ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 20242-3— 2012

Системы промышленной автоматизации и интеграция

СЛУЖЕБНЫЙ ИНТЕРФЕЙС ДЛЯ ИСПЫТАТЕЛЬНЫХ ПРИКЛАДНЫХ ПРОГРАММ

Часть 3

Служебный интерфейс виртуального устройства

ISO 20242-3:2011
Industrial automation systems and integration —
Service interface for testing applications —
Part 3: Virtual Device Service Interface
(IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН АНО «Международная академия менеджмента и качества бизнеса» на основе собственного аутентичного перевода на русский язык международного стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 100 «Стратегический и инновационный менеджмент»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 ноября 2012 г. № 1714-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 20242-3:2011 «Системы промышленной автоматизации и интеграция. Служебный интерфейс для испытательных прикладных программ. Часть 3. Служебный интерфейс виртуального устройства» (ISO 20242-3:2011 «Industrial automation systems and integration — Service interface for testing applications — Part 3: Virtual Device Service Interface»)

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru)

© Стандартинформ, 2014

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 1
4 Символы и сокращения терминов 2
5 Соглашения относительно определений и процедур, связанных с сервисами 2
5.1 Общие положения 2
5.2 Параметры 2
5.3 Порядок обслуживания 3
6 Модель VDSI-интерфейса 3
6.1 Виртуальные и физические устройства 3
6.2 Структура VDSI-интерфейса 4
6.3 Описание VDSI-сервисов 8
7 Рабочие состояния виртуального устройства 26
7.1 Контрольное VD-устройство 26
7.2 Рабочие состояния виртуальных устройств 27
8 Результаты выполнения сервисов 29
8.1 Дополнительная информация 29
8.2 Ошибки выполнения сервисов 30
Приложение А (справочное) Рекомендации по реализации VDSI-интерфейса 34
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации 42

Введение

Настоящий стандарт разработан для облегчения интеграции измерительных и автоматических, а также других периферийных устройств в различных компьютеризированных областях применения. В стандарте определены принципы создания драйверов устройств и режимы их работы в области использования измерительных автоматических средств.

Основная цель комплекса стандартов ИСО 20242 — обеспечение:

- независимости пользователя от операционной системы;
- независимости пользователя от технологии соединения (интерфейс устройства/сеть);
- независимости пользователя от поставщиков устройств;
- возможности сертификации драйверов устройств с подсоединенными к ним устройствами и выбранными режимами работы (с учетом применяемой компьютерной платформы);
- независимости пользователя от последующих технологических усовершенствований устройств.

Стандарты комплекса ИСО 20242 не распространяются на разработку новых семейств устройств или использование специальных технологий для интерфейсов (сетей). В стандартах приведены общие описания сетей существующих устройств и коммуникационных интерфейсов, обеспечивающих их совместимость с другими устройствами аналогичного типа и назначения.

Комплекс стандартов ИСО 20242 включает требования, распространяющиеся на:

- служебный интерфейс для управления ресурсами;
- служебный интерфейс виртуального устройства;
- шаблон функциональных характеристик устройства;
- служебный интерфейс прикладных программ;
- методы проверки на совместимость, критерии и отчеты о проведенных проверках.

Комплекс стандартов ИСО 20242 состоит из следующих частей:

- часть 1. Общий обзор;
- часть 2. Служебный интерфейс управления ресурсами;
- часть 3. Служебный интерфейс виртуального устройства;
- часть 4. Шаблон профиля функциональных возможностей устройства.

Следующие стандарты находятся в стадии разработки:

- часть 5. Служебный интерфейс прикладных программ;
- часть 6. Методы установления соответствия, критерии и отчеты.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы промышленной автоматизации и интеграция

СЛУЖЕБНЫЙ ИНТЕРФЕЙС ДЛЯ ИСПЫТАТЕЛЬНЫХ ПРИКЛАДНЫХ ПРОГРАММ

Часть 3

Служебный интерфейс виртуального устройства

Industrial automation systems and integration. Service interface for testing applications. Part 3. Virtual device service interface

Дата введения — 2014—01—01

1 Область применения

В настоящем стандарте установлен служебный интерфейс, предназначенный для связи с виртуальными устройствами, обладающими функциональными возможностями программных модулей и физических устройств, доступ к которым осуществляется с помощью сервисов управления ресурсами согласно ИСО 20242-2.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты, которые необходимо учитывать при применении настоящего стандарта. При ссылках на документы с указанной датой утверждения необходимо пользоваться только данной редакцией, если дата утверждения не приведена — последней редакцией ссылочных документов, включая любые поправки и изменения к ним.

ИСО 20242-1:2005 Системы промышленной автоматизации и интеграция. Сервисный интерфейс для испытаний. Часть 1. Обзор (ISO 20242-1:2005 Industrial automation and systems integration — Service interface for testing applications — Part 1: Overview)

ИСО 20242-2:2010 Системы промышленной автоматизации и интеграция. Сервисный интерфейс для испытаний прикладных программ. Часть 2. Сервисный интерфейс для управления ресурсами (ISO 20242-2:2010 Industrial automation and systems integration — Service interface for testing applications — Part 2: Resource Management Service Interface)

3 Термины и определения

В настоящем стандарте использованы следующие термины с соответствующими определениями:

3.1 **объект связи** (communication object): Существующий объект, доступ к чтению или записи значения которого может быть обеспечен с помощью коммуникационной функции.

[ИСО 20242-1:2005, статья 2.3]

3.2 **описание возможностей устройства** (device capability description): Текстовый файл, содержащий информацию о возможностях виртуальных устройств в установленных форматах (структуре, синтаксисе и т. д.).

[ИСО 20242-1:2005, статья 2.5]

3.3 **драйвер устройства** (device driver): Программный модуль, обеспечивающий интерфейс со служебной функцией, вызывающей адаптер платформы (переходное устройство) для доступа к физическим устройствам.

[ИСО 20242-2:2010, статья 3.1]

Издание официальное

3.4 **функциональный объект** (function object): Экземпляр, описывающий одну конкретную возможность виртуального устройства.

3.5 **операция** (operation): Экземпляр, описывающий одну законченную процедуру.

3.6 **переходное устройство** (platform adapter): Программный модуль, обеспечивающий интерфейс управления ресурсом (как определено в ИСО 20242-2), который инкапсулирует компьютерную платформу, включая операционную систему, аппаратные средства и их периферийные устройства.

Примечание — Адаптировано из ИСО 20242-2:2010, статья 3.2.

3.7 **виртуальное устройство** (virtual device): Представление одного или нескольких физических устройств и/или автономных программных экземпляров для получения однозначного мнения относительно ресурсов интерфейса связи.

4 Символы и сокращения терминов

В настоящем стандарте использованы следующие сокращения:
RMS — Сервисы управления ресурсами;
RMSI — Служебный интерфейс управления ресурсами;
SAP — Точка доступа к сервису;
VD — Виртуальное устройство;
VDS — Сервисы виртуального устройства;
VDSI — Служебный интерфейс виртуального устройства.

5 Соглашения относительно определений и процедур, связанных с сервисами

5.1 Общие положения

В настоящем стандарте применены соглашения, приведенные в ИСО/МЭК 10731.

Интерфейс между пользователем и провайдером VDS-сервиса описан с помощью сервис-примитивов (базисных элементов), которые передают параметры. Поскольку они не входят в область рассмотрения указанного стандарта, который посвящен особенностям передачи данных, для подтверждаемых сервисов применимы только запрашиваемые и подтверждающие сервис-примитивы. Для обработки событий, возникающих у провайдера VDS-сервисов, использованы индикаторные и ответные сервис-примитивы.

Модель обслуживания, сервис-примитивы и схемы последовательности операций являются абстрактными описаниями, которые не представляют собой спецификацию для реализации.

В приложении А приведены правила стандартной реализации.

5.2 Параметры

Сервис-примитивы, применяемые для представления взаимодействия между пользователем и провайдером услуг (см. ИСО/МЭК 10731), передают параметры, которые индицируют информацию, доступную при данном взаимодействии.

