classic 2.30 может обнаруживать эти изменения, но вам придется вручную инструктировать RSLinx Classic обновить информацию о тегах ControlLogix.

RSLinx 2.30.02 был модернизирован таким образом, чтобы сбор данных мог продолжаться почти непрерывно во время изменения тегов в процессоре ControlLogix. Теперь RSLinx Classic периодически проверяет процессор ControlLogix, чтобы узнать, были ли изменены его теги, и во многих случаях автоматически обновляет информацию тегов ControlLogix. Ниже приведены некоторые сценарии, по которым происходит изменение тегов ControlLogix во время сбора данных с RSLinx 2.30.02.

1. Теги добавляются в процессор ControlLogix в режиме Online с помощью RSLogix 5000:

RSLinx Classic определяет, что теги были добавлены в процессор ControlLogix, затем делает запись этого события в журнале регистрации событий связи RSLinx Classic (Communication Event Log) и потом обновляет список доступных тегов, чтобы включить туда вновь добавленные теги. Станции сбора данных могут запустить сбор данных из новых тегов, не останавливая проект сбора данных.

2. Теги добавляются в процессор ControlLogix при загрузке обновленной программы:

RSLinx Classic выявляет факт загрузки, делает запись этого события в журнале регистрации событий связи и обновляет список доступных тегов. Сбор данных останавливается во время загрузки новой программы и возобновляется после завершения процесса загрузки. Станции сбора данных могут запустить сбор данных из новых тегов, не останавливая проект сбора данных.

3. Теги удаляются из процессора ControlLogix в режиме Online с помощью RSLogix 5000:

RSLinx Classic определяет, что теги были удалены из процессора ControlLogix, делает запись этого события в журнале регистрации RSLinx Classic, а затем обновляет список доступных тегов. Все проекты сбора данных необходимо остановить перед внесением изменений. После удаления тегов из процессора ControlLogix процесс сбора данных необходимо возобновить. Во время удаления любых тегов из процессора ControlLogix из-

за элементов, находящихся на скане, теги не будут обновляться до тех пор, пока приложения сбора данных вновь не начнут работать.

- 4. Теги удаляются из процессора ControlLogix при загрузке обновленной программы:**
RSLinx Classic выявляет факт загрузки, делает запись этого события в журнале регистрации событий связи и обновляет список доступных тегов, чтобы больше не включать в него удаленные теги. Все проекты сбора данных необходимо остановить перед внесением изменений. После удаления тегов из процессора ControlLogix процесс сбора данных необходимо возобновить. Все элементы на скане перестанут обновляться после загрузки новой программы в процессор ControlLogix, что приведет к удалению тегов.
- 5. Изменение типов данных тегов в процессоре ControlLogix или изменение ссылки на теги псевдонимов (aliases):**
Все проекты сбора данных необходимо остановить перед внесением изменений. Процесс сбора данных необходимо возобновить после изменения типов данных тегов в процессоре ControlLogix. RSLinx Classic не может обновлять эту информацию во время процесса сбора данных.
- 6. Любое изменение размерности массивов тегов в процессоре ControlLogix:**
Все проекты сбора данных необходимо остановить перед внесением изменений. Процесс сбора данных необходимо возобновить после изменения типов данных тегов в процессоре ControlLogix. RSLinx Classic не может обновлять эту информацию во время процесса сбора данных.
- 7. Изменения любого вида в структурах UDT (User Defined Types) при добавлении, удалении или изменении их элементов в процессоре ControlLogix:**
Все проекты сбора данных необходимо остановить перед внесением изменений. Процесс сбора данных необходимо возобновить после изменения типов данных тегов в процессоре ControlLogix. RSLinx Classic не может обновлять эту информацию во время процесса сбора данных.

Оптимизация чтения

Оптимизация чтения позволяет считывать множество тегов из процессора ControlLogix при меньшем количестве запросов. Это экономит ресурсы центрального процессора и памяти ControlLogix, а также уменьшает сетевой трафик. Запросы от приложений сбора данных на RSLinx Classic оптимизируются до меньшего количества запросов от RSLinx Classic на ControlLogix. Это меньшее число запросов и есть оптимизированные пакеты ControlLogix (ControlLogix Optimized Packet). Без такой оптимизации каждый запрос тега из приложений сбора данных соответствовал бы запросу из RSLinx Classic на ControlLogix.

Существует несколько различных уровней оптимизации. А именно, UDT (структуры пользователя), Array (массивы), и Первый уровень или Scattered (Оптимизация Разбросанных данных). RSLinx Classic определяет вид используемой оптимизации, когда теги помещаются на скан. Эти типы оптимизации описываются в следующих нескольких разделах этого документа.

Оптимизация массива

Этот вид оптимизации охватывает несколько элементов массива и обрабатывает их как один элемент контроллера. Преимущества этого вида оптимизации часто бывают незаметны, поскольку клиенты HMI зачастую вынуждены запрашивать отдельно каждый элемент массива, а не все элементы сразу. Другими словами, чтобы получить максимальную оптимизацию, надо издать запрос типа: INTArray [0], L10. Вместо отдельных запросов INTArray [0], INTArray [1], ..., INTArray [9]. Если вы не можете использовать систему обозначений L, оптимизация массива все равно будет происходить между ControlLogix и RSLinx Classic, но будет использовано больше ресурсов компьютера.

Требования:

- Одномерный массив типа BOOL, SINT, INT, DINT, или REAL.
- Вариантов настройки в RSLinx Classic нет. Оптимизация массива происходит автоматически.
- Если размер запрашиваемого массива превышает 488 байтов, RSLinx Classic все равно будет проводить оптимизацию массива, но при этом будет использовано большее количество пакетов. Постарайтесь использовать следующие теги:

Сценарий:

Задается массив INTArray[100]. Запрашиваются следующие теги: INTArray[0], INTArray [20] и INTArray[90]. RSLinx Classic создает один пакет, запрашивающий все элементы между INTArray [0] и INTArray [90]. ControlLogix обрабатывает все эти элементы массива как один элемент, и для передачи этих тегов используется один оптимизированный пакет ControlLogix. Если после этого будет запрошен массив INTArray [60], RSLinx Classic будет уже иметь его на скане, и для передачи этих данных не понадобятся дополнительные издержки.

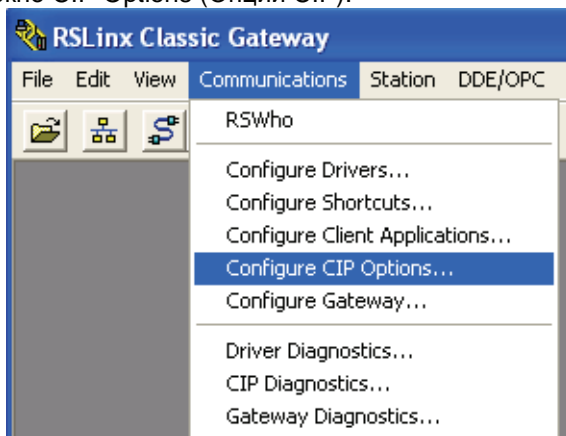
Оптимизация Булева массива (Boolean Array)

Начиная с версии RSLinx 2.20, благодаря оптимизации булевых массивов каждый индивидуальный логический элемент можно рассматривать как отдельный битовый объект, а не 32-разрядный объект, как это было ранее. Используя этот метод, в один оптимизированный пакет ControlLogix можно уместить до 3872 логических выражений. Ранее можно было уместить максимум 81 логическое выражение. Когда речь идет об индивидуальных булевых тегах, а не о тегах булевых массивов, теги будут рассматриваться как 32-разрядные элементы, и только 81 элемент сможет войти в оптимизированный пакет ControlLogix.

