

# **КОМПЛЕКС ПРОГРАММ «ЗОНД»**

## **УСО ОРСМ**

Руководство пользователя

Версия 4.40.0342

**Москва, 2016**

## СО Д Е Р Ж А Н И Е

1. Кратко о технологии ОРС .....	4
2. Реализация интерфейса ОРСМ .....	6
3. Алгоритм задачи опроса .....	8
4. Панель инженера и окно протокола .....	11
5. Параметры БД УСО ОРСМ .....	13
6. Список используемых документов .....	15

***Как связаться с разработчиками?***

тел. \ факс.       (495) 382-56-34  
газовая связь:   тел. (700) 52-490, 52-495  
e-mail:           [zond@gpa.ru](mailto:zond@gpa.ru)  
Web:              [www.gpa.ru/zond](http://www.gpa.ru/zond)

## 1. Кратко о технологии OPC

Технология OPC (OLE for Process Control) предназначена для импорта технологических данных из приложения OPC сервера в приложение OPC клиент и воздействия на технологический процесс в обратном направлении. Технология определяется несколькими спецификациями (работа с текущими значениями, событиями, историческими данными), стандартизированными консорциумом OPC Foundation.

OPC – это надстройка над COM/DCOM (Distributed Component Object Model, технология Windows). Ключевое понятие DCOM – интерфейс, описание набора методов (функций интерфейса), на зависящее от платформы, имеющее только алгоритмические рекомендации к реализации. DCOM требует и предоставляет разработчику средства для обеспечения уникальности интерфейсов.

Взаимодействие по текущим значениям определяется спецификацией OPC Data Access 2.0 [2]. В рамках спецификации приложения (различных производителей) OPC серверов и OPC клиентов обязаны содержать реализации DCOM-интерфейсов, требуемый набор и описание которых зафиксированы стандартом OPC. При четкой реализации и следовании рекомендациям спецификации OPC получается предсказуемое поведение приложений, что позволяет осуществлять гибкий межпроцессный обмен технологическими данными и управление между двумя приложениями – клиентом и сервером. Приложения могут находиться на одном или разных компьютерах (хостах), маршрутизация вызовов и перенаправление данных параметров реализуется DCOM. DCOM действует только в локальной сети с протоколом TCP IP.

В общем случае приложения сервера и клиента многопоточные. DCOM связывает соответствующие друг другу потоки сервера и клиента, продолжая их в теле разных приложений. Взаимодействие сервера и клиента осуществляется прямыми вызовами (поток идет по пути клиент-DCOM-сервер-DCOM-клиент) и обратными вызовами (поток идет по пути сервер-DCOM-клиент-DCOM-сервер). Вызовы методов интерфейсов обозначаются {Интерфейс}::{Метод}.

Спецификация OPC определяет два вида объектов для реализации на стороне сервера – сервер и группа – и задает обязательный к реализации набор их интерфейсов. Объекты – не только как сущности взаимодействия, но и как внутрипрограммные компоненты, так как COM по определению оперирует объектами (см. термин объектно-ориентированное программирование). Для доступа к данным клиент выстраивает на стороне сервера конструкцию взаимосвязанных объектов – серверов, наполненных группами. Количество объектов серверов и групп в каждом не ограничено. Алгоритмика клиента (распределение данных по группам, временные параметры взаимодействия) определяется разработчиком клиента.

Классы при разработке серверов получают имя и уникальный идентификатор CLSID. Фирма Microsoft предоставляет утилиту для генерации новых уникальных идентификаторов (для всех ОС Windows во всем мире). На разработчика сервера DCOM накладывается требование дать ему CLSID и обеспечить механизм DCOM-регистрации сервера. После регистрации Windows имеет механизм разрешения имён классов в их CLSID. Пользователи чаще оперируют именами классов для идентификации серверов, а программы преобразуют их в CLSID.

Для корректного обращения клиента к серверу на другом хосте нужно, чтобы выполнялись следующие условия:

- \* Хосты в локальной сети, не в домене;
- \* Учётные записи, от которых запускаются приложения, и их группы на хостах совпадают;

- \* На хосте клиента сервер зарегистрирован;
- \* На обоих хостах в настройках DCOM (**dcomcnfg.exe**) установлены параметры: удостоверение – текущий пользователь (Interactive user), разрешение на локальный и удалённый доступ (и активацию в XP SP2), запрет локального и удалённого запуска для группы.

Изначально данные сервера представлены клиенту как древовидная структура. Навигация по дереву обеспечивается методами интерфейса IOPCBrowseServerAddressSpace. Существуют приложения просмотрщиков тегов, позволяющие осуществлять навигацию и запоминать пути в дереве нужных элементов. Интерфейс IOPCBrowseServerAddressSpace не является обязательным к реализации в рамках спецификации OPC Data Access 2.0.