В настоящем стандарте использована табличная форма представления параметров компонентов для сервис-примитивов VDS. Параметры, которые применимы к каждой группе сервис-примитивов VDS, приведены в таблицах остальной части настоящего стандарта. Каждая из них включает по три столбца, содержащих наименование сервис-параметра и параметра направления переноса, используемого VDS-сервисом:

- входные параметры запрашиваемого или индицируемого примитива;
- выходные параметры подтверждающего или ответного примитива.

Один параметр (или его часть) указан в каждой строке соответствующей таблицы. Под столбцами сервис-примитивов находится код, определяющий тип применения параметра в направлении примитива и параметра, приведенного в столбце:

M — обязателен для примитива;

I — опция программной реализации, которая может предусматриваться или не предусматриваться, в зависимости от реализации провайдера VDS-сервисов;

С — условный, зависящий от других параметров или технических средств пользователя VDS-сервисов;

S — выбираемый элемент;

О — дополнительный для сервиса, наличие которого зависит от содержания описания функциональных возможностей устройства согласно стандарту ИСО 20242-4.

Примечание — Отсутствие значения никогда не указывается.

5.3 Порядок обслуживания

5.3.1 Подтвержденные VDS-сервисы

Запрашивающий пользователь предоставляет запросный примитив в служебный интерфейс виртуального устройства (VDSI), что предполагает наличие точки доступа к сервису (SAP). Соответствующий элемент обработки сервиса передает примитив подтверждения пользователю после всех необходимых взаимодействий или появления ошибки.

5.3.2 Обработка событий с помощью VDS-сервиса

Пользователь создает точку доступа к сервису (SAP) в VDSI-интерфейсе для обработки событий, где их формирует индикаторный примитив, а пользователь VDSI-интерфейса выдает ответный примитив после завершения всех необходимых взаимодействий или появления ошибки (см. рисунок 1).

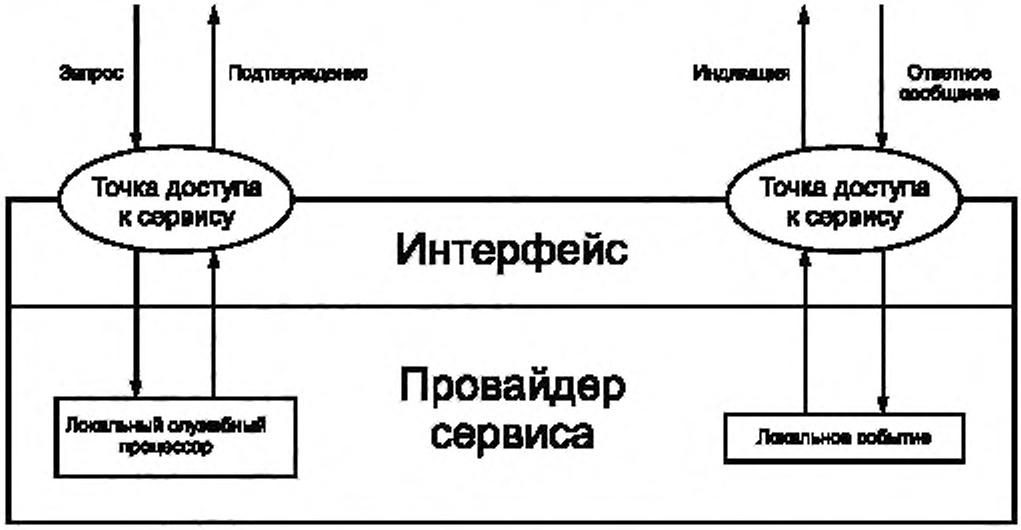


Рисунок 1 — Обработка локальных событий с помощью VDS-сервиса

6 Модель VDSI-интерфейса

6.1 Виртуальные и физические устройства

Виртуальные устройства в ИСО 20242 определены исходя из потребностей пользователя VDSI-интерфейса. Их применение требует таких функциональных возможностей, которые могут представляться одним или несколькими физическими устройствами и/или программными модулями (см. рисунок 2). Эти возможности (характеристики) сгруппированы в соответствии с виртуальными устройствами, которые могут принадлежать:

- специализированному физическому устройству;
- ряду физических устройств;
- части физического устройства;
- программному модулю в драйвере устройства или в устройстве сопряжения платформы.

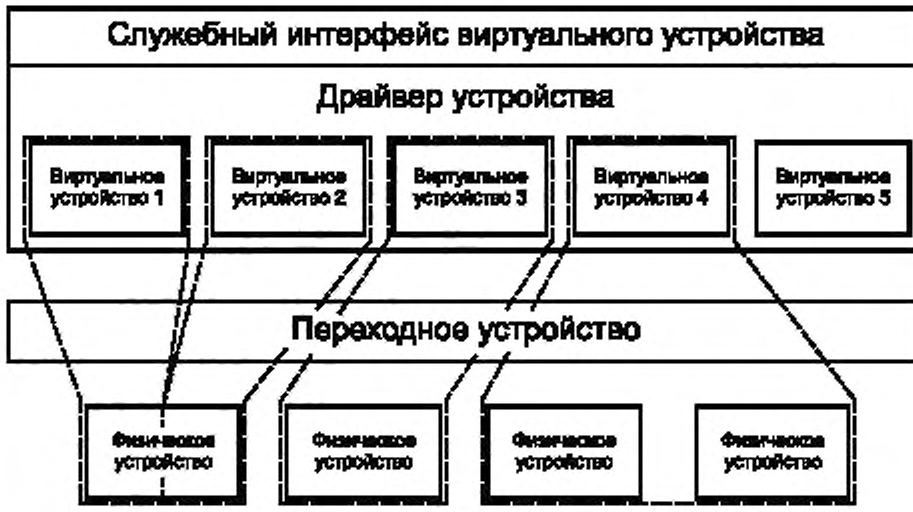


Рисунок 2 — Конфигурация виртуальных устройств

6.2 Структура VDSI-интерфейса

6.2.1 Общие положения

Сущность сервисов виртуальных устройств (VDS) содержит точку доступа к сервисам их конструирования и использования и обеспечивает служебный интерфейс VDSI. Виртуальные устройства содержат функциональные объекты с операциями и коммуникационными объектами. На рисунке 3 представлена структура UML диаграммы классов.

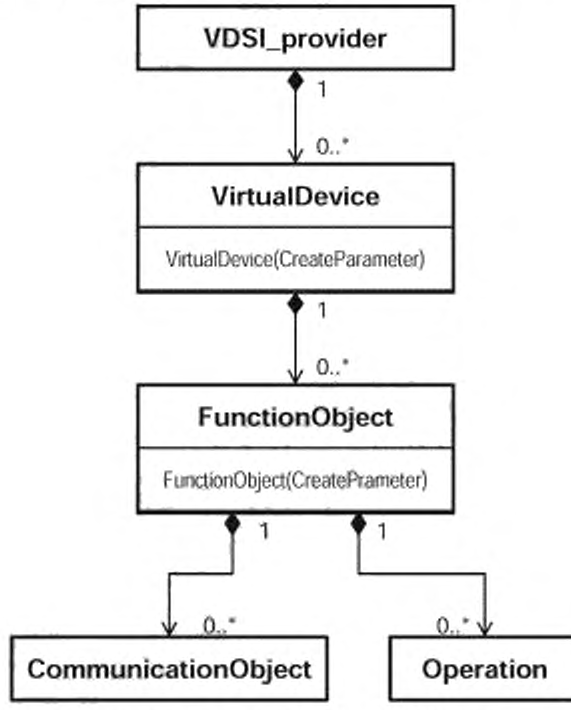


Рисунок 3 — Диаграмма UML-класса VDSI-модели

6.2.2 Базовое управление

Сервисы для базового управления открывают VDSI-интерфейс для его применения и контроля других сервисов. В таблице 1 приведен их перечень.

