

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

СОГЛАСОВАНО
Генеральный директор
ЗАО «ИТИ»


Бакиев О.Р.
2011 г.

УТВЕРЖДАЮ
Ректор НИУ ИТМО



Васильев В.Н.
2011 г.

**МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE**

**ПРОГРАММНЫЙ КОМПОНЕНТ ИСПОЛНЕНИЯ WF
CLAVIRE/EXECUTOR**

ОПИСАНИЕ ПРОГРАММЫ

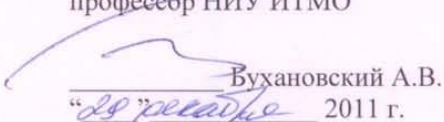
ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 13 29-ЛУ

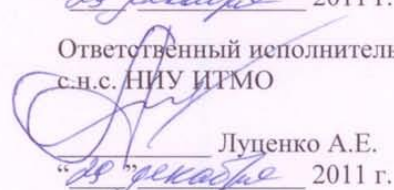
Име. № подл.	Подп. и дата	Взам. име. №	Име. № дубл.	Подп. и дата

Представители
Организации-разработчика

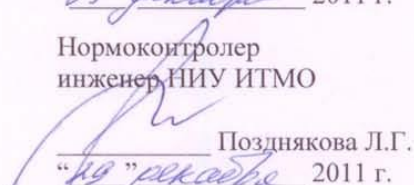
Руководитель разработки,
профессор НИУ ИТМО


Бухановский А.В.
"29" декабря 2011 г.

Ответственный исполнитель,
с.п.с. НИУ ИТМО


Луценко А.Е.
"29" декабря 2011 г.

Нормоконтролер
инженер НИУ ИТМО


Позднякова Л.Г.
"29" декабря 2011 г.

2011

УТВЕРЖДЕН

RU.СНАБ.80066-06 13 Ошибка! Источник ссылки не найден.-ЛУ

**МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE**

**ПРОГРАММНЫЙ КОМПОНЕНТ ИСПОЛНЕНИЯ WF
CLAVIRE/EXECUTOR**

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 ОШИБКА! ИСТОЧНИК ССЫЛКИ НЕ НАЙДЕН.

Инв.№ подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТОВ 17

2011

АННОТАЦИЯ

Документ содержит описание программного компонента исполнения WF CLAVIRE/Executor RU.СНАБ.80066-06 01 29, реализующего функционал по осуществлению запусков заданий на распределённой вычислительной инфраструктуре МИТП. Программный компонент исполнения WF разработан в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

СОДЕРЖАНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ	4
2.	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	4
3.	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	4
3.1.	Принципы функционирования исполнения WF	4
3.2.	Основные классы компонента исполнения WF	6
3.2.1.	Класс Task	7
3.2.2.	Класс TaskDescription	9
3.2.3.	Класс TaskFileDescription	10
3.2.4.	Класс TaskSchedule	10
4.	ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА	11
5.	ВЫЗОВ И ЗАГРУЗКА	11
6.	ВХОДНЫЕ ДАННЫЕ	13
7.	ВЫХОДНЫЕ ДАННЫЕ	14
	ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	16

1. ОБЩИЕ СВЕДЕНИЯ

Программный компонент исполнения WF CLAVIRE/Executor RU.СНАБ.80066-06 01 29 предназначен для обеспечения запуска и исполнения заданий в распределенном облаке сервисов на основе концепции iPSE, в рамках моделей (а) монопольного использования суперкомпьютера, (б) метакомпьютинга, (в) Грид (применительно к особенностям Грид ННС). Выбор задания из общей очереди и назначение на вычислительный ресурс осуществляется на основе расписания исполнения WF, полученного от компонента планирования исполнения WF (CLAVIRE/Scheduler). Программный компонент CLAVIRE/Executor разработан на языке C# платформы .Net версии 4.0 в виде веб-сервиса с применением технологии Windows Communication Foundation (WCF) на основе стандарта SOAP. Развертывание сервиса производится на базе веб-сервера Internet Information Services (IIS) версии не ниже 6.0.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программный компонент CLAVIRE/Executor предназначен для организации запуска расчетов на распределенных вычислительных ресурсах (отдельных кластерах, группах персональных компьютеров или ресурсах Грид-сети), что включает в себя передачу входных данных и параметров запуска, контроль процесса работы приложения, получение выходных данных с удаленного вычислителя.