Элементом групп являются теги (Items). Теги имеют уникальные текстовые имена, определяющие их местоположение в дереве сервера. При добавлении тега в группу клиентом указывается имя. Для успешно добавленного тега возможно обновление текущего значения (обратными вызовами со стороны сервера по изменениям, метод IOPCDataCallback::OnDataChange()), чтение и запись значений со стороны клиента. Теги и группы могут быть в активном и неактивном состоянии. Обратные вызовы идут только по активным тегам активных групп.

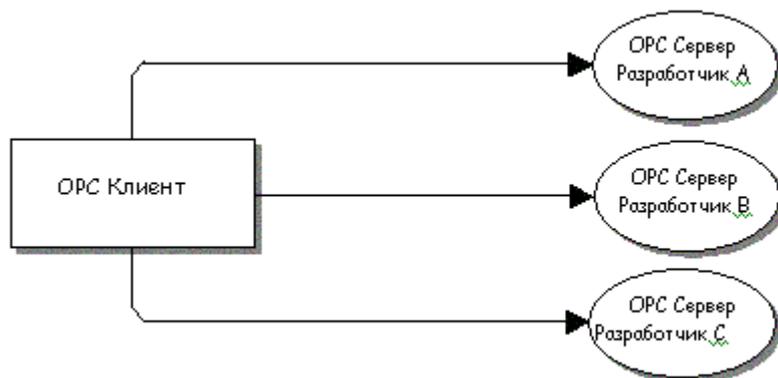
При добавлении клиентом тега в группу, он задаёт параметр “запрашиваемый тип” (vtRequestedType, наиболее приемлемый формат значений для клиента). Сервер может добавить или не добавить тег. Если тег добавлен, сервер в ответ заполняет параметр “канонический тип” (vtCanonicalType, наиболее приемлемый формат значений для сервера), при этом значения обратного вызова будут идти запрашиваемого типа, приведённые сервером. Если при добавлении указать пустой тип (VT\_EMPTY), то значения обратного вызова будут идти канонического типа, данные затем приводятся клиентом к внутренним типам. Очевидно, что наиболее благоприятная ситуация, когда запрашиваемый и канонический тип совпадают.

Порция информации по одному тегу в обратном вызове включает текущее значение (типы определены в DCOM), его достоверность (типы определены в DCOM) и штамп времени (когда значение получено на сервере).

Со стороны клиента возможны вызовы синхронных и асинхронных методов Write() для тегов группы. Значения методов имеют описанную выше структуру. Синхронный метод IOPCSyncIO::Write() выполняется в одном потоке, делающем прямой вызов. Асинхронный метод предполагает две стадии: Первая, прямой вызов IOPCAsyncIO2::Write(), возвращается, инициируя на стороне сервера некий процесс, который перед своим окончанием делает обратный вызов IOPCDataCallback::OnWriteComplete().

Существуют методы IOPCAsyncIO2::Refresh(), обязывающие сервер безусловно сделать обратный вызов IOPCDataCallback::OnDataChange() по всем элементам группы, обновив текущие значения.

Существует штатный механизм уведомления сервером клиента о желании остановить работу (метод IOPCShutdown::ShutdownRequest()). Клиент обязан при его получении демонтировать свои построения. Сервер, почувствовав демонтаж, прекращает работу.



**Рис. 1-1. Принцип работы УСО OPCM**

## 2. Реализация интерфейса OPCM

УСО OPCM (OPC мастер) использует спецификацию OPC Data Access 2.0.

УСО OPCM реализовано только в SCADA-модулях Зонд платформы Windows [1].

Возможно запустить одновременно до 16 задач, каждая из которых может взаимодействовать с одним OPC сервером.

Параметры, определяющие работу линии, хранятся в файле **uso\_conf\opcм.cfg** директории БД.

Имя тега рассматривается как часть подключения паспорта параметра БД. Имена тегов БД Зонд хранятся в файле **zond.opc** директории БД.

При добавлении в группу клиент указывает запрашиваемый тип.

Соответствие (жёсткое) типов данных DCOM-OPC и ПК Зонд приведено (Таблица 1).