Таблица 1 — Сервисы для базового управления

Наименование сервиса	Обозначение сервиса для его идентификации	Примечания
Attach VDSI Entity (прикрепление VDSI-объекта)	VDSI_Attach	Позволяет открывать VDSI-объект для всех других сервисов и создавать точку доступа к сервису при обработке событий (см. 5.3.2)
Cancel Service (отмена сервиса)	VDSI_Cancel	Позволяет отменять выполнение выбранного сервиса

6.2.3 Обработка виртуального устройства

Перед использованием виртуального устройства на его основе должны быть созданы экземпляры (см. 6.2.6) согласно описанию функциональных возможностей ИСО 20242-4. Число экземпляров устройства зависит от требований к его применению и может ограничиваться физическими или программными ресурсами. В таблице 2 приведен перечень сервисов, которые должны реализовываться виртуальными устройствами.

Таблица 2 — Сервисы виртуального устройства

Наименование сервиса	Обозначение сервиса для его идентификации	Примечания
Instantiate Virtual Device (реализация виртуального устройства)	VDSI_Initiate	Позволяет создавать экземпляры виртуального устройства и открывает их для использования другими сервисами
Remove Virtual Device (удаление виртуального устройства)	VDSI_Conclude	Позволяет удалять экземпляры виртуального устройства при отсутствии условий его поддержания
Destroy Virtual Device (разрушение виртуального устройства	VDSI_Abort	Позволяет безусловно удалять экземпляры виртуального устройства
Get Virtual Device Status (получение состояния виртуального устройства)	VDSI_Status	Позволяет получать состояние виртуального устройства
Identify Virtual Device (идентификация виртуального устройства)	VDSI_Identify	Позволяет получать экземпляр и неоспоримую идентификацию для поставщика виртуального устройства

6.2.4 Обработка функционального объекта

Функциональные объекты — возможности виртуальных устройств, необходимые для выполнения поставленных перед ними задач. Их примерами являются программные объекты, описываемые как классы или измерительные каналы со специальной обработкой сигналов и заданными параметрами. Функциональный объект перед его активным использованием по назначению необходимо охарактеризовать. Класс таких объектов определяется в описании функциональных возможностей устройства в соответствии с ИСО 20242-4. Число экземпляров объекта зависит от требований к его применению и может быть ограничено физическими и программными ресурсами. После определения характеристик физического объекта могут использоваться связанные операции и создаваться ассоциированные объекты связи. В таблице 3 приведен перечень сервисов, которые должны применяться совместно с функциональными объектами и их операциями.

Таблица 3 — Сервисы функционального объекта

Наименование сервиса	Обозначение сервиса для его идентификации	Примечания
Instantiate Function Object (реализация функционального объекта)	VDSI_Create-FuncObject	Позволяет создавать экземпляры функционального объекта и открывать его для применения в других сервисах
Remove Function Object (удаление функционального объекта)	VDSI_Delete-FuncObject	Позволяет удалять экземпляры функционального объекта, если нет условий по его поддержанию
Execute Operation (выполнение операции)	VDSI_Execute	Позволяет запускать выполнение процедуры, связанной с функциональным объектом

6.2.5 Обработка объектов связи

Объекты связи — источники и пункты назначения для обмена данными. Примерами таких объектов являются параметры для сигнальных процессоров или результаты измерений реальных устройств или атрибутов класса для программных объектов, описываемых в этих классах. Объекты связи должны характеризоваться перед получением доступа к ним. Тип их данных определяется в описании функциональных возможностей устройства в соответствии с ИСО 20242-4. При этом может создаваться только один экземпляр объекта связи. После создания экземпляра он может использовать незатребованные сообщения для посылки данных для применения или запроса из приложений. Управление подобными действиями может выполняться с помощью дополнительных функциональных объектов, определенных в описании функциональных возможностей устройства. В таблице 4 приведен перечень сервисов, которые должны применяться для объектов связи или с их помощью.

Таблица 4 — Сервисы объектов связи

Наименование сервиса	Обозначение сервиса для его идентификации	Примечания
Instantiate Communication Object (реализация объекта связи)	VDSI_Create-CommObject	Позволяет создавать экземпляр объекта связи и открывать его для использования другими сервисами
Remove Communication Object (удаление объекта связи)	VDSI_Delete-CommObject	Позволяет удалять экземпляр объекта связи, если отсутствуют условия его поддержания
Write Data to Communication Object (запись данных в объекте связи)	VDSI_Write	Позволяет запускать передачу данных из приложения в объект связи запросом приложения
Read Data from Communication Object (считывание данных из объекта связи	VDSI_Read	Позволяет запускать передачу данных из объекта связи в приложение запросом приложения
Report Data to Application (сообщение данных в приложение)	VDSI_InfReport	Позволяет запускать передачу данных из объекта связи в приложение запросом объекта связи
Request Data from Application (запрос данных из приложения)	VDSI_Accept	Позволяет запускать передачу данных из приложения в объект связи запросом объекта связи

6.2.6 Время жизни VDSI-объектов

Рисунок 4 иллюстрирует процедуры создания и удаления VDSI-объектов. Функциональные объекты не могут создаваться до того, как связанное виртуальное устройство не будет существовать, объекты связи — до того, как связанный функциональный объект не будет существовать.

Функциональные объекты не могут быть удалены до того, как не будут удалены связанные объекты связи. Операции будут удаляться в точности вместе с их функциональным объектом. Виртуальные устройства не могут удаляться до того, пока не будут удалены связанные функциональные объекты. Исключения из этого правила рассмотрены в 7.1.3.8 при использовании специального виртуального устройства для процедур контроля.

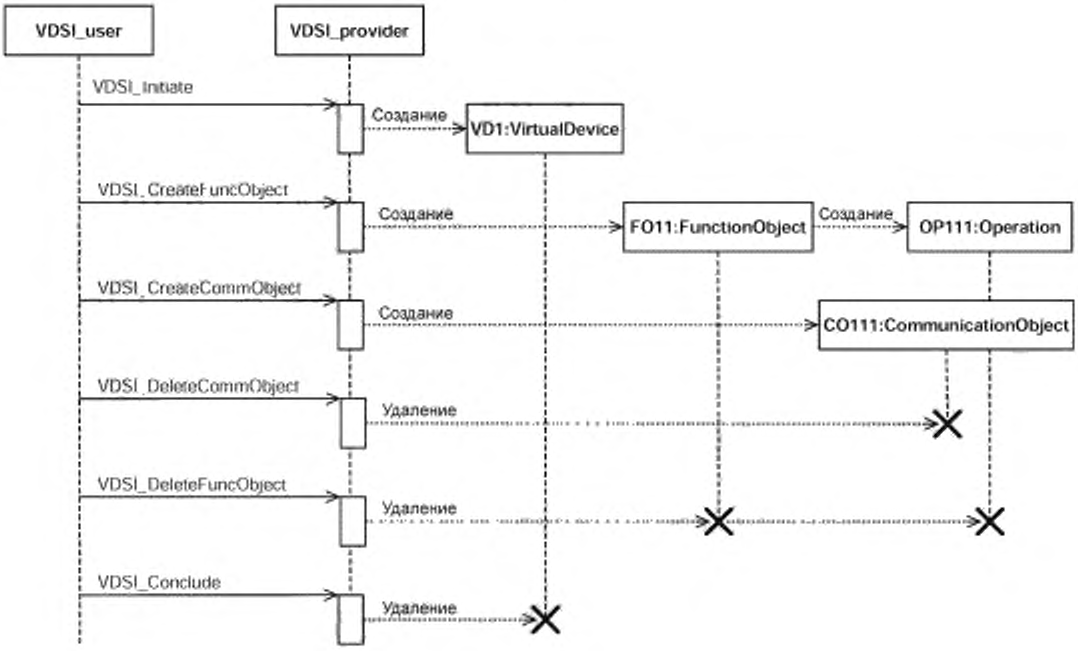


Рисунок 4 — UML-схема последовательности операций для продолжительности жизни VDSI-объектов

6.2.7 Выполняемые операции

Выполнение операций начинается с запроса сервис-примитива VDSI_Execute. Операции принадлежат к функциональным объектам и могут быть входными параметрами и результатами выполнения с выходными параметрами.