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Принципы функционирования исполнения WF

Программный компонент CLAVIRE/Executor включает в себя модули контроля вычислительной инфраструктуры и управления исполнением заданий на вычислительных сервисах, входящих в ее состав. В интерфейс данного компонента входят базовые команды работы с элементами WF: определение заданий для запуска, соответствующих элементам WF; запуск и остановка выполнения заданий; получение информации о текущем состоянии заданий, включая идентификаторы файлов расчета в хранилище данных.

RU.СНАБ.80066-06 13 **Ошибка! Источник ссылки не найден.**

Архитектура компонента автоматизации удаленного исполнения заданий и его связь с прочими компонентами комплекса показана на рис. 3.1.

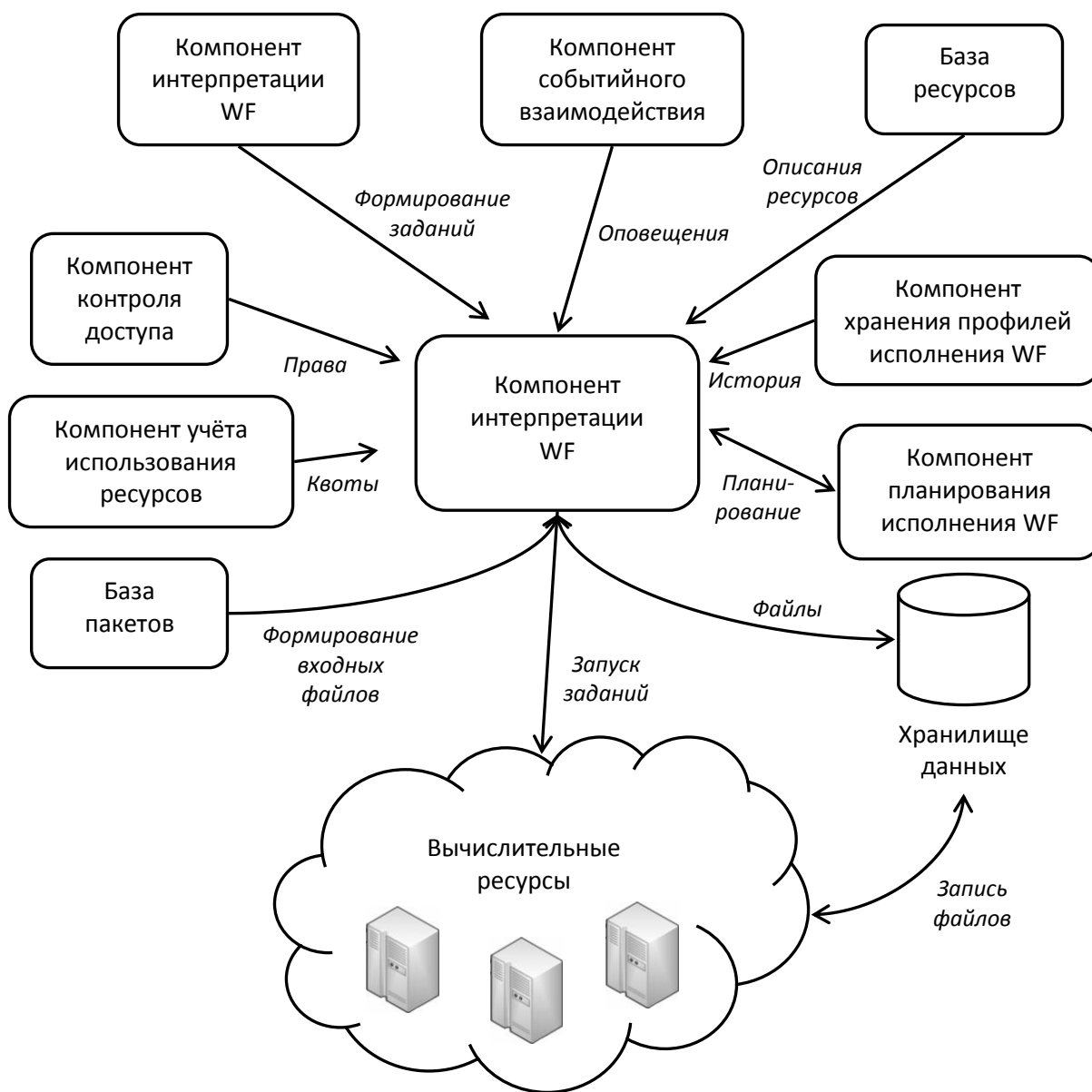


Рисунок 3.1 - Схема работы CLAVIRE/Executor и его связи с другими компонентами

В интерфейс компонента исполнения WF входят базовые команды работы с заданиями: описание заданий и связей между ними; запуск и остановка выполнения конкретных заданий; получение информации о текущем состоянии заданий, включая идентификаторы файлов расчета в хранилище данных. С каждым заданием ассоциируется множество входных файлов с именами и уникальными идентификаторами файла в

RU.СНАБ.80066-06 13 **Ошибка! Источник ссылки не найден.**

хранилище, а также множество параметров запуска. Все сформированные задания, не зависящие от других, помечаются как готовые к исполнению, а с заданий, получивших зависимость по данным, такая пометка снимается.