**Таблица 1. Соответствие типов OPC и ПО Зонд**

Типы данных ПК Зонд	Запрашиваемый тип	Совместимые типы	Трансляция в значение Зонд
Аналоговый	VT_R4	VT_R8	Присваивание
Дискретный однобитный	VT_BOOL	VT_I1, VT_UI1, VT_I2, VT_UI2,	0 – 0, не 0 – 1
Дискретный двухбитный, восьмипозиционный,	VT_UI4	VT_I4, VT_UI4, VT_INT, VT_UINT	Присваивание с контролем превышения значения
Внешний счётчик, Внешний таймер	VT_UI4	VT_I4, VT_UI4, VT_INT, VT_UINT	Присваивание
Дата-время	VT_DATE или VT_UI4 ***	VT_I4, VT_UI4, VT_INT, VT_UINT	***

\*\*\* в зависимости от значения параметра задачи “параметр DATE как time\_t” – нет (по умолчанию) соответствует VT\_DATE, да - VT\_UI4. Указанием запрашиваемого типа инициируется определённый тип значения обратного вызова. Значение в обратном вызове типа VT\_DATE обрабатывается как время UTC (согласно DCOM), значение в обратном вызове типа VT\_UI4 или совместимых обрабатывается как время в формате time\_t (локальное время, число секунд с 01.01.1970), независимо от значения параметра.

Понятие совместимого типа – если после указания запрашиваемого типа обратный вызов все же пойдёт с другим значением (такова реализация сервера), значения части типов (совместимых) будет обработана корректно.

Соответствие достоверности данных DCOM-ОРС и ПК Зонд приведено (Таблица 2).  
Штамп времени не обрабатывается.

**Таблица 2. Соответствие достоверности ОРС и ПО Зонд**

Атрибут КАЧЕСТВО (Quality) данных ОРС	Интерпретация по ОРС	Достоверность данных ПК Зонд
good	Достоверно	Достоверно
good, local override	Достоверно, замещающее или установленное вручную значение	Достоверно
Bad	Дефектное, причина неизвестна	Недостоверно
Bad, config error	Дефектное, имеются некоторые зависящие от сервера проблемы с конфигурацией. Например, запрашиваемый элемент был уже удален из конфигурации	Недостоверно
Bad, no connected	Дефектное. Требуется, чтобы устройство ввода было Нет соединений в сегментах устройства-поставщика данных.	Недостоверно
Bad, device failure	Дефектное, отказ устройства	Недостоверно
Bad, sensor failure	Дефектное, отказ датчика	Недостоверно
Bad, last known	Дефектное, отказ коммуникаций. Доступно последнее известное значение. Значение кэша не обновлялось	Недостоверно
Bad, comm failure	Дефектное. Отказ коммуникаций. Последнего известного значения нет в наличии.	Недостоверно
Bad, out of service	Дефектное. Просмотр блока выключен или как-то заблокирован. Это качество используется также для неактивного состояние элемента или группы.	Недостоверно
Uncertain	Не определено, причина неизвестна	Недостоверно
Uncertain, last usable	Не определено. Значение прекратило обновление. Возвращаемое значение должно рассматриваться, как устаревшее. Состояние связано конкретно с обнаруженной ошибкой коммуникации на 'выбранном' значении. Эта ошибка связана с отказом некоторых внешних источников 'внести' изменение в значение в пределах допустимого периода времени.	Недостоверно
Uncertain, sensor cal	Не определено. Либо значение 'зафиксировалось' на одном из пределов датчика (в таком случае поле предела должно быть установлено в 1 или 2), либо с помощью какой-то формы внутренней диагностики стало известно, что у датчика сбилась калибровка (в таком случае поле предела	Недостоверно

Атрибут КАЧЕСТВО (Quality) данных OPC	Интерпретация по OPC	Достоверность данных ПК Зонд
	должно быть 0).	
Uncertain, egu exceeded	Не определено. Возвращаемое значение находится за пределами, определенными для этого параметра.	Недостоверно
Uncertain, sub normal	Не определено. Значение получено из множества источников, число Хороших источников меньше, чем требуется	Недостоверно

### 3. Алгоритм задачи опроса

Задача может быть запущена в работу автоматически при старте модуля Зонд или вручную.