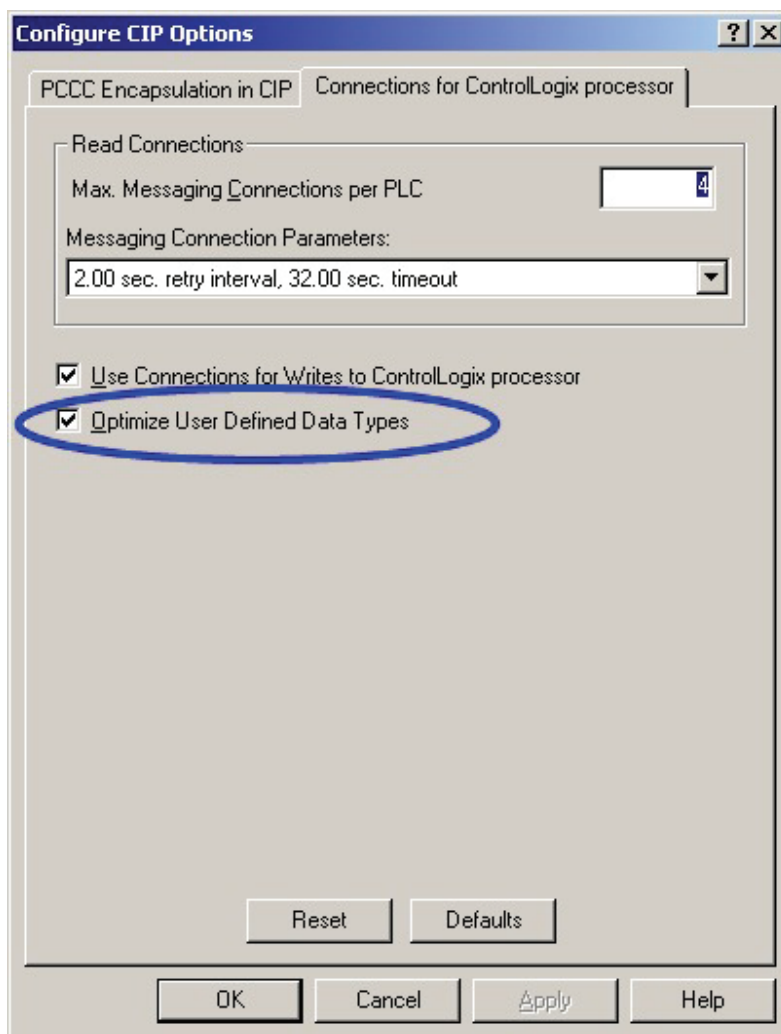
Оптимизация UDT

Тип данных UDT (User Defined Type - Определяемый Пользователем Тип) – это пользовательский тип данных в процессоре ControlLogix. Пользователи создают типы UDT, содержащие собственные типы данных, такие как INT, DINT, FLOAT, другие UDT и т.д. Когда элементы типа UDT запрашиваются из приложений сбора данных и опция *UDT Optimization* разрешена в RSLinx Classic, то RSLinx Classic получит всю структуру UDT в качестве одного элемента из процессора ControlLogix. Это позволит сократить использование ресурсов центрального процессора и памяти ControlLogix, необходимых для обслуживания данных. Чем больше элементов UDT используется в приложениях сбора данных, тем выше производительность. Если на скане находится незначительное количество элементов UDT, то полностью обслуживать всю структуру UDT в RSLinx Classic будет целесообразно. В таком сценарии оптимизацию лучше запретить. Оптимизацию UDT можно разрешить или запретить для всех процессоров ControlLogix/разделов. Эту опцию нельзя настроить конкретно для каждого раздела. Ниже описано, как настроить данную опцию:

1. Откройте диалоговое окно CIP Options (Опции CIP).



2. Опция Optimize User Defined Data Types (Оптимизировать UDT) находится в закладке Connection for ControlLogix Processor (Подключение для процессора ControlLogix) диалогового окна CIP Options.



Диалоговое окно CIP Options в RSLinx Classic

Требования по оптимизации UDT:

- Вышеупомянутая опция должна быть разрешена в RSLinx Classic.
- Структура UDT должна иметь размер не более 480 байтов.

Некоторые сценарии оптимизации UDT:

1. Размер UDT превышает 480 байтов:

RSLinx Classic не будет осуществлять оптимизацию UDT, а осуществит только Оптимизацию Разбросанных данных (Scattered).

2. UDT содержит массивы типа SINTS, INTS, DINTS или REALS:

RSLinx Classic выполнит оптимизацию UDT и получит всю структуру. Если размер UDT превышает 480 байтов, то оптимизация будет выполняться на любых массивах. “Разбросанная” оптимизация будет выполнена только на оставшихся элементах.

3. Массив элементов UDTs:

RSLinx Classic не будет выполнять оптимизацию UDT для каждого UDT-элемента массива.

4. UDT, содержащийся в UDT:

RSLinx Classic будет выполнять оптимизацию UDT на внешнем UDT, если его размер составляет менее 480 байтов.

Оптимизация первого уровня или Оптимизация Разбросанных данных (Scattered)

Это оптимизация, заданная по умолчанию, которая применяется в том случае, когда невозможно выполнить оптимизацию UDT или массива. RSLinx Classic поместит теги размером до 480 байт в оптимизированный пакет ControlLogix (ControlLogix Optimized Packet). Теги обычно «Разбросаны» по памяти процессора ControlLogix. Для данного типа оптимизации необходимо большее количество ресурсов и памяти процессора ControlLogix, чем для оптимизации массива или UDT.

Сколько данных можно прочитать в одном оптимизированном пакете ControlLogix (ControlLogix Optimized Packet)?

В данной таблице показано, сколько элементов каждого собственного типа данных процессора ControlLogix может войти в один оптимизированный пакет ControlLogix. Эти данные соответствуют версии RSLinx 2.20 и более поздним версиям.

Тип данных	Максимальное число элементов, которые можно уместить в пакет для чтения
Boolean	81
Boolean Arrayed	3872
SINT	81
SINT Arrayed	81
INT	121
INT Arrayed	243
DINT	81
DINT Arrayed	121
REAL	81
REAL Arrayed	121

Оптимизация записи

Оптимизация записи позволяет повысить производительность при загрузке пакетов и рецептов в процессор ControlLogix. Зачастую процессор ControlLogix может быть настолько перегружен, что при запросах на запись иногда возникает тайм-аут и загрузку пакетов/рецептов осуществить невозможно. При оптимизации записи меньше запросов на запись будет отправляться из RSLinx Classic на процессор ControlLogix. Меньшее количество запросов записи уменьшает количество работы, которую процессор ControlLogix должен выполнить для принятия загруженных пакетов/рецептов, а также сокращает возможность отказов при их загрузке.

Каждый запрос на запись из RSLinx Classic на процессор ControlLogix называется *Оптимизированным сетевым пакетом записи (Optimized Network Write Packet)*. До выхода версии RSLinx 2.30.02 любой отдельный записываемый тег необходимо было отправлять в своем собственном Оптимизированном пакете записи. Теперь с использованием оптимизации записи множество записываемых тегов можно отправлять при помощи меньшего количества Оптимизированных сетевых пакетов записи. Для тегов одного процессора ControlLogix, которые имеют лишь косвенное отношение друг к другу или не имеют его вовсе, для оптимизации записи используется оптимизация первого уровня (First Level) или Оптимизация Разбросанных данных (Scattered). Оптимизация записи группирует их в как можно меньшее количество Оптимизированных сетевых пакетов записи. Кроме того, возможна также оптимизация массивов, но оптимизация UDT не производится.

Как использовать оптимизацию записи

Для раздела RSLinx Classic, осуществляющего связь с процессором ControlLogix, опция Optimize Poke Packets (Оптимизация пакетов записи) должна быть разрешена. За дополнительной информацией обращайтесь к разделу Конфигурирование RSLinx Classic. Кроме того, вы должны использовать версию прошивки ControlLogix 10.x или более позднюю версию.

Сбор данных ControlLogix в RSLinx Classic

Оптимизация массива при записи

Она схожа с оптимизацией массива при чтении. Элементы массива могут быть оптимизированы, чтобы можно было поместить еще большее количество данных в Оптимизированный сетевой пакет записи (Optimized Network Write Packet). Чтобы воспользоваться оптимизацией массива, элементы массива должны записываться непрерывно, и запись должна осуществляться по одной команде.

Булевы массивы – заметим, что при использовании оптимизации записи для булевых массивов нельзя использовать указатель "L". Каждый элемент массива необходимо рассматривать индивидуально как отдельный тег

Сколько тегов можно отправить в ControlLogix в одном пакете записи?

Тип данных	Максимальное число элементов, которые можно уместить в пакет записи
Boolean	32
Boolean Arrayed	32
SINT	32
SINT Arrayed	460
INT	30
INT Arrayed	230
DINT	26
DINT Arrayed	115
REAL	26
REAL Arrayed	115

Эти данные используются в версиях ControlLogix 10.x и более поздних версиях. В предыдущих версиях ControlLogix эти показатели были ниже, так как оптимизированные пакеты записи должны были содержать фактические имена тегов. Поэтому мы рекомендуем использовать ControlLogix версии 10.x или более поздней версии для получения более высокой производительности. В версиях, выпущенных до RSLinx 2.30.02, оптимизация записи не поддерживается.