Каждый раз при добавлении нового задания или освобождении вычислительного ресурса (вследствие окончания расчета) компонентом исполнения WF инициируется процесс планирования, т. е. построения компонентом планирования WF отображения множества готовых к выполнению заданий на множество вычислительных ресурсов. Согласно полученному расписанию, компонентом исполнения WF осуществляется запуск заданий на соответствующих вычислительных ресурсах, контроль их исполнения, а также пред- и постобработка, выполняемая при помощи компонента базы пакетов.

Список ресурсов для построения расписания получается от компонента базы ресурсов. Затем, в случае наличия у пользователя на счету средств для запуска (проверяется путем запроса к компоненту учёта использования ресурсов), накладываются ограничения на возможность запуска задач на конкретных ресурсах, используя информацию, полученную от компонента контроля доступа, и состояние вычислительных ресурсов (наличие на них соответствующих пакетов и свободных узлов для запуска).

При успешном завершении одного из заданий компонентом исполнения WF определяются все зависимые от завершившегося шага элементы WF. Файлы расчета в соответствии с типами файлов найденных зависимостей забираются с вычислительного ресурса и записываются в хранилище. Все элементы WF, которые более не зависят от незавершившихся шагов, помечаются как готовые к исполнению.

При сбое одного из заданий, оно помечается готовым к новому запуску на ресурсах, отличных от тех, на которых ранее происходили сбои при выполнении данного элемента WF. Повторный запуск осуществляется в обычном порядке. При превышении количества неуспешных попыток запуска установленной критической величины, весь WF считается закончившимся с ошибкой.

3.2. Основные классы компонента исполнения WF

Структура основных классов, используемых в компоненте исполнения WF, показана на рисунке 3.2.