6.2.8 Обмен данными с объектами связи

Обмен данными с объектами связи может запускаться пользователем VDSI-интерфейса или локальным событием у его провайдера. На рисунке 5 показана процедура обмена данными.

Если пользователь VDSI-интерфейса запускает сервис VDSI_Read, то его провайдер выводит данные из адресуемого объекта связи (getData на рисунке 5) и доставляет их пользователю.

Если пользователь VDSI-интерфейса запускает сервис VDSI_Write, то его провайдер забирает переносимые этим сервисом данные в адресуемый объект связи (getData на рисунке 5).

Если локальное событие возникает при посылке объекта связи, то данные будут направляться провайдеру VDSI-интерфейса (putData на рисунке 5) для предоставления их его пользователю с сервисом VDSI_InfReport.

Если локальное событие возникает при приеме объекта связи, то данные будут запрашиваться от провайдера VDSI-интерфейса, который отбирает данные от его пользователя с сервисом VDSI_Accept и предоставляет их в объект связи (getData на рисунке 5).

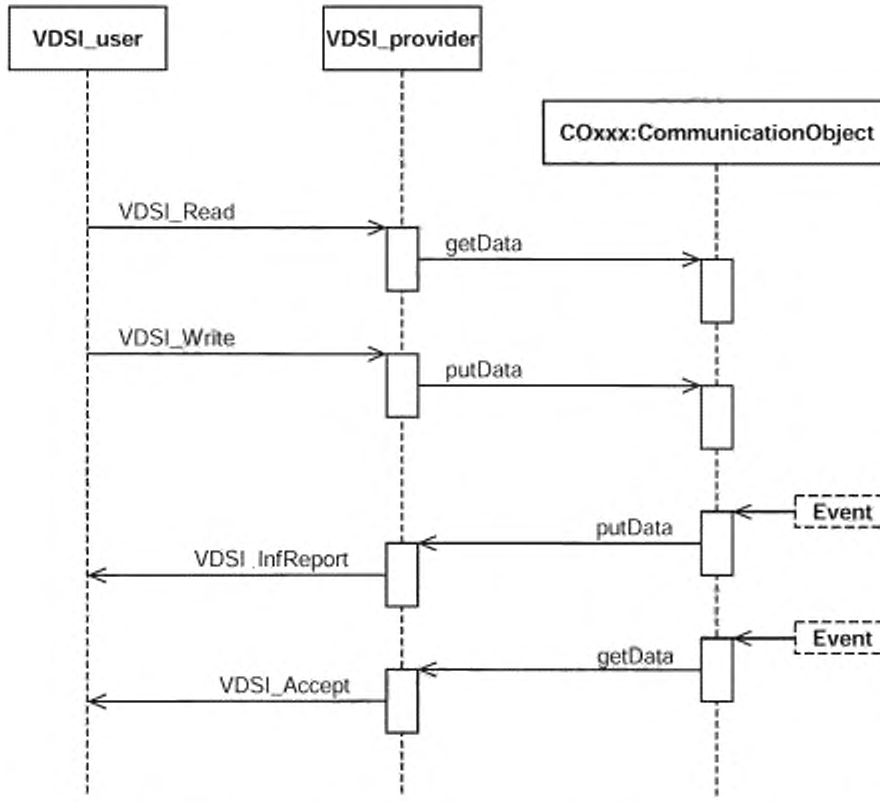


Рисунок 5 — UML-схема последовательности операций для обмена данными с объектами связи

6.3 Описание VDSI-сервисов

6.3.1 Сервис Attach VDSI Entity (Достижение VDSI-интерфейса)

6.3.1.1 Краткое описание сервиса

Данный сервис используется для открытия объекта VDSI-интерфейса для применения других сервисов или установки точек доступа к ним для локальных событий.

6.3.1.2 Параметры сервиса

6.3.1.2.1 Общие положения

Параметры данного сервиса указаны в таблице 5.

Таблица 5 — Параметры сервиса Attach VDSI Entity

Наименование параметра	Req	Cnf
Argument	C	
Fetch service access point	C	
Report service access point	C	
Result (+)		S
Result (–)		S
Invocation Error		M

6.3.1.2.2 Параметр Argument (Аргумент)

6.3.1.2.2.1 Общие положения

Argument содержит параметры запроса на сервис.

6.3.1.2.2.2 Параметр Fetch service access point (Выбор точки доступа к сервису)

Данный параметр позволяет определять точку доступа к сервису, который будет использоваться виртуальным устройством для выбора данных, не запрашиваемых пользователем VDSI-интерфейса и описанных в 6.3.16.

6.3.1.2.2.3 Параметр Report service access point (Сообщение о точке доступа к сервису)

Данный параметр позволяет определять точку доступа к сервису, который применяется виртуальным устройством для посылки данных, не запрашиваемых пользователем VDSI-интерфейсом и описанных в 6.3.7.

6.3.1.2.3 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.1.2.4 Параметр Result (–) (Отрицательный результат)

6.3.1.2.4.1 Общие положения

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.1.2.4.2 Параметр Invocation Error

Данный параметр должен индицировать следующие виды ошибок:

- VDSI-интерфейс уже открыт;
- прочие.

6.3.1.3 Процедура выполнения сервиса

Если VDSI-интерфейс пока не открыт, то он готовится к выполнению сервиса; в противном случае будет возвращаться ошибка. Если точки доступа к сервисам заданы, то VDSI-интерфейс будет активирован для передачи непредусмотренных (незапрашиваемых) данных с помощью индикационных и ответных сервис-примитивов.

6.3.2 Сервис Cancel (Отмена)

6.3.2.1 Краткое описание сервиса

Этот сервис позволяет отменять любые открытые сервисы, за исключением Attach VDSI Entity, Instantiate Virtual Device или другие сервисы отмены.

6.3.2.2 Параметры сервиса

Параметры данного сервиса представлены в таблице 6.

Таблица 6 — Сервис Cancel

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
User Service Handle	M	
User Handle of cancelled Service	M	
Result (+)		S
User Service Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		M

6.3.2.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.2.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство обработкой, определенной в VDSI-интерфейсе с помощью сервиса Initiate Virtual Device (см. 6.3.3).

6.3.2.2.1.2 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.2.2.1.3 Параметр User Handle of cancelled Service (Дескриптор пользователя отменяемого сервиса)

Данный параметр является задаваемым пользователем идентификатором для подлежащего отмене сервиса.

6.3.2.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.2.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для сервиса, предоставляемого вместе с запросом на него.

6.3.2.2.2.2 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.2.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.2.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.2.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.2.3 Процедура выполнения сервиса

Если VDSI-интерфейс еще не открыт, то он готовится к выполнению сервисов; в противном случае будет возвращаться ошибка. Если задаются точки доступа к сервису для локальных событий, то VDSI-интерфейс активируется для передачи незатребованных (непредусмотренных) данных с помощью индикационных и ответных сервис-примитивов.

6.3.3 Сервис Initiate Virtual Device (Реализация (инстанциация) виртуального устройства)

6.3.3.1 Краткое описание сервиса

Данный сервис позволяет создавать «пустое» виртуальное устройство, содержание которого определяется последующими сервисами. Тип виртуального устройства устанавливается с помощью описания его функциональных возможностей согласно ИСО 20242-4.

6.3.3.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 7.

Таблица 7 — Параметры сервиса Instantiate Virtual Device

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Type Identifier	M	
User Service Handle	M	
Create Parameter	O	
Result (+)		S
User Service Handle		M
Virtual Device Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.3.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.3.2.1.1 Параметр Virtual Device Type Identifier (Идентификатор типа виртуального устройства)

Данный параметр позволяет идентифицировать тип виртуального устройства, определяемый с помощью описания его функциональных возможностей согласно ИСО 20242-4.

6.3.3.2.1.2 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором указанного сервиса.

6.3.3.2.1.3 Параметр Create Parameter (Создание параметра)

Данный параметр является вспомогательным. Его наличие и тип зависят от соответствующих определений в описании функциональных возможностей устройства для объекта VDSI-интерфейса.