Адаптивный опрос (Smart Polling) RSLinx Classic

С помощью адаптивного опроса RSLinx Classic можно уменьшить использование ресурсов центрального процессора компьютера системы RSLinx Classic при сборе данных. Если данные запрашиваются быстрее чем контроллер и сеть могут их отправить (например, при использовании слишком высокой скорости опроса на более медленной сети), то RSLinx Classic будет "тормозить" чрезмерный опрос, и запрашивать данные из процессора на той скорости, на которой они фактически отправляются. Эта возможность поддерживается в версии RSLinx 2.20 и более поздних версиях. Параметров настройки не существует, все операции выполняются автоматически.

Как RSLinx Classic использует Подключения (Connections) ControlLogix

При сборе данных из ControlLogix RSLinx Classic использует Подключения, то есть специальные каналы связи. При чтении данных из процессора ControlLogix каждый Оптимизированный пакет ControlLogix (ControlLogix Optimized Packet) использует данные Подключения. Он получает свое Подключение из пула Подключений (pool of connections), общего для всех Оптимизированных пакетов ControlLogix. Максимальное количество Подключений, которые используются в этом пуле, можно изменить, но по умолчанию задано четыре Подключения, так как во многих случаях это количество является оптимальным.

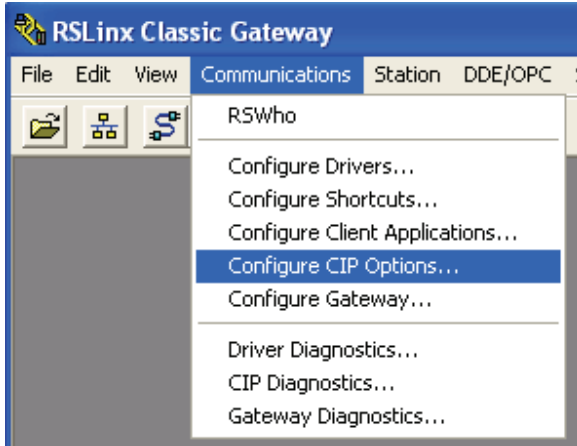
Использование Подключений при записи данных RSLinx Classic в процессор ControlLogix отличается от их использования при чтении. При записи данных в процессор ControlLogix он может использовать одно Подключение, или вообще не использовать ни одного. Если запись в процессор ControlLogix происходит достаточно часто (в пределах 32 сек. по умолчанию) или непрерывно, тогда вам лучше использовать Подключения для записи. Если вы отправляете

Сбор данных ControlLogix в RSLinx Classic

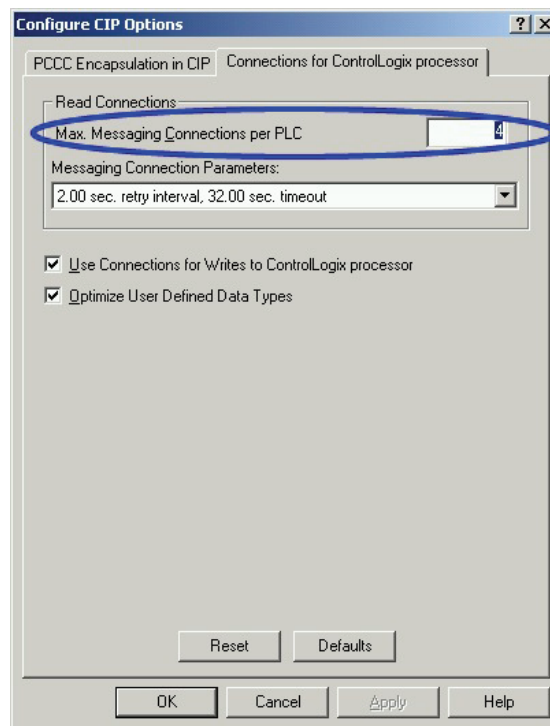
данные реже, то нет необходимости использовать Подключения для записи. Например, если пакет/набор команд запускается каждые несколько минут, то вы можете не использовать Подключения для записи.

Ниже описано, как получить доступ к диалоговому окну, в котором можно задать максимальное количество Подключений (Connections) для чтения на каждый ControlLogix и разрешить/запретить использование Подключений для записи:

1. В главном меню RSLinx Classic выберите Communications (Коммуникации) ...Configure CIP Options (Конфигурирование опций CIP).



2. Перейдите на закладку Connections for ControlLogix Processor (Подключения для процессора ControlLogix).
3. Максимальное количество Подключений, которые будет использовать RSLinx Classic для каждого процессора ControlLogix, можно задать в этой закладке. В этом диалоговом окне также можно разрешить/запретить использование Подключений для записи.



Диалоговое окно Configure CIP Options (Конфигурирование опций CIP) RSLinx Classic

Использование памяти в процессоре ControlLogix

Процессор ControlLogix выделяет некоторый объем памяти для внутренней поддержки структур Оптимизированных пакетов ControlLogix (ControlLogix Optimized Packets). Объем памяти, который будет использоваться в ControlLogix, зависит от:

- Количества Подключений для передачи сообщений на процессор (На одно Подключение приходится около 1.1К памяти)
- Общего количества оптимизированных пакетов ControlLogix (На один оптимизированный пакет приходится около 80 байтов)
- Общего количества оптимизированных элементов ControlLogix (На один элемент приходится около 32 байтов)

Ниже приводятся некоторые результаты тестов, выполненных для определения фактического объема памяти, используемой в процессоре для различных запросов данных. Вычисления для оптимизированных элементов (Optimized Items) и оптимизированных пакетов (Optimized Packets) были получены с помощью диагностики RSLinx Classic, а объем используемой памяти был определен с помощью элемента “@freemem” в DDE/OPC.

Для элементов, запрашиваемых в массивах, или структурах и UDT потребуется меньше памяти, чем прежде, если будут использованы соответствующие типы оптимизации. Это связано с тем, что оптимизации массивов и UDT в целом мало, в чем уступают Оптимизированным пакетам ControlLogix (ControlLogix Optimized Packets).

За дополнительной информацией об оптимизациях массивов и UDT обращайтесь к разделу 2.

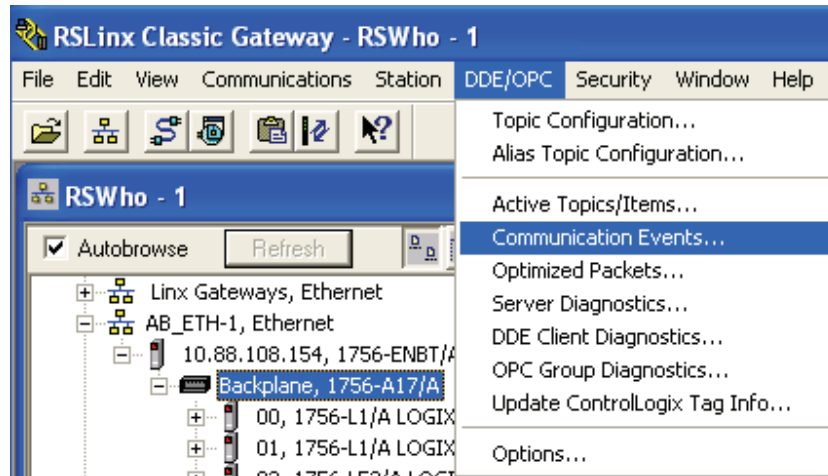
Испытательная установка	Оптимизации массива?	Оптимизированные элементы ControlLogix	Оптимизированные пакеты ControlLogix	Подключения, используемые в процессоре	Память, используемая в процессоре
1 целый тег	Оптимизации массива нет	1	1	1	1364 байта
	Полная оптимизация массива	1	1	1	1364 байта
100 целых тегов	Оптимизации массива нет	100	1	1	4532 байта
	Полная оптимизация массива	1	1	1	1560 байтов
500 целых тегов	Оптимизации массива нет	500	5	5	22660 байтов
	Полная оптимизация массива	3	3	3	5084 байта
1000 целых тегов	Оптимизации массива нет	1000	9	8*	42736 байтов
	Полная оптимизация массива	5	5	5	7556 байтов

* При выполнении тестирования RSLinx Classic был сконфигурирован для использования 8 Подключений.