RU.СНАБ.80066-06 13 **Ошибка! Источник ссылки не найден.**

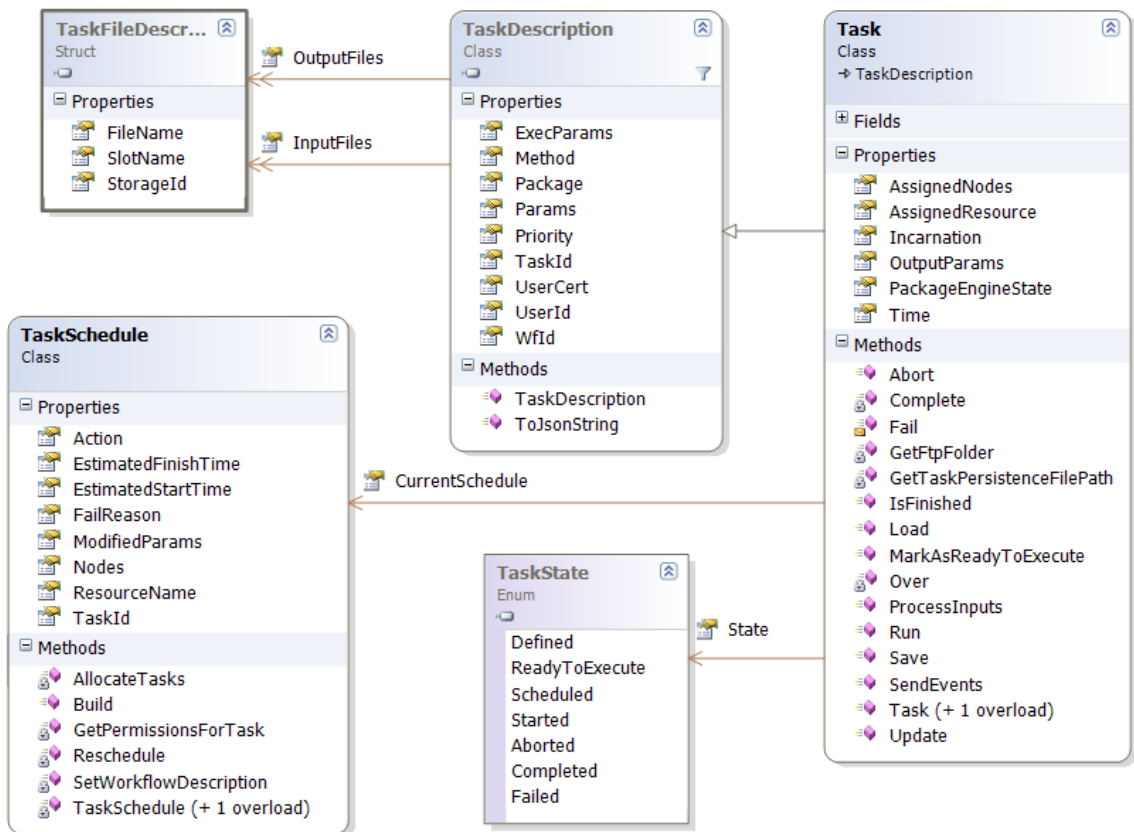


Рисунок 3.2. Структура основных классов компонента CLAVIRE/Executor

Ниже приводятся сокращенные описания структуры и методов основных классов компонента исполнения WF.

3.2.1. Класс Task

Центральный класс, описывающий поступившее на выполнение задание, который наследуется от класса TaskDescription.

Основные открытые поля:

- State (тип: TaskState) — текущее состояние задачи. Возможные значения: определена (Defined), готова к выполнению (ReadyToExecute), запланирована на выполнение (Scheduled), запущена на вычислительном ресурсе (Started), прервана (Aborted), завершена успешно (Completed), завершена неуспешно (Failed).
- AssignedResource (тип: ResourceTotals) — описание ресурса, на который запланировано задание.
- AssignedNodes (тип: IEnumerable<NodeTotals>) — описание вычислительных узлов, на которых запланировано задание.

RU.СНАБ.80066-06 13 Ошибка! Источник ссылки не найден.