6.3.3.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.3.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.3.2.2.2 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр — дескриптор реализованного виртуального устройства, который используется вместе с другими сервисами для их направления в это устройство.

6.3.3.2.2.3 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.3.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.3.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.3.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.3.2.3.3 Параметр Invocation Error

Выбор подобного типа параметра должен показывать следующие виды ошибок:

VDSI-интерфейс не открывается;

недопустимый идентификатор типа виртуального устройства;

иные.

6.3.3.3 Процедура выполнения сервиса

Если VDSI-интерфейс открыт, и идентификатор шаблона виртуального устройства действующий, то оно создается, а дескриптор определяется для последующего доступа к этому устройству. Любые действия могут выполняться для гарантии его готовности к работе и зависеть от приведения в соответствие с физическими устройствами.

Существуют несколько возможных вариантов идентификаторов шаблона виртуального устройства, число которых может ограничиваться программными или аппаратными ресурсами, а также определением его функциональных возможностей.

6.3.4 Сервис Conclude Virtual Device (Завершение виртуального устройства)

6.3.4.1 Краткое описание сервиса

Данный сервис позволяет удалять виртуальное устройство, которое создано с помощью сервиса Instantiate Virtual Device (см. 6.3.3), при отсутствии локальных условий для поддержания этого варианта.

6.3.4.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 8.

Таблица 8 — Параметры сервиса Conclude Virtual Device

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.4.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.4.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Virtual Device (см. 6.3.3).

6.3.4.2.1.2 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.4.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.4.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.4.2.2.2 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.4.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.4.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.4.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.4.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен индицировать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- иные.

6.3.4.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства действующий и нет условий для поддержки его варианта, то оно будет удаляться вместе с содержимым. Пример условий для поддержки варианта — открытая процедура связи с физическими устройствами, которые не могут быть повреждены в данном состоянии.

6.3.5 Сервис Abort Virtual Device (Остановка виртуального устройства)

6.3.5.1 Краткое описание сервиса

Данный сервис позволяет удалить виртуальное устройство, которое создано с помощью сервиса Initiate Virtual Device (см. 6.3.3).

6.3.5.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 9.

Таблица 9 — Параметры сервиса Abort Virtual Device

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Result (–)		S
User Service Handle		M
Invocation Error		M

6.3.5.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.5.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с использованием дескриптора, определяемого в VDSI-интерфейсе с помощью сервиса Initiate Virtual Device (см. 6.3.3).

6.3.5.2.1.2 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.5.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.5.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.5.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.5.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.5.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.5.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

недопустимый дескриптор виртуального устройства;

иные.

6.3.5.3 Процедура выполнения сервиса

Если дескриптор действующий, то идентифицируемое виртуальное устройство будет удаляться вместе со всем содержимым. При этом неприменимы никакие условия к его сохранению.

6.3.6 Сервис Get Virtual Device Status (Получение состояния виртуального устройства)

6.3.6.1 Краткое описание сервиса

Данный сервис позволяет индицировать состояние виртуального устройства.

6.3.6.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 10.

Таблица 10 — Параметры сервиса Get Virtual Device Status

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Logical State		M
Physical State		M
Operating State		M
Additional State Info		I
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.6.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.6.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.6.2.1.2 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.6.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.6.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.6.2.2.2 Параметр Logical State (Логическое состояние)

Данный параметр описывает внутреннее состояние виртуального устройства среди следующих состояний:

- все сервисы могут использоваться без каких-либо ограничений;

- сервисы, вызывающие изменение данных или состояния указанного виртуального устройства, отбраковываются;

- будут выполняться только сервисы виртуального устройства, перечисленные в таблице 2;
- иные.

6.3.6.2.2.3 Параметр Physical State (Физическое состояние)

Данный параметр описывает физическое состояние виртуального устройства среди следующих состояний:

- является работоспособным без каких-либо ограничений;
- является частично работоспособным, при этом один или несколько сервисов могут быть неработоспособными;
- является неработоспособным;
- необходимо техническое обслуживание; не может быть введено в эксплуатацию для выполнения сервиса;
- активна процедура проверки конфигурации; анализируется структура и содержание устройства, которое находится в рабочем состоянии Check (см. следующий раздел) и не может быть переведено в другое;
- иные.

6.3.6.2.2.4 Параметр Operating State (Рабочее состояние)

Данный параметр описывает рабочее состояние виртуального устройства среди следующих состояний:

- инициализированное;
- подготовка;
- проверка;
- работа;
- оценка;
- проверка.

Для получения более подробной информации см. раздел 7.

6.3.6.2.2.5 Параметр Additional State Info (Дополнительная информация о состоянии)

Данный параметр является опцией реализации и может быть типа Result Information, описанного в 8.1, или типа Result Error, описанного в 8.2, и дающего дополнительную информацию относительно состояния виртуального устройства.

6.3.6.2.2.6 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

3.6.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.6.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.6.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.6.2.3.3 Параметр Invocation Error

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- иные.

6.3.6.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства действующий, то его состояние будет анализироваться, передаваться и не определяться для Control VD (см. 7.1). Если данный сервис применим к контрольному VD-устройству, то параметр Invocation Error будет возвращаться.

6.3.7 Сервис Identify Virtual Device (Идентификация виртуального устройства)

6.3.7.1 Краткое описание сервиса

Данный сервис позволяет идентифицировать виртуальное устройство.

6.3.7.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 11.

Таблица 11 — Параметры сервиса Identify Virtual Device

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Virtual Device Version		M
Virtual Device Type Description		M
Version of VDSI		M
Virtual Device Vendor		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.7.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.7.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр идентифицирует виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.7.2.1.2 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.7.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.7.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.7.2.2.2 Параметр Virtual Device Version (Версия виртуального устройства)

Данный параметр определяет версию виртуального устройства, идентифицируемую его дескриптором. Эта версия может отличаться для различных вариантов типов такого устройства и зависеть от его соответствия физическим устройствам и информации, предоставляемой ими.

При рассмотрении контрольного виртуального устройства (Control VD) (см. 7.1) этот параметр описывает специальную версию для всех VDSI-интерфейсов устройств.

6.3.7.2.2.3 Параметр Virtual Device Type Description (Описание типа виртуального устройства)

Данный параметр содержит описание типа виртуального устройства и поэтому соответствует его шаблонам в описании функциональных возможностей согласно стандарту ИСО 20242 (часть 4).

При рассмотрении контрольного виртуального устройства (Control VD) (см. 7.1) этот параметр позволяет специально описывать VDSI-интерфейс и будет соответствовать информации в заголовке описания его функциональных возможностей.

6.3.7.2.2.4 Параметр Version of VDSI (Версия VDSI-интерфейса)

Данный параметр дает версию VDSI-интерфейса и может проверяться относительно версии, содержащейся в описании функциональных возможностей устройства для гарантии того, что принятые шаблоны действительно предоставляются указанным VDSI-интерфейсом.

6.3.7.2.2.5 Параметр Virtual Device Vendor (Поставщик виртуального устройства)

Данный параметр дает информацию для индивидуализации рассматриваемого виртуального устройства и может использоваться только в том случае, когда описания его возможностей недостаточно для правильной работы устройства. Содержимое этого параметра определяет поставщик.

При рассмотрении контрольного виртуального устройства (Control VD) (см. 7.1) данный параметр дает информацию для идентификации поставщика VDSI-интерфейса.

6.3.7.2.2.6 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.7.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.7.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.7.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.7.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- иные.

6.3.7.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства действующий, то будет передаваться идентификационная информация о нем.

6.3.8 Параметр Instantiate Function Object (Реализация (инстанциация) функционального объекта)

6.3.8.1 Краткое описание сервиса

Данный сервис позволяет создавать функциональный объект вместе с его операциями. Объекты связи, принадлежащие этому объекту, созданы с помощью дополнительных сервисов.

6.3.8.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 12.