Журнал регистрации событий связи (Communications Event Log)

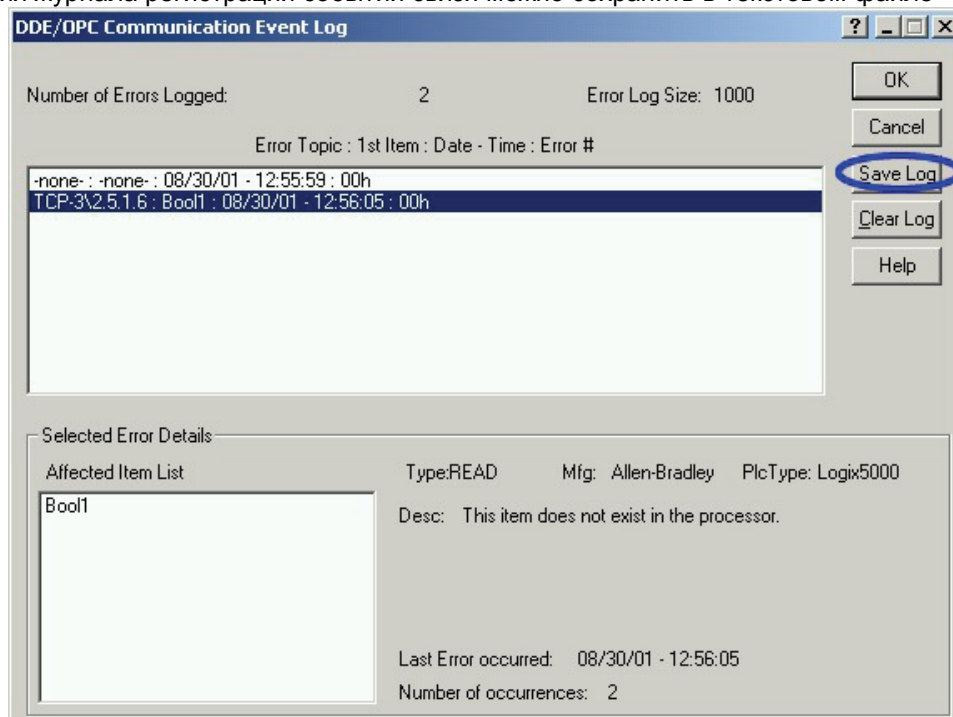
В журнал регистрации событий связи RSLinx Classic вносятся события, которые могут произойти при выполнении сбора данных. Некоторые из них могут означать сбой связи, другие же просто указывают на возможность возникновения проблемы. Содержимое этого журнала можно экспортировать в текстовый файл.

1. Как получить доступ к журналу регистрации событий связи из главного меню RSLinx Classic.



Открытие журнала регистрации событий связи (Communication Event Log) из главного меню RSLinx Classic

2. События журнала регистрации событий связи можно сохранить в текстовом файле



Журнал регистрации событий связи RSLinx Classic

Получение данных RSLinx Classic в приложениях

Утилита импорта тегов Logix5000 (Logix5000 Tag Import Utility) – импорт тегов ControlLogix в RSVIEW и RSSql

Используйте утилиту импорта тегов Logix5000 для импорта тегов ControlLogix из файла L5K в RSVIEW или RSSql. Эта утилита генерирует CSV файл, который можно импортировать в RSVIEW ME и RSVIEW SE. Она также генерирует TSV файл для RSSql. RSVIEW32 может импортировать теги непосредственно из утилиты.

Вы можете загрузить эту утилиту и примечание по применению непосредственно из интерактивной базы знаний Rockwell Software (Rockwell Software's Online Knowledgebase):

[A839 Logix 5000 Tag Import Utility for RSVIEW32, RSVIEW ME / SE and RSSql \(RSVIEW32\)](#)

Просмотр и выбор тегов ControlLogix

Другим способом передачи тегов в приложения является просмотр (browsing) и выбор доступных тегов ControlLogix. RSVIEW и RSSql - примеры приложений, которые поддерживают интерфейс OPCBrowse. Обратите внимание, что вы можете просмотреть список доступных тегов в режиме Offline и Online.

Online - RSLinx Classic напрямую запрашивает у контроллера список доступных тегов.

Offline – RSLinx Classic запрашивает ACD файл RSLogix 5000. Раздел должен быть сконфигурирован, чтобы был выбран параметра *Use Symbols* (Использование символов) с файлом ACD. RSLogix 5000 автоматически создаст его при сохранении проекта. Эта возможность поддерживается в RSLinx 2.30 SP1 и RSLogix версии 10 и более поздних версиях.

Ниже приводится пример просмотра тегов ControlLogix из RSVIEW:

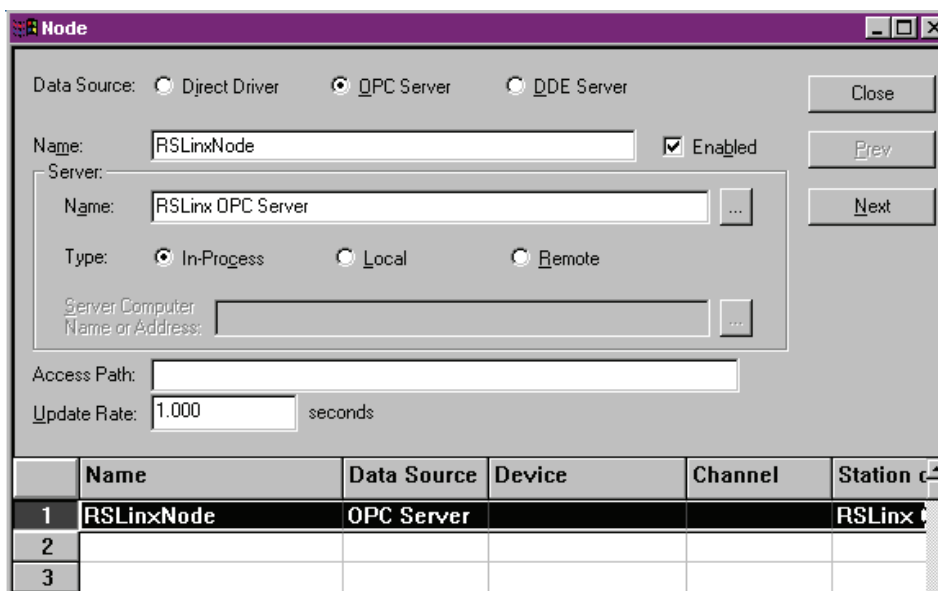
1. Создайте новый узел (Node) в RSVIEW с нижеприведенными настройками:

Источник данных - OPC-сервер ("OPC Server")

Имя сервера - "RSLinx OPC Server"

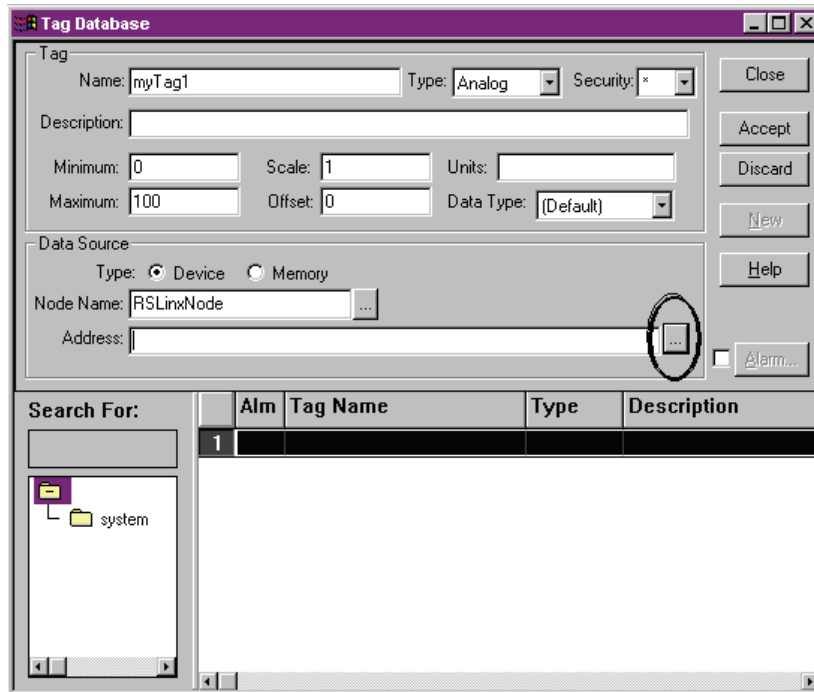
Необходимо выбрать тип сервера "In-Process".

Поле Access Path (Путь доступа) оставьте пустым.