- Incarnation (тип: IncarnationParams) — описание сформированных параметров, специфичных непосредственно для запуска на ресурсе (таких, как командная строка).
- OutputParams (тип: Dictionary<string, string>) — параметры, полученные компонентом базы пакетов после обработки выходных файлов задания.
- Time (тип: TaskTimeMeasurement) — составное поле учёта времени работы задания, оценок времени работы и измеренных накладных расходов.

Основные открытые методы:

- MarkAsReadyToExecute — пометить задачу, как готовую к запуску.
 - Входных параметров нет.
 - Возвращаемое значение отсутствует (void).
- ProcessInputs — использовать компонент базы пакетов для обработки и формирования входных данных.
 - Входных параметров нет.
 - Возвращаемое значение отсутствует (void).
- Run — запустить задачу на исполнение, используя указанное расписание.
 - Входные параметры: расписание (тип: TaskSchedule) и описание ресурсов (тип: IEnumerable<Resource>).
 - Возвращаемое значение отсутствует (void).
- Abort — принудительно завершить выполнение задачи.
 - Входные параметры: описание и текущее состояние вычислительных ресурсов (тип: IEnumerable<Resource>).
 - Возвращаемое значение отсутствует (void).
- Update — проверить и обновить состояние выполняющейся задачи. Если задача выполнена, то вызвать соответствующий метод (Complete или Fail) для перехода в новое состояние.
 - Входные параметры: описание и текущее состояние вычислительных ресурсов (тип: IEnumerable<Resource>).
 - Возвращаемое значение отсутствует (void).
- IsFinished — проверить, является ли состояние задачи конечным (Aborted, Completed, Failed).
 - Входных параметров нет.
 - Возвращается логическое значение, соответствующее результату проверки.

RU.СНАБ.80066-06 13 **Ошибка! Источник ссылки не найден.**

- **Complete** — завершить задачу, обработать выходные файлы базой пакетов и поместить их в хранилище данных.
 - Входной параметр: URL-адрес папки обмена (тип: string).
 - Возвращаемое значение отсутствует (void).
- **Fail** — завершить задачу и запомнить причину неудачного завершения.
 - Входной параметр: причина завершения задания (тип: string).
 - Возвращаемое значение отсутствует (void).
- **SendEvents** — отослать накопленные события, используя компонент событийного взаимодействия.
 - Входных параметров нет.
 - Возвращаемое значение отсутствует (void).

3.2.2. Класс *TaskDescription*

Класс, представляющий из себя исходное описание задачи, направляемой на выполнение компоненту CLAVIRE/Executor.

Основные открытые свойства:

- **WfId** (тип: string) — идентификатор WF, которому принадлежит запускаемая задача.
- **TaskId** (тип: ulong) — идентификатор задачи.
- **UserId** (тип: string) — идентификатор пользователя, запустившего задачу.
- **UserCert** (тип: string) — идентификатор сертификата пользователя в хранилище для запуска в Грид, если такой сертификат был передан, иначе — null.
- **Package** (тип: string) — идентификатор запускаемого пакета. Должен быть разрешённым именем в базе пакетов.
- **Method** (тип: string) — идентификатор запускаемого метода.
- **Priority** (тип: TaskPriority) — приоритет задачи. Поддерживаются два варианта: обычный и экстренный.
- **Params** (тип: IDictionary<string, string>) — параметры задачи, специфицированные пользователем или сформированные базой пакетов.
- **ExecParams** (тип: IDictionary<string, string>) — параметры задачи, относящиеся к специфике запуска, например, границы времени для экстренного WF.
- **InputFiles** (тип: TaskFileDescription[]) — описание входных файлов, переданных пользователем.

RU.СНАБ.80066-06 13 **Ошибка! Источник ссылки не найден.**

- OutputFiles (тип: TaskFileDescription[]) — описание выходных файлов, полученных или ожидаемых в результате расчёта.

3.2.3. Класс *TaskFileDescription*

Класс задаёт описание файлов задания.