Таблица 12 — Параметры сервиса Instantiate Function Object

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
Function Object Template Identifier	M	
User Service Handle	M	
Create Parameter	O	
Result (+)		S
User Service Handle		M
Function Object Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.8.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.8.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.8.2.1.2 Параметр Function Object Template Identifier (Идентификатор шаблона функционального объекта)

Данный параметр позволяет идентифицировать шаблон функционального объекта, определенный в описании функциональных возможностей устройства согласно стандарту ИСО 20242 (часть 4).

6.3.8.2.1.3 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.8.2.1.4 Параметр Create Parameter (Создание параметра)

Данный параметр является вспомогательным, наличие и тип которого зависят от соответствующих определений в описании функциональных возможностей устройства для объекта — VDSI-интерфейса.

6.3.8.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.8.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.8.2.2.2 Параметр Function Object Handle (Дескриптор функционального объекта)

Данный параметр — дескриптор для реализации экземпляра функционального объекта, используемый вместе с другими сервисами для вхождения в указанный функциональный объект.

6.3.8.2.2.3 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации экземпляра и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.8.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.8.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.8.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.8.2.3.3 Параметр Invocation Error (Ошибка вызова процедуры или функции)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- VDSI-интерфейс не открывается;
- недопустимый идентификатор шаблона функционального объекта;
- иные.

6.3.8.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства и идентификатор шаблона функционального объекта действующие, то этот объект создается, а дескриптор — определяется для последующего доступа к нему. Любые действия могут выполняться для гарантии пригодности данного функционального объекта.

Возможно несколько вариантов идентификатора шаблона одного и того же функционального объекта, число которых может ограничиваться программными и аппаратными ресурсами, а также определением в описании функциональных возможностей устройства.

6.3.9 Сервис Remove Function Object (Удаление функционального объекта)

6.3.9.1 Краткое описание сервиса

Данный сервис позволяет удалять функциональный объект, который создан с помощью сервиса Instantiate Function Object (см. 6.3.8), если отсутствуют локальные условия для сохранения экземпляра.

6.3.9.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 13.

Таблица 13 — Параметры сервиса Remove Function Object

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
Function Object Handle	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.9.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.9.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с применением дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.9.2.1.2 Параметр Function Object Handle (Дескриптор функционального объекта)

Данный параметр позволяет идентифицировать функциональный объект с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Function Object (см. 6.3.8).

6.3.9.2.1.3 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса.

6.3.9.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.9.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.9.2.2.2 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.9.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно

6.3.9.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.9.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.9.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- недопустимый дескриптор функционального объекта;
- объект связи еще существует;
- иные.

6.3.9.3 Процедура выполнения сервиса

Если дескрипторы виртуального устройства и функционального объекта действующие и отсутствуют условия для сохранения экземпляра этого объекта, его следует удалить вместе со всем содержимым. Примером условия сохранения такого экземпляра является открытая процедура связи с физическими устройствами, которые в этом состоянии не могут быть повреждены.

Если имеются реализованные объекты связи с данным функциональным объектом, они должны удаляться до его удаления.

6.3.10 Сервис Execute Operation

6.3.10.1 Краткое описание сервиса

Данный сервис позволяет выполнять процедуру, связанную с функциональным объектом.

6.3.10.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 14.

Таблица 14 — Параметры сервиса Execute Operation

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
Function Object Handle	M	
Operation Identifier	M	
User Service Handle	M	
Operation Input Data	O	
Result (+)		S
User Service Handle		M
Operation Output Data		O
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.10.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.10.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр идентифицирует виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.10.2.1.2 Параметр Function Object Handle (Дескриптор функционального объекта)

Данный параметр позволяет идентифицировать функциональный объект с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.8).

6.3.10.2.1.3 Параметр Operation Identifier (Идентификатор операции)

Данный параметр позволяет идентифицировать операцию по отношению к рассматриваемому функциональному объекту.

6.3.10.2.1.4 Параметр Operation Input Data (Входные данные для операции)

Указанный параметр является дополнительным и содержит входные данные для операции. Его наличие и тип зависят от соответствующих определений в описании функциональных возможностей устройства для данного VDSI-интерфейса.

6.3.10.2.1.5 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.10.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.10.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.10.2.2.2 Параметр Operation Output Data (Выходные данные для операции)

Указанный параметр является дополнительным и содержит выходные данные для операции. Его наличие и тип зависят от соответствующих определений в описании функциональных возможностей устройства для данного VDSI-интерфейса.

6.3.10.2.2.3 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации экземпляра и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.10.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.10.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.10.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.10.2.3.3 Параметр Invocation Error (Ошибка вызова процедуры или функции)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- недопустимый дескриптор функционального объекта;
- недопустимый идентификатор операции;
- иные.

6.3.10.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства и функционального объекта, а также идентификатор операции являются действующими, то рассматриваемая операция будет выполняться.

6.3.11 Сервис Instantiate Communication Object (Реализованный объект связи)

6.3.11.1 Краткое описание сервиса

Данный сервис позволяет создавать объект связи в функциональном объекте.

6.3.11.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 15.

Таблица 15 — Параметры сервиса Instantiate Communication Object

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
Function Object Handle	M	
Communication Object Identifier	M	
User Object Handle	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.11.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.11.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Virtual Device (см. 6.3.3).

6.3.11.2.1.2 Параметр Function Object Handle (Дескриптор функционального объекта)

Данный параметр позволяет идентифицировать функциональный объект с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.8).

6.3.11.2.1.3 Параметр Communication Object Identifier (Идентификатор объекта связи)

Данный параметр позволяет идентифицировать объект связи, определенный в описании функциональных возможностей устройства согласно ИСО 20242 (часть 4).

6.3.11.2.1.4 Параметр User Object Handle (Дескриптор объекта пользователя)

Данный параметр позволяет идентифицировать объект связи, определенный пользователем при передаче незапрашиваемых (непредусмотренных) данных для этого объекта, запрашиваемого у пользователя (см. также сервисы Report Data to Application (см. 6.3.15) и Request Data from Application (см. 6.3.16).

6.3.11.2.1.5 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором указанного сервиса.

6.3.11.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.11.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.11.2.2.2 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации экземпляра и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.11.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.11.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.11.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.11.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- недопустимый дескриптор функционального объекта;

- недопустимый идентификатор объекта связи;
- иные.

6.3.11.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства и функционального объекта, а также идентификатор операции являются действующими, то рассматриваемый объект связи будет создаваться, а дескриптор объекта пользователя — присоединяться. При возникновении локального события для передачи данных между объектом связи и пользователем VDS-сервиса дескриптор объекта пользователя будет применяться для идентификации объекта связи у пользователя VDS-сервиса. Существует лишь один возможный экземпляр для каждого идентификатора этого объекта.

6.3.12 Сервис Remove Communication Object

6.3.12.1 Краткое описание сервиса

Данный сервис позволяет удалять объект связи, который создан с помощью сервиса Instantiate Communication Object (см. 6.3.11) при отсутствии локальных условий сохранения варианта.

6.3.12.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 16.

Таблица 16 — Параметры сервиса Remove Communication Object

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
Function Object Handle	M	
Communication Object Identifier	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
User Object Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.12.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.12.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с использованием дескриптора, определяемого в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.12.2.1.2 Параметр Function Object Handle (Дескриптор функционального объекта)

Данный параметр позволяет идентифицировать функциональный объект с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.8).

6.3.12.2.1.3 Параметр Communication Object Identifier (Идентификатор объекта связи)

Данный параметр позволяет идентифицировать объект связи, определенный в описании функциональных возможностей устройства согласно ИСО 20242 (часть 4).

6.3.12.2.1.4 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором указанного сервиса.

6.3.12.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.12.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на сервис.

6.3.12.2.2.2 Параметр User Object Handle (Дескриптор объекта пользователя)

Данный параметр является идентификатором объекта, определяемого пользователем, который закрыт с помощью сервиса Instantiate Communication Object (см. 6.3.11).

6.3.12.2.2.3 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации экземпляра и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.12.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.12.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.12.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.12.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- недопустимый дескриптор функционального объекта;
- иные.