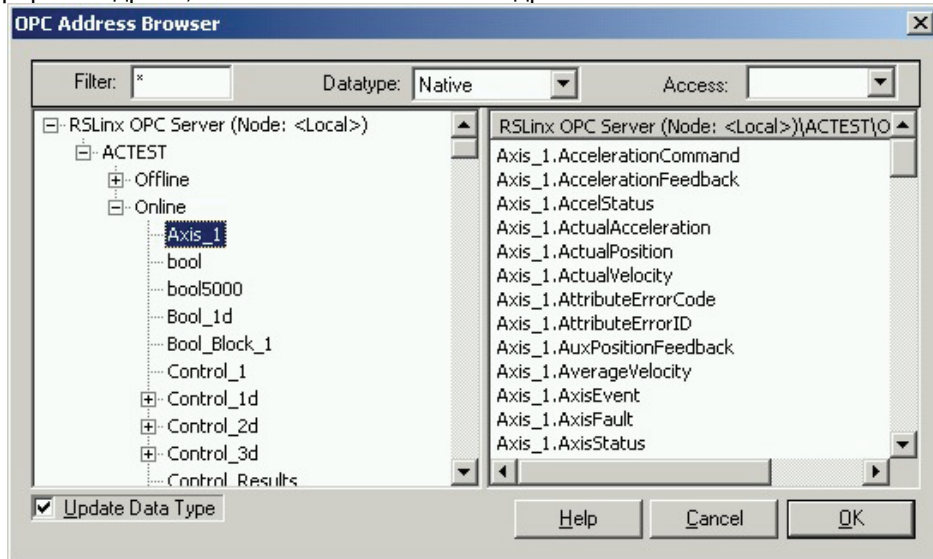


Сбор данных ControlLogix в RSLinx Classic

- 2) В базе данных тегов RSView (RSView Tag Database) создайте тег со следующими настройками:
- По желанию выберите тип тега “Analog” (Аналоговый) или “Digital” (Цифровой).
 - Выберите тип источника данных тега “Device” (Устройство).
 - Используйте имя узла (Node Name), заданного в пункте 1.
 - Рядом с полем “Address” (Адрес) появится кнопка с тремя точками (Она обведена на рисунке, изображенном ниже). Нажмите на эту кнопку.



- 3) Теперь в RSView должно появиться окно OPC Address Browser. Здесь вы можете выбрать раздел RSLinx Classic для вашего процессора ControlLogix и просмотреть иерархию адреса, чтобы связать ваш тег с адресом.

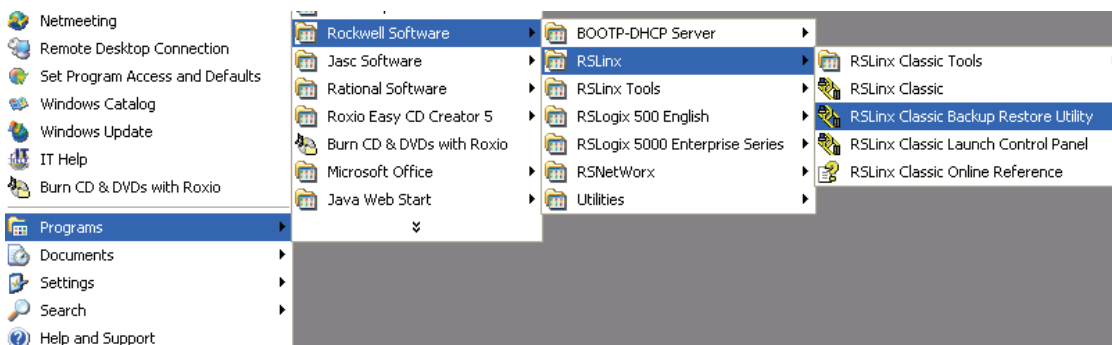


Просмотр тегов ControlLogix в RSView

Некоторые особенности RSLinx Classic и советы

Сохранение/восстановление конфигураций RSLinx Classic в/из файла

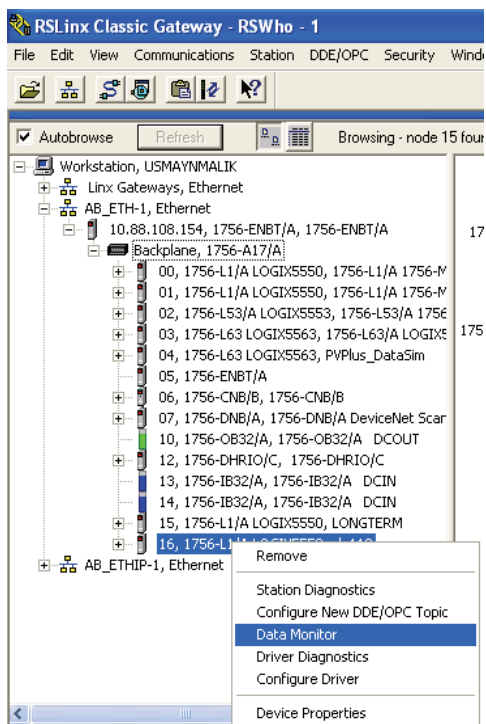
К утилите Backup/Restore RSLinx Configuration можно обратиться через меню Start (Пуск). Ниже смотрите снимок экрана. Все параметры настройки RSLinx Classic, включая драйверы и разделы, сохраняются/восстанавливаются из файла. Эта утилита также доступна из командной строки с помощью файла Backupcmd.exe, который находится там, где установлен RSLinx.exe.



Доступ к утилите RSLinx Classic Backup-Restore (Сохранение-Восстановление) из меню Пуск в Windows.

Мониторинг данных ControlLogix

RSLinx версии 2.20 и более поздние версии поддерживают функцию мониторинга данных (Data Monitor) для процессоров ControlLogix. Щелкните правой кнопкой мыши на процессоре ControlLogix в RSWWho и выберите опцию "Data Monitor", чтобы осуществлять контроль над вашими тегами. Мониторинг данных доступен только в RSLinx Classic Professional и RSLinx Classic Gateway.



Всплывающее меню RSWWho для процессора ControlLogix

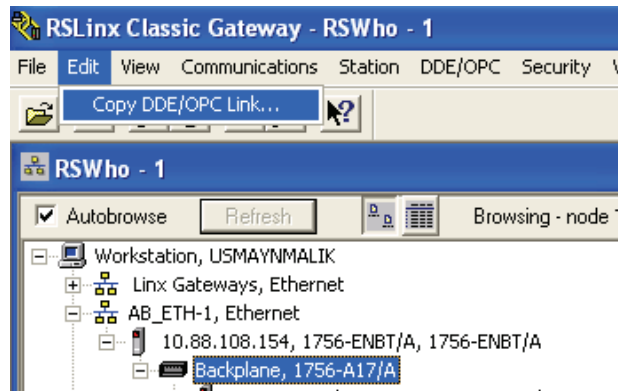
Сбор данных ControlLogix в RSLinx Classic

Tag Name	Value	Radix	Type
Counters_Clear	0	Decimal	BOOL
DATA_TABLE			
-Zero_INT	0	Decimal	DINT
-Zero_Float	0	Float	REAL
-Pos_INT	16845	Decimal	INT
-Pos_Float	3.14159	Float	REAL
-Neg_INT	-3185	Decimal	INT
-Neg_Float	-77.9	Float	REAL
-Min_INT	-32768	Decimal	INT
-Min_Float	-1.17549e-038	Float	REAL
-Min_DINT	-2147483648	Decimal	DINT
-Max_INT	32768	Decimal	INT