Алгоритм работы задачи следующий:

1. Создается объект клиента, реализующий интерфейсы IOPCShutdown, IOPCDataCallback.
2. Выполняется следующий цикл:

Если связь есть –

Выполняем процедуру операций цикла

Если в процедуре операций цикла ошибка –  
выполняем процедуру разрыва связи

Пауза

Иначе (связи нет)

Выполняем процедуру установки связи

Если не получилось

Выполняем процедуру разрыва связи

Пауза

Процедура установки связи имеет следующий алгоритм:

1. Разрешается имя класса (вызов COM CLSIDFromProgID)
2. Создаётся объект сервер на стороне OPC сервера (вызов COM CoCreateInstance)
3. Регистрируется точка обратного вызова для разрыва связи, иницируемого сервером (интерфейс IOPCShutdown)
4. Создаётся группа сервера в неактивном состоянии (метод IOPCServer::AddGroup).
5. В группу добавляются элементы (метод IOPCGroup::AddItems) согласно числу параметров БД, найденных в БД, имеющих подключение OPCM и соответствующий номер задачи.
6. Регистрируется точка обратного вызова для уведомления об изменениях значений тегов (интерфейс IOPCDataCallback)
7. Запрещаются обратные вызовы для группы (кроме метода опроса по изменениям, метод IOPCASyncIO2::SetEnable).
8. Группа переводится в активное состояние (метод IOPCGroupStateMgt::SetState). С этого момента начинают поступать обратные вызовы (для метода опроса по изменениям).

Процедура разрыва связи имеет алгоритм, демонтирующий построения процедуры установки связи в обратном порядке до исходного состояния с обеих сторон.

Процедура операций цикла имеет следующий алгоритм:

Вызов метода `GetStatus()`

Если успешно

    вызов явного метода чтения (при опросе по изменениям этого не делается)

    Если успешно

        Пауза с проверкой останова задачи опроса OPCM и  
        получения сигнала `Shutdown` от сервера.

Вызов метода `IOPCServer::GetStatus()` проходит корректно, если среда DCOM, в которой до того было успешно установлено соединение, продолжает оставаться работоспособной. Если, например, при OPC взаимодействии между хостами связь разорвана на сетевом уровне или приложение сервера неожиданно ушло в исключение, то метод вернет код ошибки.

Во время паузы анализируется прохождение вызова `IOPCShutdown::ShutdownRequest()` со стороны сервера. Если он был замечен, процедура заканчивается.

Задача опроса использует один из четырех методов опроса – по изменениям, синхронный, асинхронный и обновление. Ниже приведены используемые методы спецификации OPC, состояния ключевых параметров, поведение клиента OPCM и ожидаемое согласно спецификации поведение сервера (Таблица 3).

Термины КЭШ (CACHE) и УСТРОЙСТВО (DEVICE), по спецификации, предполагают, что большинство серверов будет читать данные в некоторого рода КЭШ. Также большинство клиентов будет читать данные из этого кэша при помощи определенных механизмов. Предполагается, что доступ к данным УСТРОЙСТВА будет ‘медленным’ и будет использоваться, главным образом, для диагностики или для особо критических операций.

Таблица 3. Методы опроса

Метод опроса, используемые методы и интерфейсы OPC	Поведение клиента	Поведение сервера согласно спецификации
По изменениям IOPCDataCallback::OnDataChange	“Опрос из” - не важно, обратные вызовы разрешены, состояние группы и элементов активное	Значение и Качество – значения, которые сервер получает от устройства с периодичностью, достаточной для того, чтобы обеспечить заданную UpdateRate. Если Качество изменилось с момента последней передачи Качества клиенту, то новое значение и новое качество будут переданы клиенту методом IOPCDataCallback::OnDataChange, а кэш сервера должен быть обновлен полученными значением и качеством. Если Качество не изменилось с последней передачи клиенту, сервер должен сличить полученное значение для изменения, превышающего критерий мертвого времени. Если изменение в значении превышает критерий мертвого времени, тогда новое значение и новое качество будут посланы клиенту методом IOPCDataCallback::OnDataChange, а кэш сервера должен быть обновлен полученными значением и качеством.
Синхронное чтение IOPCSyncIO::Read	“Опрос из” КЭШ, обратные вызовы запрещены, состояние группы и элементов активное	Значения и Качество для запрошенных элементов возвращаются клиенту как возвращаемые значения метода. Значение и Качество – значения, которые сервер держит в кэше
Синхронное чтение IOPCSyncIO::Read	“Опрос из” УСТРОЙСТВА, обратные вызовы запрещены, состояние группы и элементов активное (не важно)	Значения и Качество для запрошенных элементов возвращаются клиенту как возвращаемые значения метода. Значение и Качество – значения, которые сервер получает из устройства при вызове этого метода. Кэш сервера должен быть дополнен полученными значением и качеством.
Асинхронное чтение IOPCAsyncIO2::Read	“Опрос из” УСТРОЙСТВА, обратные вызовы запрещены, состояние группы и элементов активное (не важно)	Значения и Качество для запрошенных элементов передаются клиенту методом IOPCDataCallback::OnReadComplete. Значение и Качество – значения, которые сервер получает из УСТРОЙСТВА при вызове этого метода. Кэш сервера должен быть дополнен полученными значением и качеством.
Обновление IOPCAsyncIO2::Refresh	“Опрос из” КЭШ, обратные вызовы запрещены, состояние группы и элементов активное	Значения и Качество для всех Активных элементов в группе передаются клиенту методом IOPCDataCallback::OnDataChange. Значение и Качество – значения, которые сервер держит в кэше.