6.3.12.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства и функционального объекта, а также идентификатор операции являются действующими, а локальные условия сохранения объекта связи отсутствуют, то он будет удаляться. Примером условия сохранения варианта объекта связи является открытая процедура связи с физическими устройствами, которая не может быть прервана в этом состоянии.

6.3.13 Сервис Write Data to Communication Object (Запись данных в объект связи)

6.3.13.1 Краткое описание сервиса

Указанный сервис позволяет передавать данные от пользователя VDSI-интерфейса в объект связи.

6.3.13.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 17.

Таблица 17 — Параметры сервиса Write Data to Communication Object

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
Function Object Handle	M	
Communication Object Identifier	M	
User Data for Communication Object	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.13.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.13.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр позволяет идентифицировать виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.13.2.1.2 Параметр Function Object Handle (Дескриптор функционального объекта)

Данный параметр позволяет идентифицировать функциональный объект с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.8).

6.3.13.2.1.3 Параметр Communication Object Identifier (Идентификатор объекта связи)

Данный параметр позволяет идентифицировать объект связи, определенный в описании функциональных возможностей устройства согласно ИСО 20242 (часть 4).

6.3.13.2.1.4 Параметр User Data for Communication Object (Данные пользователя для объекта связи)

Указанный параметр содержит данные пользователя, которые должны записываться в объект связи. Тип данных определяется в описании функциональных возможностей устройства для VDSI-интерфейса согласно ИСО 20242 (часть 4).

6.3.13.2.1.5 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором указанного сервиса.

6.3.13.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.13.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора для указанного сервиса, предоставляемого вместе с запросом на него.

6.3.13.2.2.2 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации экземпляра и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.13.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.13.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.13.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.13.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- недопустимый дескриптор функционального объекта;
- недопустимый идентификатор объекта связи;
- недопустимые данные пользователя;
- иные.

6.3.13.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства и функционального объекта, а также идентификатор объекта связи действующие, то данные пользователя будут записываться в объект связи. Если этот объект находится внутри физического устройства, то будут выполняться все необходимые связи для передачи данных пользователя на физическое устройство через RMSI-интерфейс.

6.3.14 Сервис Read Data from Communication Object (Считывание данных из объекта связи)

6.3.14.1 Краткое описание сервиса

Указанный сервис позволяет передавать данные от объекта связи на VDSI-интерфейс пользователя.

6.3.14.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 18.

Таблица 18 — Параметры сервиса Read Data from Communication Object

Наименование параметра	Req	Cnf
Argument	M	
Virtual Device Handle	M	
Function Object Handle	M	
Communication Object Identifier	M	
User Service Handle	M	
Result (+)		S
User Service Handle		M
Data from Communication Object		M
Result Information		I
Result (–)		S
User Service Handle		M
Result Error		S
Invocation Error		S

6.3.14.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.14.2.1.1 Параметр Virtual Device Handle (Дескриптор виртуального устройства)

Данный параметр идентифицирует виртуальное устройство с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.3).

6.3.14.2.1.2 Параметр Function Object Handle (Дескриптор функционального объекта)

Данный параметр позволяет идентифицировать функциональный объект с использованием дескриптора, определенного в VDSI-интерфейсе с помощью сервиса Instantiate Virtual Device (см. 6.3.8).

6.3.14.2.1.3 Параметр Communication Object Identifier (Идентификатор объекта связи)

Данный параметр позволяет идентифицировать объект связи, определенный в описании функциональных возможностей устройства согласно ИСО 20242 (часть 4).

6.3.14.2.1.4 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является определяемым пользователем идентификатором для указанного сервиса.

6.3.14.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.14.2.2.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.14.2.2.2 Параметр Data from Communication Object (Данные от объекта связи)

Указанный параметр содержит данные пользователя, которые считываются из объекта связи. Их тип задается в описании функциональных возможностей устройства для VDSI-интерфейса согласно ИСО 20242 (часть 4).

6.3.14.2.2.3 Параметр Result Information (Полученная информация)

Данный параметр является опцией реализации и подробно определяется в 8.1. Он может давать дополнительную информацию относительно выполнения сервиса.

6.3.14.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.14.2.3.1 Параметр User Service Handle (Дескриптор сервиса пользователя)

Данный параметр является копией задаваемого пользователем идентификатора указанного сервиса, предоставляемого вместе с запросом на него.

6.3.14.2.3.2 Параметр Result Error (Полученная ошибка)

Выбор подобного типа параметра, который подробно определен в 8.2, указывает на причину неудачи запроса на сервис.

6.3.14.2.3.3 Параметр Invocation Error (Ошибка вызова)

Выбор подобного типа параметра должен показывать следующие виды ошибок:

- недопустимый дескриптор виртуального устройства;
- недопустимый дескриптор функционального объекта;
- недопустимый идентификатор объекта связи;
- иные.

6.3.14.3 Процедура выполнения сервиса

Если дескриптор виртуального устройства и функционального объекта, а также идентификатор объекта связи действующие, то данные будут считываться. Если этот объект находится внутри физического устройства, то будут выполняться все необходимые связи для выбора данных из этого устройства через RMSI-интерфейс.

6.3.15 Сервис Report Data to Application (Сообщение данных в приложение)

6.3.15.1 Краткое описание сервиса

Указанный сервис позволяет обеспечить пользователя объектом связи для передачи ему данных VDSI-интерфейса.

6.3.15.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 19.

Таблица 19 — Параметры сервиса Report Data to Application

Наименование параметра	Ind	Rsp
Argument	M	
User Object Identifier	M	
Data	M	
Result (+)		S
Result (–)		S
Error		M

6.3.15.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.15.2.1.1 Параметр User Object Identifier (Идентификатор объекта пользователя)

Данный параметр позволяет идентифицировать объект связи для приложения. Этот идентификатор определяется с помощью сервиса Instantiate Communication Object (см. 6.3.11).

6.3.15.2.1.2 Параметр Data (Данные)

Указанный параметр определяет данные, передаваемые пользователю VDSI-интерфейса. Тип этих данных приведен в описании функциональных возможностей устройства для VDSI-интерфейса.

6.3.15.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.15.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.15.2.3.1 Параметр Error (Ошибка)

Данный параметр должен указывать одну из ошибок следующего вида:

- недопустимый идентификатор объекта пользователя;
- доступ к данным временно невозможен;
- иные.

6.3.15.3 Процедура выполнения сервиса

Если идентификатор указанного объекта действующий, а доступ к данным возможен, то пользователь VDSI-интерфейса должен иметь доступ к ним; в противном случае ошибка будет возвращаться.

6.3.16 Сервис Request Data from Application (Запрос данных из приложения)

6.3.16.1 Краткое описание сервиса

Указанный сервис используется объектом связи для запроса данных от пользователя VDSI-интерфейса.

6.3.16.2 Параметры сервиса

Параметры данного сервиса указаны в таблице 20.

Таблица 20 — Параметры сервиса Request Data from Application

Наименование параметра	Ind	Rsp
Argument	M	
User Object Identifier	M	
Result (+)		S
Data		M
Result (–)		S
Error		M

6.3.16.2.1 Параметр Argument (Аргумент)

Argument содержит параметры запроса на сервис.

6.3.16.2.1.1 Параметр User Object Identifier (Идентификатор объекта пользователя)

Данный параметр позволяет идентифицировать объект связи для приложения. Этот идентификатор определяется с помощью сервиса Instantiate Communication Object (см. 6.3.11).

6.3.16.2.2 Параметр Result (+) (Положительный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел успешно.

6.3.16.2.2.1 Параметр Data (Данные)

Указанный параметр определяет передачу данных на объект связи. Их тип приведен в описании функциональных возможностей устройства для VDSI-интерфейса.

6.3.16.2.3 Параметр Result (–) (Отрицательный результат)

Выбор подобного типа параметра показывает, что запрос на сервис прошел неудачно.

6.3.16.2.3.1 Параметр Error (Ошибка)

Данный параметр должен указывать одну из ошибок следующего вида:

- недопустимый идентификатор объекта пользователя;
- доступ к данным временно невозможен;
- иные.