Мониторинг данных ControlLogix

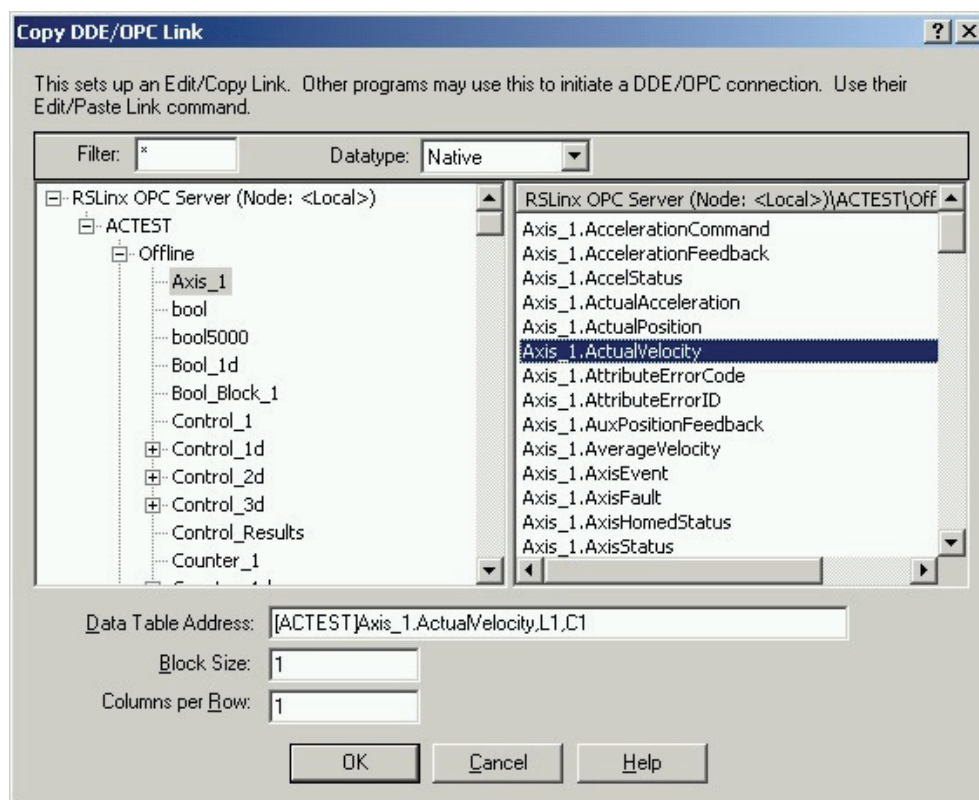
Совет: Щелкните правой кнопкой мыши на тегах и выберите *Copy To Clipboard* (Копировать в Буфер обмена). Это передаст в буфер обмена копию тега, который вы выбрали, и даст возможность вставить ссылку (hotlink) в Microsoft Excel или имя тега в другое приложение. RSLinx Classic будет использовать существующий раздел, но если ни один раздел не был сконфигурирован, RSLinx Classic потребует выполнения конфигурирования.

Функция RSLinx Classic "Copy DDE/OPC link" поддерживает просмотр тегов ControlLogix



Выбор опции Copy DDE/OPC Link из главного меню RSLinx Classic

После того, как вы выберете ваш тег, вы сможете вставить ссылку DDE (DDE hotlink) в Microsoft Excel или просто вставить имя тега в другое приложение (например, в VB).



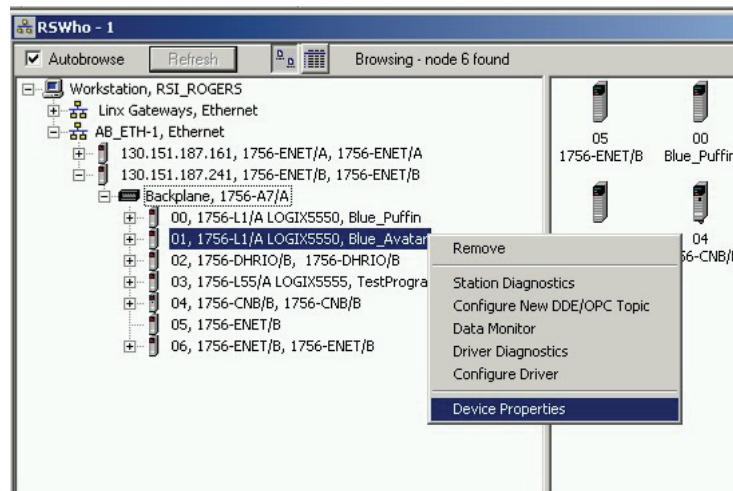
Диалоговое окно Copy DDE/OPC Link.

Получение информации о модулях в RSLinx Classic

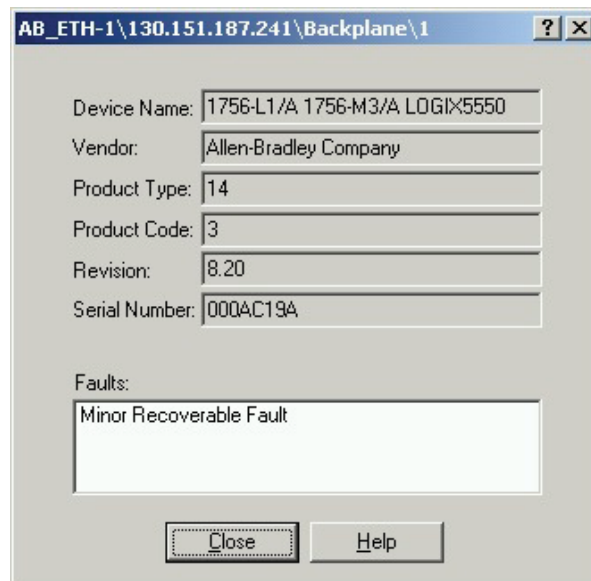
Чтобы узнать, какая версия прошивки (Firmware version) установлена в процессоре ControlLogix, выполните следующие действия:

(Применяется для версий RSLinx Classic 2.20 и более поздних версий)

1. Щелкните правой кнопкой мыши на процессоре ControlLogix в RSWho, чтобы открыть всплывающее меню, показанное ниже.



2. Выберите пункт меню под названием "Device Properties" (Свойства устройства), чтобы открыть диалоговое окно, содержащее информацию об устройстве.



В этом диалоговом окне вы сможете увидеть информацию о версии прошивки ControlLogix в поле Revision (Модификация). В данном примере мы видим, что версия прошивки - 8.20. Обратите внимание, что эту возможность поддерживает большинство устройств, а не только процессоры ControlLogix.

Замечания по применению

Вы можете найти многие технические примечания в интерактивной базе знаний Rockwell Software. Многие разделы этой главы соответствуют техническим примечаниям (указан номер документа и заголовок, а также даны гиперссылки). Интерактивная база знаний находится здесь: <http://support.rockwellautomation.com/>

Одновременное помещение тегов на скан

Смотрите технические примечания, данные в интерактивной базе знаний Rockwell Software: [A2062 ControlLogix communication packet optimization techniques and observations \(RSView32\)](#)

Изменение тегов ControlLogix, находящихся на скане

Возможно, вы захотите изменить теги в процессоре ControlLogix при выполнении сбора данных. Действия, которые вы должны выполнить, зависят от изменений, которые вы хотите внести. RSLinx Classic выгружает все теги ControlLogix и их типы данных, как только первый тег помещается на скан. RSLinx Classic использует эту информацию для формирования Оптимизированных пакетов ControlLogix. Если теги в ControlLogix меняются, то RSLinx Classic должен будет выгрузить новую информацию о тегах. Ниже описываются действия, которые необходимо выполнить при изменении тегов в процессоре ControlLogix.

Меняются типы данных, размеры массива, описания UDT или теги удаляются из процессора. Здесь необходимо выполнить, даже если теги не помещены на скан при помощи приложений сбора данных, следующее:

1. Отключите приложения сбора данных от RSLinx Classic. Если соединение осуществляется удаленно через RSLinx Classic Gateway, отключите все станции сбора данных. При этом RSLinx Classic не нужно закрывать или перезапускать. Компьютеры также не следует перезагружать. Например, в RSView есть команда запрещения узла (node disable command), которую можно привязать к кнопке или выполнить как часть скрипта.
2. После внесения изменений в процессор ControlLogix перезапустите приложения сбора данных.