Открытые свойства:

- FileName (тип: string) — имя переданного или сформированного файла.
- SlotName (тип: string) — идентификатор типа файла.
- StorageId (тип: string) — идентификатор файла в хранилище данных.

3.2.4. Класс *TaskSchedule*

Класс, описывающий расписание — результат планирования задачи.

Основные открытые свойства:

- TaskId (тип: ulong) — идентификатор задачи, к которой относится результат планирования.
- Action (тип: ScheduledAction) — решение, принятое планировщиком относительно запуска задачи. Может принимать следующие значения: запустить (Run), завершить без запуска (Fail), прекратить уже идущее выполнение (Abort), запустить в будущем (None).
- EstimatedFinishTime (тип: DateTime?) — оценка времени окончания задачи.
- EstimatedStartTime (тип: DateTime?) — оценка времени старта задачи.
- ResourceName (тип: string) — имя ресурса, на котором запланировано выполнение.
- Nodes (тип: IEnumerable<NodeConfig>) — описание конфигурации вычислительных узлов, на которых необходимо осуществить запуск (например, какие конкретно узлы ресурса задействовать, какое количество ядер на них использовать).

Основные открытые методы:

- Build — построить расписание выполнения множества заданий, используя компонент планирования WF на основе актуальной информации о состоянии ресурсов, полученной от базы ресурсов, с учётом имеющихся у пользователей прав и ограничений.
 - Входные параметры: список заданий для планирования (тип: IEnumerable<Task>), описание ресурсов МИТП (IEnumerable<Resource>

RU.СНАБ.80066-06 13 **Ошибка! Источник ссылки не найден.**

- resources), список зависимостей между заданиями (тип: IEnumerable<TaskDependency>).
- Возвращается расписание выполнения для переданных заданий (тип: IEnumerable<TaskSchedule>).
- **GetPermissionsForTask** — определить список ресурсов, на которых разрешён запуск выбранной задачи.
 - Входные данные: описание задачи (тип: Task), описание ресурсов МИТП (тип: IEnumerable<Resource>).
 - Возвращается список имён узлов ресурсов, на которых разрешён запуск указанной задаче (тип: IEnumerable<string>). Опционально выдаётся текст оповещения для пользователя, указывающий, по какой причине задача не может быть запущена.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Компонент CLAVIRE/Executor реализован в виде веб-сервиса на основе технологии WCF платформы Microsoft .Net версии 4.0. Для установки и запуска компонента требуется установленный веб-сервер IIS версии не ниже 6.0. Программный компонент поставляется в виде .Net сборок и предназначен для использования в рамках платформы .Net. Требования к аппаратной платформе диктуются требованиями реализации платформы .Net:

- x86-совместимый процессор с тактовой частотой 1 ГГц или выше;
- 512 МБ оперативной памяти или больше;
- минимальное место на диске: 850 МБ для x86 версии и 2 ГБ для x64 версии платформы .Net.

5. ВЫЗОВ И ЗАГРУЗКА

Компонент исполнения WF CLAVIRE/Executor реализован в виде SOAP WCF-сервиса платформы .Net. Для запуска сервиса используется стандартный механизм используемого web-сервера. Загрузка сервиса в этом случае выполняется web-сервером автоматически по мере поступления запросов от клиентов сервиса.

Вызов сервиса производится стандартным для технологии WCF способом: по опубликованному описанию сервиса (WSDL) создаётся прокси-класс, через который

RU.СНАБ.80066-06 13 **Ошибка! Источник ссылки не найден.**

осуществляется взаимодействие с сервисом путем вызова необходимых методов. Процедура создания прокси-класса зависит от того, на базе каких технологий строится клиентское приложение. В случае если выбран язык программирования C# и платформа .Net, построение клиента производится за счет вызова служебного программного средства svcutil.exe, распространяемого в составе платформы .Net, либо за счет создания ссылки на сервис в среде Microsoft VisualStudio.