<p>Обновление IOPCAsyncIO2::Refresh</p>	<p>“Опрос из” УСТРОЙСТВА, обратные вызовы запрещены, состояние группы и элементов активное</p>	<p>Значения и Качество для всех элементов в группе передаются клиенту методом IOPCDataCallback::OnChange. Значение и Качество – значения, которые сервер получает из устройства при вызове этого метода. Кэш сервера должен быть дополнен полученными значениями и качеством</p>
---	--	--

#### 4. Панель инженера и окно протокола

Панель инженера (см. Рис. 4-1) предназначена для настройки параметров интерфейса и контроля за работой задачи УСО OPCM.

Панель состоит из двух частей:

- дерева параметров конфигурации задачи;
- список параметров БД, имеющих подключение к текущей линии УСО OPCM.

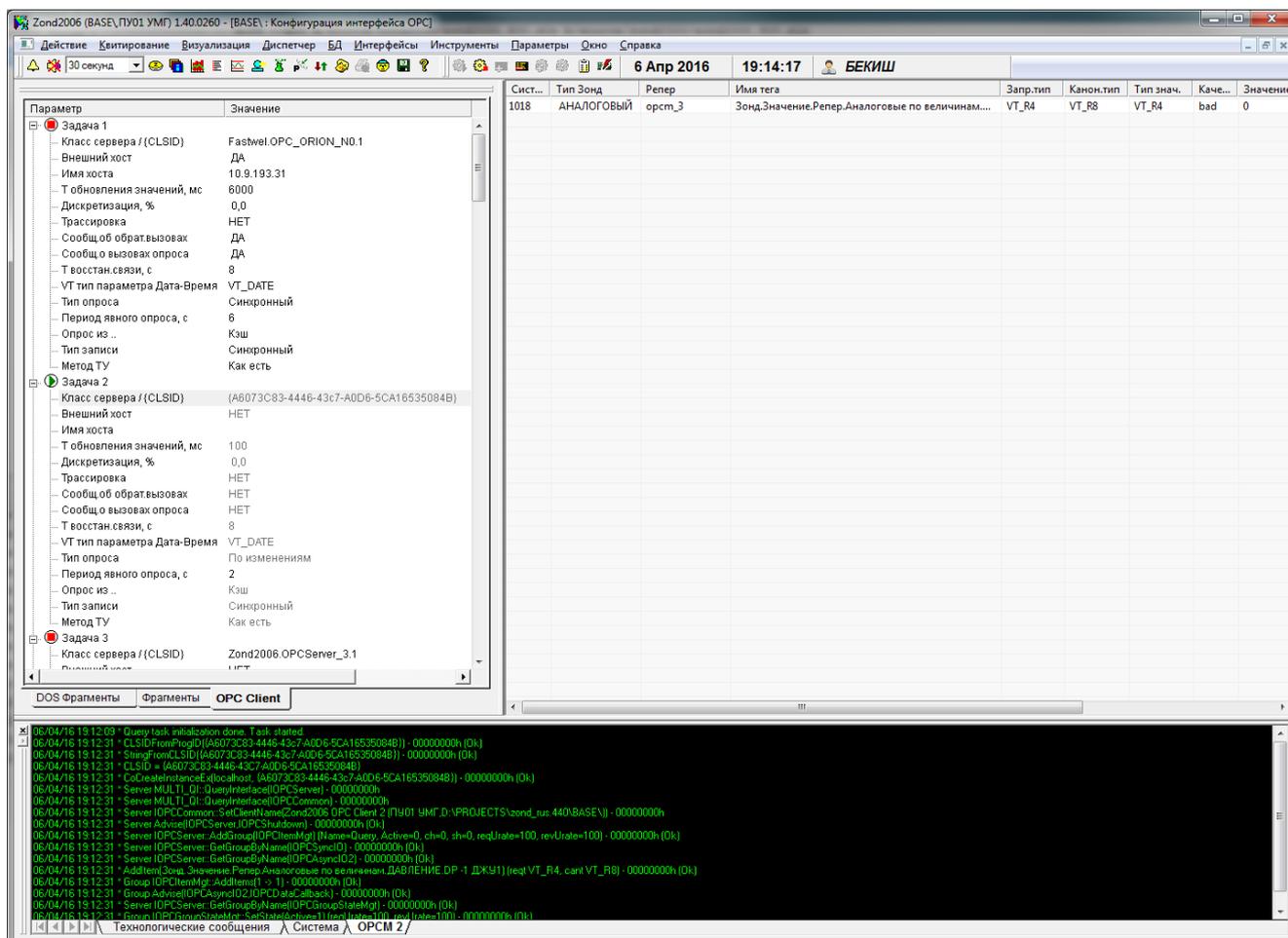


Рис. 4-1. Панель УСО OPCM

Сообщения об ошибках, возникающих во время работы задачи, выводятся в окно системных сообщений (закладка “Система”).