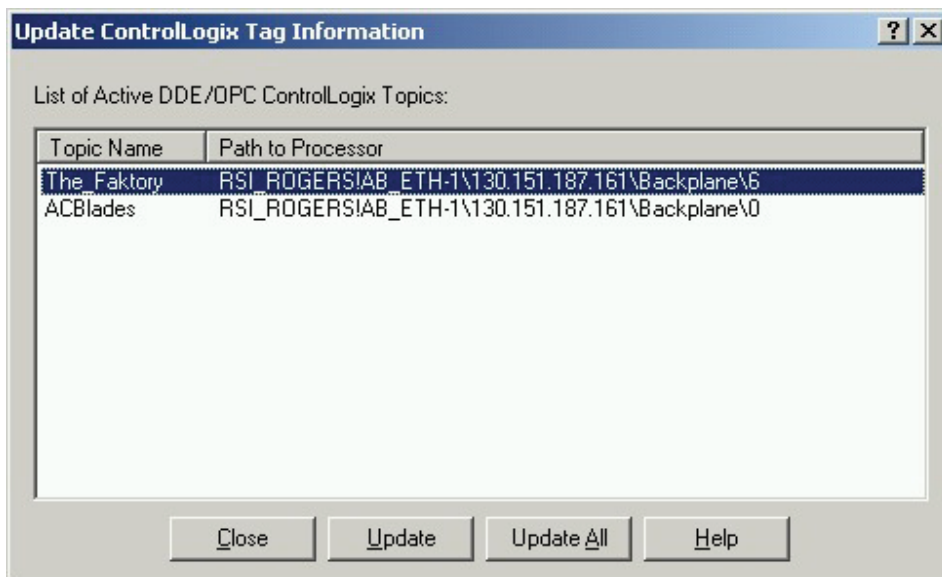
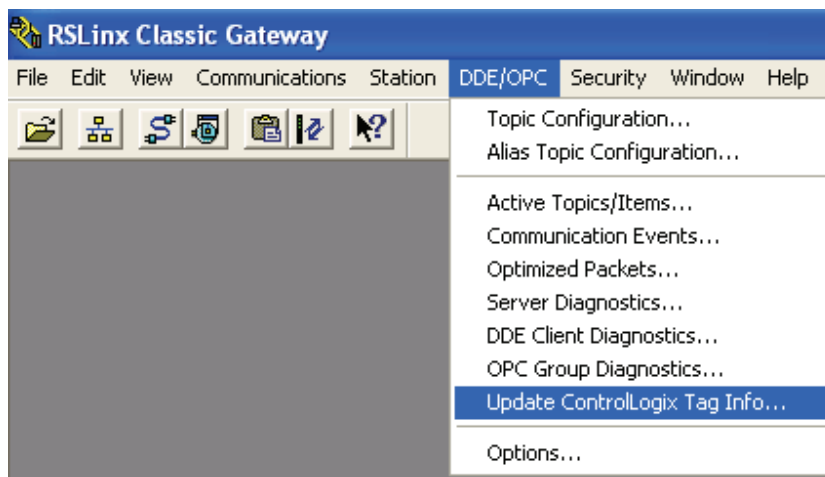
Если теги добавляются к контроллеру ControlLogix, но изменения типов данных или удаления тегов не происходит:

Вариант 1 (необходима версия RSLinx 2.30 или более поздние версии)

1. Добавьте теги в RSLinx 5000 в режиме Offline
2. Загрузите программу в ControlLogix
3. RSLinx Classic автоматически внесет событие в журнал регистрации событий связи (Communication Event Log) и выгрузит новую информацию о тегах. Станции сбора данных возобновят сбор данных после завершения процесса загрузки.

Вариант 2 (необходима версия RSLinx 2.30 или более поздние версии)

1. Добавьте теги в процессор ControlLogix в режиме Online во время выполнения сбора данных.
2. Дайте команду RSLinx Classic заново выгрузить теги ControlLogix. Это можно сделать одним из трех способов:
 - a. RSLinx Classic автоматически обнаружит изменения и затем заново выгрузит информацию о тегах ControlLogix. RSLinx Classic подает запрос на эти изменения на процессор ControlLogix каждые 20-30 секунд. RSLinx Classic внесет запись этого события в журнал регистрации событий RSLinx, когда изменения будут обнаружены и выгружены (версия RSLinx 2.31 и более поздние версии).
 - b. Выполните считывание предопределенного элемента @CLTagUpdateAddressSpace
 - c. Выберите опцию в главном меню RSLinx Classic в DDE/OPC



Резервные сервера RSLinx Classic Gateway и команда NodeSwitch RSView

Смотрите технические примечания, данные в интерактивной базе знаний Rockwell Software: [R679 RSView32 6.30.16 Patch for switching between OPC nodes \(RSView32\)](#)

Резервирование процессоров ControlLogix

Существует несколько вариантов включения резервирования на уровне ControlLogix в вашу систему сбора данных.

Вариант 1: Используйте аппаратное резервирование. Вы сможете сделать это, только если сбор данных осуществляется через ControlNet.

Вариант 2: Используйте псевдоним (Alias) для разделов в RSLinx Classic.

Вариант 3: Используйте команду переключения узла (NodeSwitch) RSView.

Смотрите технические примечания, данные в интерактивной базе знаний Rockwell Software: [A952 Node Level Redundancy - A comparison of the Alias Topic and the NodeSwitch Command \(RSView32\)](#)

Использование переменных типа STRING в ControlLogix

Вы должны использовать как минимум версию RSLinx 2.3 и версию прошивки ControlLogix 8.x, чтобы использовать строки.

Версия RSLinx 2.31 и более поздние версии

Сбор данных ControlLogix в RSLinx Classic

Синтаксис адресации для строковых типов данных ControlLogix – это просто имя тега строки. Таким образом, если ваша строка имеет имя “MyString”, то в вашем приложении сбора данных к ней необходимо обращаться как “MyString”, независимо от того, сколько символов вы хотите прочитать. RSLinx автоматически обновляет длину строки в процессоре ControlLogix при записи нового значения в строку.

Версии, выпущенные до RSLinx 2.31

Синтаксис адресации в ControlLogix для строковых типов данных – это StringTag.DATA,SC# (где # - число символов). Например, если в процессоре ControlLogix имеется строковый тег по имени “MyString”, то необходимо использовать MyString.DATA,SC10, чтобы прочитать первые десять символов MyString.

Поиск неисправностей

Чтобы загрузить руководство по поиску неисправностей с нашего сайта технической поддержки, [нажмите здесь](#). Ниже приводится адрес ссылки в Интернет:
http://www.software.rockwell.com/download/comms/rslinx/controllogix_data_collection_troubleshooting_guide.doc

Производительность

Пропускная способность в зависимости от Системных Издержек (System Overhead Time Slice) для непрерывной задачи

Испытательная установка:

- Одна непрерывная задача, которая выполняется в 1756-5555 версии 10.24.
- 5000 тегов типа DINT были собраны с помощью OPC test client (тестирующего клиента). Частота обновления = 1мс
- Модуль ENBT в коммутируемой сети, имеющей скорость 100 Мбайт/с. Драйвер Ethernet в RSLinx Classic. Единственный другой узел сети являлся компьютер.
- DELL Optiplex GX260. 1.8Ghz
- RSLinx 2.40.01
- Движение или ввод/вывод не производился

Тестирование выполнялось для трех различных организаций тегов. Теги были Разбросаны, оптимизированы в UDT или в Массивах. Ниже следует описание каждой из них.

Разбросанные (Scattered) теги означают один из трех вариантов:

1. Каждый тег был индивидуально задан в процессоре ControlLogix.
2. Теги были заданы в качестве элементов в UDT с запрещенной опцией "Optimize User Defined Data Types" (Оптимизация типов данных, определенных пользователем) в Classic.
3. Теги были заданы в качестве элементов в UDT, которые оказались слишком большими для оптимизации (UDT>480 байтов).

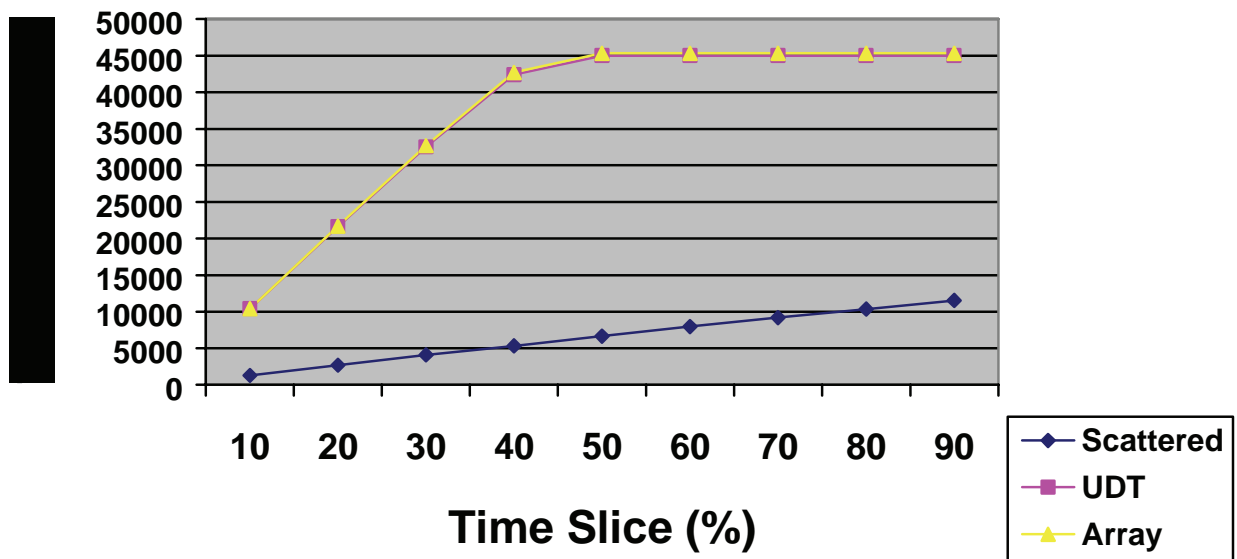
Массив (Array) – Все данные были заданы в одном большом массиве. Программа OPC Test Client поместила на скан все 5000 тегов одновременно, чтобы получить наилучшую оптимизацию массива.