Пример текста программы, использующей сервис компонента исполнения WF для запуск тестового пакета «testp», показан в листинге 5.1.

Листинг 5.1. Пример использования сервиса компонента CLAVIRE/Executor в клиентской программе

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ExecutorClient
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var service = new ExecutionService.ExecutionBrokersServiceClient())
            {
                var task = new ExecutionService.TaskDescription()
                {
                    TaskId = service.GetNewTaskId(),

                    Package = "testp",
                    Method = "arithm",

                    Params = new Dictionary<string, string>()
                    {
                        {"operation", "plus"},
                    },

                    InputFiles = new ExecutionService.TaskFileDescription[]
                    {
                        new ExecutionService.TaskFileDescription
                        {
                            StorageId = "number1",
                            FileName = "my0.in",
                            SlotName = "inf0"
                        },

                        new ExecutionService.TaskFileDescription
                        {
                            StorageId = "number25",
                            FileName = "my1.in",
                            SlotName = "inf1"
                        },
                    },

                    OutputFiles = new ExecutionService.TaskFileDescription[]
                    {
                        new ExecutionService.TaskFileDescription
                        {
                            StorageId = null,
                            FileName = "out.txt",
                            SlotName = "out_file"
                        }
                    }
                };

                service.DefineTask(task);
            }
        }
    }
}
```

```
        service.Execute(new[] { task.TaskId });  
    }  
} }  
}
```

6. ВХОДНЫЕ ДАННЫЕ

Входными данными компонента являются SOAP-запросы к соответствующему веб-сервису. Кроме того, к входным данным компонента пакетного запуска относятся:

- расписание исполнения WF, полученное от компонента планирования;
- описания ресурсов, получаемые от базы ресурсов;
- пользовательские права и квоты;
- события, получаемые в ходе выполнения заданий от компонента событийного взаимодействия;
- входные и выходные файлы, формируемые базой пакетов.

Схема классов, предоставляющих основные элементы интерфейса сервиса CLAVIRE/Executor, показана на рисунке 6.1.

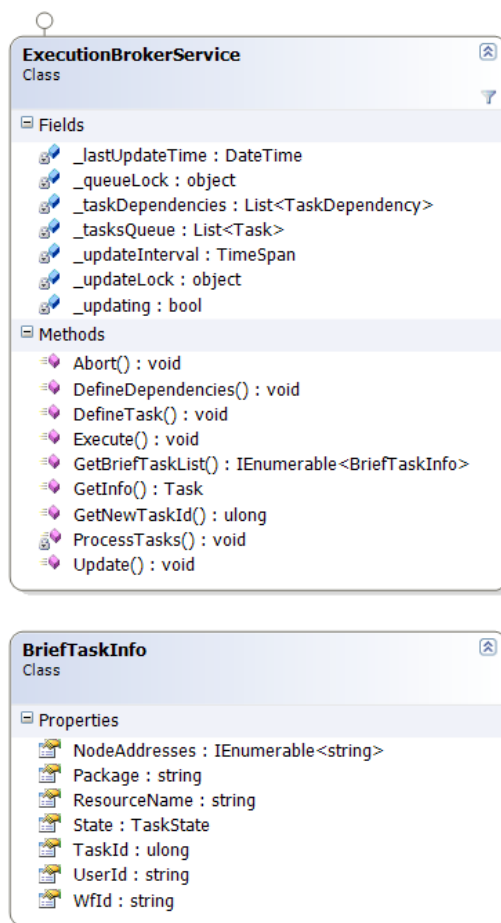


Рисунок 6.1 - Интерфейс сервиса CLAVIRE/Executor

7. ВЫХОДНЫЕ ДАННЫЕ

Выходными данными программного компонента CLAVIRE/Executor являются ответы на поступившие SOAP-запросы, полученные выходные файлы, а также скрипты и команды к управляющим системам вычислительных ресурсов.

Пример ответа сервиса компонента исполнения WF приведён на рисунке 7.1. Для отправки запроса и получения ответа использовано стандартное средство «WCF Test Client», идущее в поставке с Microsoft Visual Studio 2010.

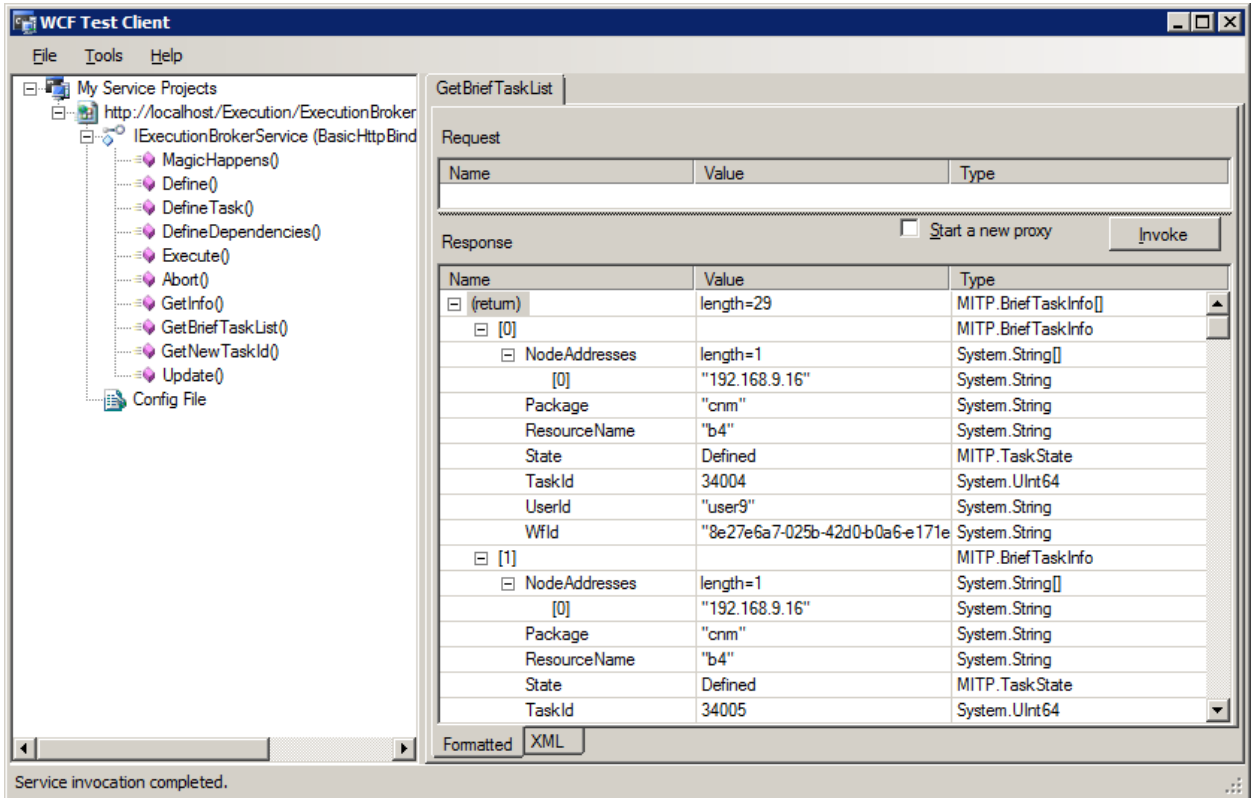


Рисунок 7.1 - Ответ компонента CLAVIRE/Еxecutor на запрос о списке активных задач

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

WSDL	Web Services Description Language, язык описания веб-сервисов
WCF	Windows Communication Foundation
IIS	Internet Information Services
МИТП	Многофункциональная инструментально-технологическая платформа