Таблица содержит следующие столбцы (Таблица 4):

**Таблица 4. Столбцы таблицы панели инженера OPCM**

Параметр	Значение
Системный номер	Системный номер параметра БД
Тип Зонд	Тип параметра БД ПК Зонд
Репер	Репер параметра БД.
Имя тега	Имя тега
Запр.тип	Запрашиваемый тип (vtRequestedType)
Канон.тип	Канонический тип (vtCanonicalType)
Тип знач.	Тип значения
Качество	Качество значения
Значение	Значение в формате VT_TYPE (используется в OPC)
Физика	Текущее значение параметра БД
Штамп времени	Штамп времени значения в обратном вызове

При нажатии правой кнопки мыши на строке таблицы выводится контекстное меню с возможными действиями для соответствующего параметра БД.

На закладке «OPCMx» (x – номер задачи) выводится окно протокола, записи которого описывают вызовы методов OPC.

Также возможна трассировка протокола в файл (в директории uso\_trace директории БД), который затем можно просмотреть в специальном редакторе Зонд (Главное меню – инструменты – просмотр трассировки обменов)

Параметры конфигурации OPCM (Таблица 5):

**Таблица 5. Параметры интерфейса OPCM**

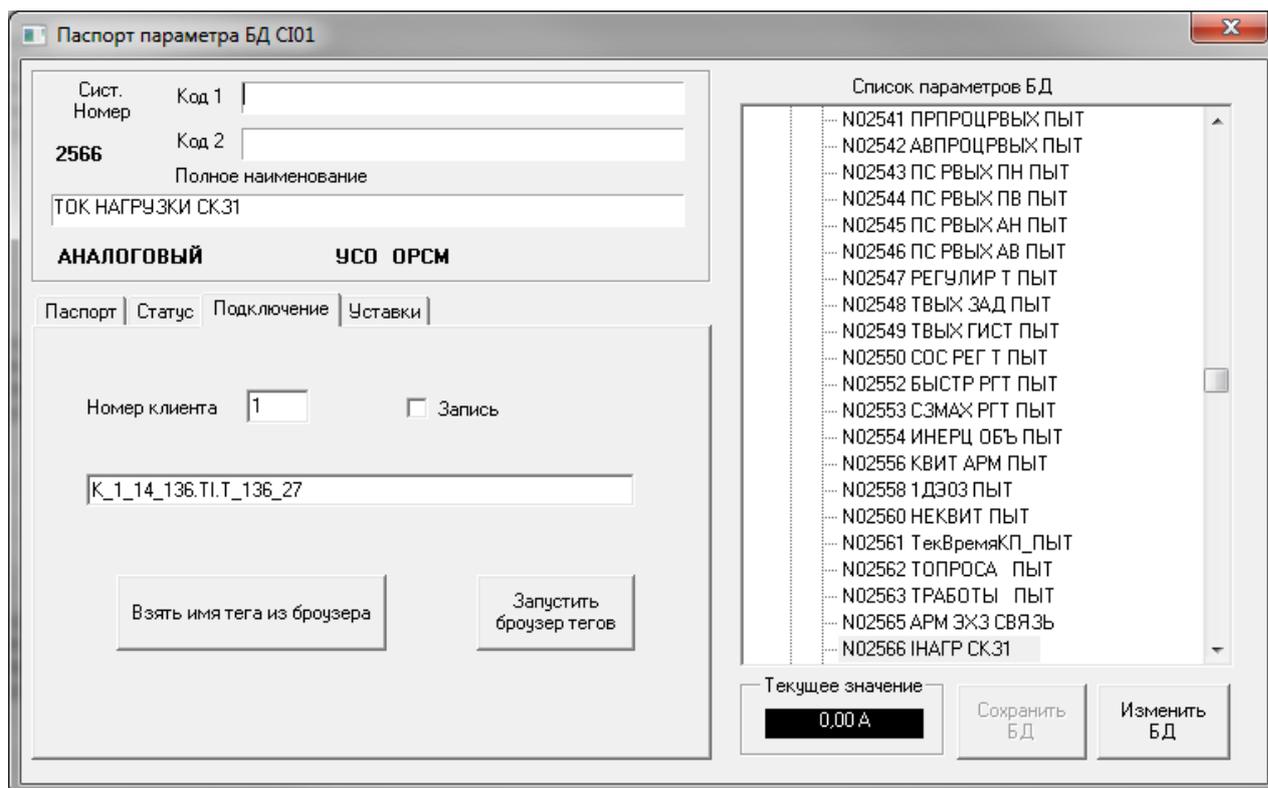
Параметр	Смысл
Класс сервера / {CLSID}	Имя класса сервера, с которым взаимодействует задача или его идентификатор в формате {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}. Внимание, идентификатор должен начинаться с фигурной скобки '{'.
Внешний хост	Да – обращаться к серверу на другой компьютер, Нет – обращаться к серверу на тот компьютер, где работает клиент (Зонд)
Имя хоста	При установке «внешний хост-да» запрос на создание объекта сервер выдаётся на указанный IP адрес. Можно указать UNC или DNS имя. Например: 124.196.8.1; krupki; krupki.btg.com
Т обновления значений, с.	Время обновления данных – параметр группы, который передается ей при создании. Определяет быстроту реакции сервера на изменения. 0 – быстрота определяется сервером.
Дискретизация, %.	Параметр квантования диапазона аналоговых сигналов сервера по уровню. При пересечении значением границы кванта делается обратный вызов по изменению значения. Может использоваться для уменьшения количества обратных вызовов за счет закругления данных. 0 – не использовать этот механизм.
Трассировка	Да – включить трассировку в файл
Сообщ. об обрат. вызовах	Да – делать сообщения об обратных вызовах в окно сообщений и в файл трассировки, если она включена.