UDT – 120 элементов типа DINT были заданы в UDT (UDT размером 480 байтов). Опция "Optimize User Defined Data Types" в RSLinx Classic включена.

Теги в секунду (Tags per Second) = (5000 тегов/количество оптимизированных пакетов ControlLogix (из RSLinx Classic Server Diagnostics)) * Количество пакетов в секунду (из RSLinx Classic Driver Diagnostics).

Системные Издержки	Разбросанные Теги в секунду	UDT Теги в секунду	Массивы Теги в секунду
10%	1270	10400	10400
20%	2680	21600	21700
30%	4070	32500	32700
40%	5340	42400	42700
50%	6620	45000	45300
60%	7940	45000	45300
70%	9190	45000	45300
80%	10360	45000	45300
90%	11530	45000	45300

Performance vs. Time Slice



Результаты:

Пропускная способность в тегах в секунду зависит от организации тегов и от Системных Издержек (*System Overhead Time Slice*). Тестирование показало, что наилучшие результаты были получены от оптимизации массивов и UDT. Наихудшие результаты были получены от Разбросанной организации тегов. Так как типичные приложения включают сочетание различных массивов, UDT, и т.д., то их производительность будет показывать средние результаты между наилучшими и наихудшими.

Пропускная способность в зависимости от интервала перезапуска (Periodic Rate) периодической задачи

Испытательная установка:

- Одна периодическая задача, для выполнения которой в Logix5555 требуется 63 мс.
- 5000 тегов типа DINT были собраны с помощью OPC test client. Частота обновления = 1мс
- Модуль ENBT в коммутируемой сети, имеющей скорость 100 Мбайт/с. Драйвер Ethernet в RSLinx Classic. Единственным другим узлом в сети являлся компьютер. DELL Optiplex GX260. 1.8Ghz
- RSLinx 2.40.01
- ControlLogix версии 10.24
- Движение или ввод/вывод не выполнялся

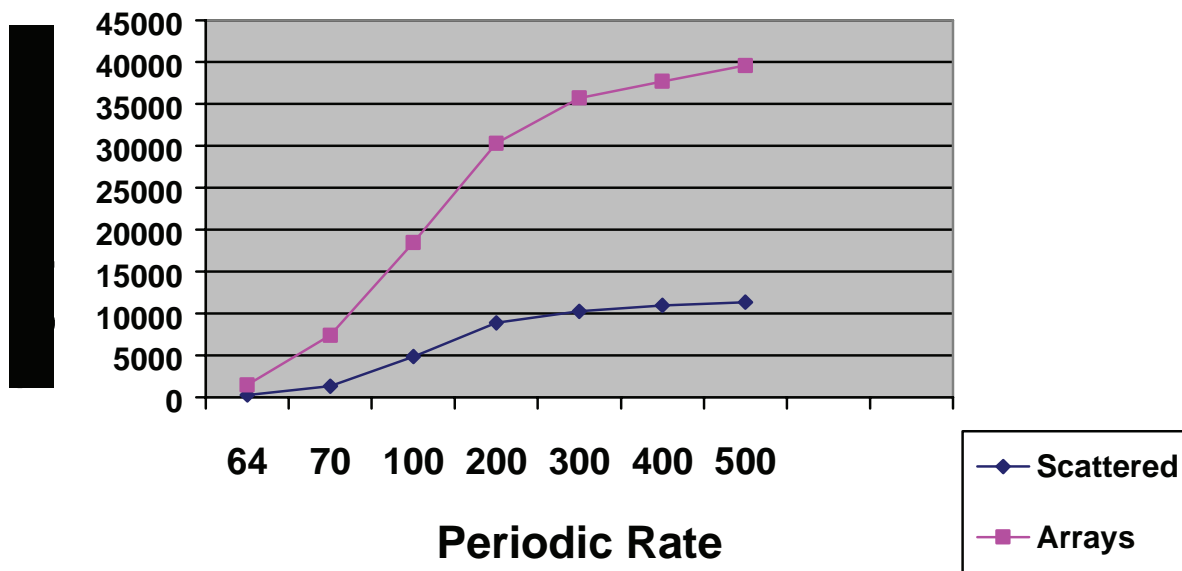
Тестирование выполнялось для двух различных организаций тегов. Теги были разбросаны или оптимизированы в Массивах. Результаты UDT здесь не приводятся, так как они соответствуют результатам оптимизации массивов.

Теги в секунду = (5000 элементов/количество оптимизированных пакетов ControlLogix (из RSLinx Classic Server Diagnostics)) * Количество пакетов в секунду (из RSLinx Classic Driver Diagnostics).

В данном тестировании для выполнения периодической задачи потребовалось 63 мс. Чтобы повысить производительность, необходимо было увеличить интервал перезапуска (Periodic Rate).

Интервал перезапуска для периодической задачи (мс)	Элементы в секунду Отсутствие массивов или UDT	Элементы в секунду Массивы
64	260	1450
70	1300	7400
100	4840	18450
200	8890	30300
300	10250	35700
400	10950	37700
500	11350	39600

Performance vs. Periodic Rate



Результаты тестирования:

Пропускная способность в тегах в секунду зависит от организации тегов и от интервала перезапуска (*Periodic Rate*). Тестирование показало, что наилучшие результаты были получены от оптимизации массивов и UDT. Наихудшие результаты были получены от Разбросанной организации тегов. Так как типичные приложения включают сочетание различных массивов, UDT, и т.д., то их производительность будет показывать средние результаты между наихудшими и наилучшими.

Пропускная способность в зависимости от Системных Издержек и интервала перезапуска для непрерывной и периодической задачи

В тестировании были использованы непрерывная и периодическая задачи. Для выполнения периодической задачи потребовалось 63мс.

Системные Издержки	Интервал перезапуска для периодической задачи	Максимальное время выполнения непрерывной задачи (мс)	Разбросанные теги в секунду
10%	70	696	140
50%	70	1,190	610
90%	70	6,376	1150
10%	100	196	480
50%	100	388	1990
90%	100	1,801	4230
10%	200	134	860
50%	200	187	2340
90%	200	795	6600
10%	400	134	1070
50%	400	192	1960
90%	400	914	7660

Результаты:

При сочетании периодической и непрерывной задач полоса пропускания связи еще больше уменьшается. Выделение полосы пропускания связи подразумевает нахождение баланса между Интервалом Перезапуска (*Periodic Rate*) и Системными Издержками (*Time Slice*). Увеличение параметра *Time Slice* без увеличения *Periodic Rate* может значительно увеличить время выполнения Непрерывной задачи, но при этом использование ресурсов связи почти не сократится.

Производительность в зависимости от типа процессора

Испытательная установка:

- Одна непрерывная задача, которая выполняется в 1756-5555
- 5000 Разбросанных тегов типа DINT были собраны с помощью OPC test client. Частота обновления = 1мс
- Модуль ENBT в коммутируемой сети, имеющей скорость 100 Мбайт/с. Драйвер Ethernet в RSLinx Classic. Единственным другим узлом в сети являлся компьютер. DELL Optiplex GX260. 1.8Ghz
- RSLinx 2.40.01
- ControlLogix версии 10.xx
- Движение или ввод/вывод не выполнялся

Системные Издержки	Теги в секунду (DINT) 5550	Теги в секунду (DINT) 5555	Теги в секунду (DINT) 5563
10%	970	1270	4670
20%	1970	2675	9540
30%	2980	4070	14230
40%	3920	5340	19300
50%	4840	6620	20500
60%	5850	7940	21050
70%	6700	9190	21050
80%	7570	10360	21050
90%	8410	11532	21050

Performance vs. Processor Type