Сообщ. о вызовах опроса	Да – делать сообщения о вызовах циклического опроса в окно сообщений и в файл трассировки, если она включена.
Т восстан. связи, с	
VT тип параметра Дата-Время	Да – для параметров типа Дата-Время пытаться получить данные обратного вызова типа VT_UI4 и интерпретировать их как время в формате time_t
Тип опроса	По изменениям, Синхронный, Асинхронный, Обновление (Таблица 3)
Период явного опроса, с.	Период паузы в цикле опроса алгоритма задачи клиента (см. п.3) Для типа опроса “по изменениям” с таким периодом проверяется только связь с сервером, для остальных – еще и опрос одним из методов
Опрос из ..	Возможные значения: КЭШ, УСТРОЙСТВО . Параметр востребован для типов опроса Синхронный и Обновление (Таблица 3)
Тип записи	“Синхронный” или “Асинхронный”
Метод ТУ	”Как есть” – устанавливается значение из БД зонд, без преобразования; ”Зонд простой” – для двухбитных типа «кран» к значению в БД добавляется 20, для всех остальных 30 (как принято в Зонд OPC-server). ”Зонд стадии” – не реализован.

## 5. Параметры БД УСО ОРСМ

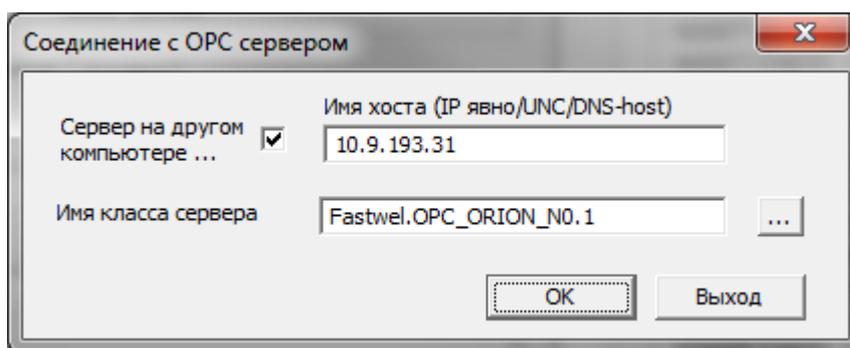
В подключении параметра в БД указывается (Рис. 5-1):

Поле подключения	Комментарий
Номер клиента	Номер задачи клиента 1...16
Запись	Признак управления, параметр управляем
Имя тега	Имя тега, начиная от корня дерева сервера



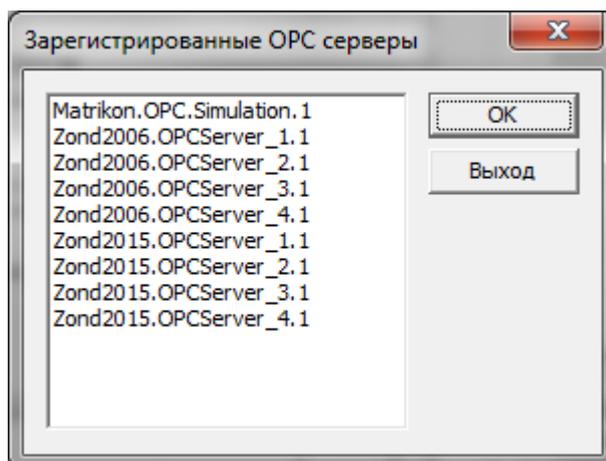
**Рис. 5-1. Подключение параметра БД**

Имя тега можно задать явно или выбрать из браузера тегов. Во втором случае при нажатии кнопки “Запустить браузер тегов” появится окно, позволяющее задать местонахождение и имя класса сервера. По умолчанию будут взяты значения из параметров задачи клиента, указанной на закладке подключения, но их можно исправить. При указании сервера другого компьютера нужно указать IP адрес этого компьютера явно или в форме UNC или DNS имени, которое должно разрешаться в адрес механизмами локальной сети.



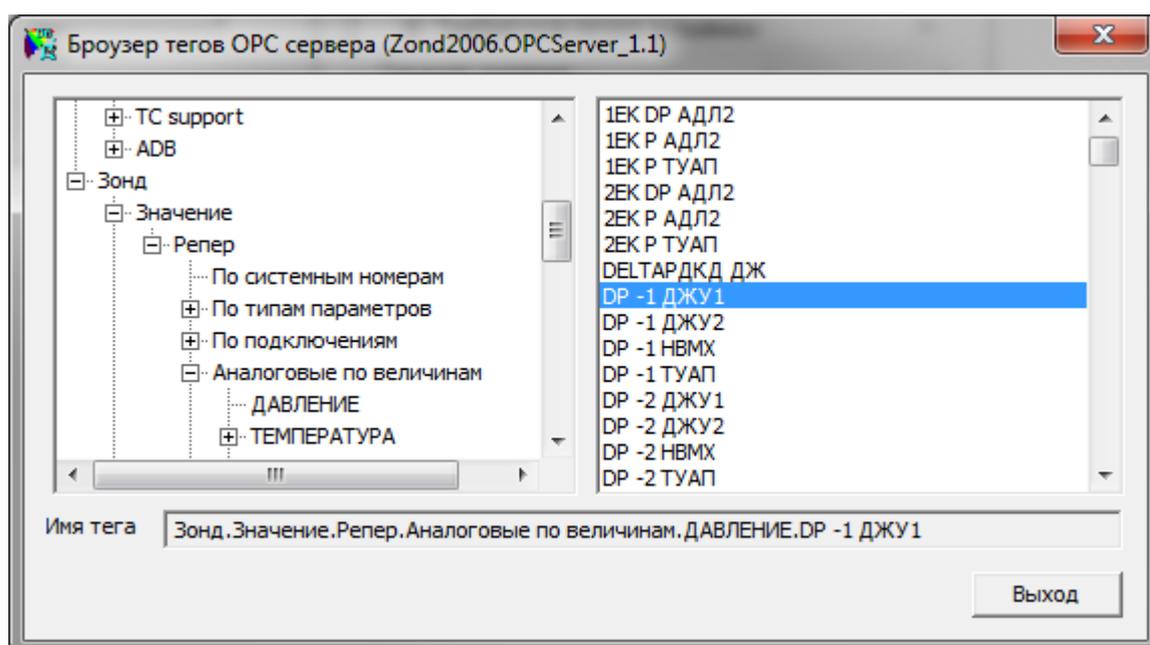
**Рис. 5-2. Диалог настройки браузера тегов**

При нажатии кнопки “...” будет предпринята попытка сформировать список зарегистрированных в системе локального или удалённого компьютера серверов OPC Data Access 2.0. Из списка можно сделать выбор имени класса.



*Рис. 5-3. Диалог выбора OPC сервера*

При нажатии кнопки “ОК” браузер попытается соединиться с сервером и отобразить его дерево. Далее возможна навигация по дереву. При установке маркера на элемент дерева в нижней строке отобразится имя тега. Нажатием кнопки паспорта “Взять имя тега из браузера” его можно скопировать в поле паспорта.



*Рис. 5-4. Браузер тегов*

## 6. Список используемых документов

Док. 1. Комплекс программ «Зонд». Zond2006. Описание применения.

Док. 2. OPC Data Access Custom Interface Specification. Version 2.04. OPC Foundation., September 5, 2000