6.3.16.3 Процедура выполнения сервиса

Если идентификатор указанного объекта действующий, а доступ к данным возможен, то пользователь VDSI-интерфейса будет посылать в ответ данные; в противном случае будет возвращаться ошибка.

7 Рабочие состояния виртуального устройства

7.1 Контрольное VD-устройство

7.1.1 Общие сведения

Виртуальные устройства создаются для специализированных применений. Инстанциация (создание экземпляров) требуемых устройств, их функциональных объектов и объектов связи (вместе с записями исходных данных в объекты связи) называется «конфигурацией». Рабочие состояния виртуальных устройств ориентированы на процедуры конфигурирования и не являются предметом внутренних событий, однако находятся под контролем пользователя VDSI-интерфейса, для чего определяется специальное устройство для локального контроля — «контрольное виртуальное устройство». Существуют обязательные функциональные объекты, связанные с ним.

7.1.2 Базовый функциональный объект устройства Device Base Function Object

Данный функциональный объект является лишь одним из экземпляров реализации контрольного VD-устройства, который дает экземпляр VDSI-интерфейса в качестве рабочих выходных данных (см. 6.3.10).

7.1.3 Функциональный объект перехода Transition Function Object

7.1.3.1 Общие сведения

Данный функциональный объект имеет лишь один экземпляр в контрольном VD-устройстве и содержит операции переключения между рабочими состояниями рассматриваемого виртуального устройства, дескриптор которого является рабочим входным параметром (см. 6.3.10). Если переход к новому состоянию невозможен, то соответствующий параметр Result Error предоставляется сервисом Execute Operation.

7.1.3.2 Операция StartDefinition

При выполнении этой операции рассматриваемое виртуальное устройство будет переключаться между состояниями Initialized и Preparation (см. 7.2).

7.1.3.3 Операция EndDefinition

При выполнении этой операции рассматриваемое виртуальное устройство будет переключаться между состояниями Preparation и Check (см. 7.2).

7.1.3.4 Операция StartWorking

При выполнении этой операции рассматриваемое виртуальное устройство будет переключаться между состояниями Check (или Revise) и Working (см. 7.2).

7.1.3.5 Операция AddDefinition

При выполнении этой операции рассматриваемое виртуальное устройство будет переключаться между состояниями Working и Revise (см. 7.2).

7.1.3.6 Операция End Working (Конец работы)

При выполнении этой операции рассматриваемое виртуальное устройство будет переключаться между состояниями Working (или Check) и Evaluation (см. 7.2).

7.1.3.7 Операция ChangeDefinition

При выполнении этой операции рассматриваемое виртуальное устройство будет переключаться между состояниями Evaluation и Preparation (см. 7.2).

7.1.3.8 Операция ClearAllObjects

При выполнении этой операции рассматриваемое виртуальное устройство будет переключаться между состояниями Evaluation и Initialized (см. 7.2). При этом все объекты связи и функциональные объекты удаляются, а виртуальное устройство становится «пустым».

7.2 Рабочие состояния виртуальных устройств

7.2.1 Общие сведения

Возможные рабочие состояния виртуальных устройств иллюстрируются рисунком 6. Переходы между различными состояниями осуществляются с использованием операций сервиса функционального объекта Transition (см. 7.1.3) или с помощью сервисов, указанных в таблице 21.

Таблица 21 — Переходы, вызываемые запросом на сервис

Переход	Связанный сервис
Initiate	Instantiate Virtual Device
Conclude	Remove Virtual Device
Abort	Destroy Virtual Device

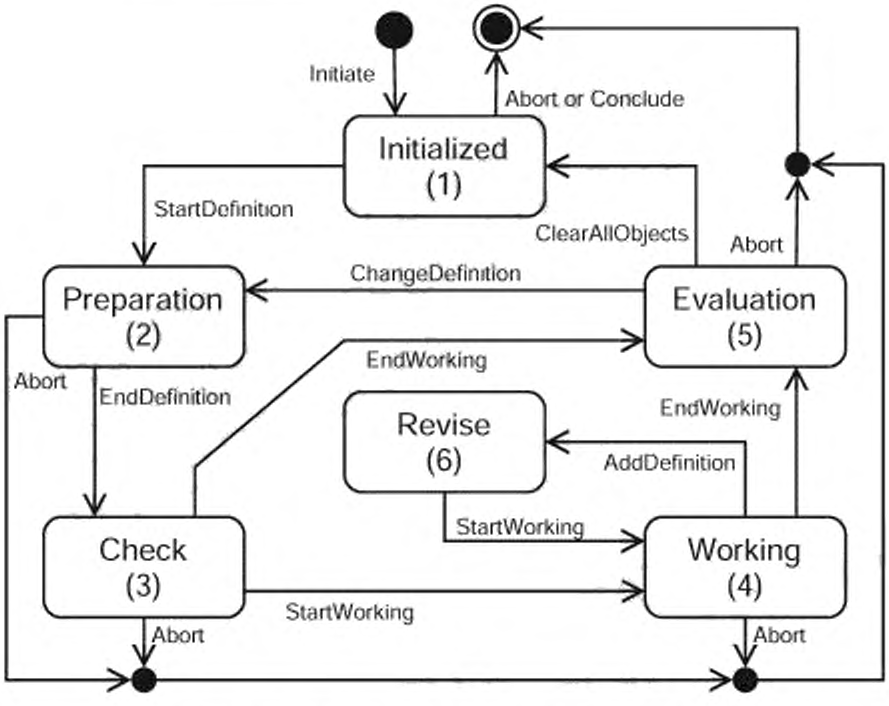


Рисунок 6 — Рабочие состояния виртуальных устройств

7.2.2 Состояние Initialized (Инициализированное)

Данное состояние предназначено для простого тестирования виртуального устройства. VDSI-интерфейс может проверять возможность применения требуемых фундаментальных ресурсов, например с помощью инициализации связи с соответствующими физическими устройствами.

Состояние Initialized достигается созданием виртуального устройства или выполнением операции ClearAllObjects в функциональном объекте Transition контрольного VD-устройства. В состоянии Initialized допустимы только сервисы для обработки самого виртуального устройства.

Таблица 22 — Свойства состояния Initialized

Переходы при вводе	Переходы при выводе	Допустимые сервисы
Initiate ClearAllObjects	Conclude Abort StartDefinition	Remove Virtual Device Destroy Virtual Device Get Virtual Device Status Identify Virtual Device

7.2.3 Состояние Preparation (Подготовка)

Данное состояние предназначено для конфигурирования, т.е. формирования виртуального устройства вместе с его содержимым, отвечающего требованиям приложения. При этом все необходимые функциональные возможности уже установлены и параметризированы.

Состояние Preparation достигается выполнением операции StartDefinition или ChangeDefinition в функциональном объекте Transition контрольного VD-устройства. В этом состоянии допустимы все сервисы, запрашиваемые VDSI-интерфейсом пользователя данного виртуального устройства. Локальные события VDS-сервиса не выполняются.

Таблица 23 — Свойства состояния Preparation

Переходы при вводе	Переходы при выводе	Допустимые сервисы
StartDefinition ChangeDefinition	Abort EndDefinition	Все сервисы для данного виртуального устройства, за исключением: - Remove Virtual Device - Report Data to Application - Request Data from Application

7.2.4 Состояние Check (Проверка)

Данное состояние предназначено для полного тестирования виртуального устройства VDSI-интерфейсом без помех со стороны его пользователя. VDSI-интерфейс проверяет адекватность конфигурации и возможности использования всех необходимых ресурсов, например тестированием связанных физических устройств. Фактическое состояние процедуры проверки можно исследовать с помощью сервиса Get Virtual Device Status.

Состояние Check достигается выполнением операции EndDefinition в функциональном объекте Transition контрольного VD-устройства. В этом состоянии допустимы только сервисы обработки виртуального устройства.

Таблица 24 — Свойства состояния Check

Переходы при вводе	Переходы при выводе	Допустимые сервисы
EndDefinition	Abort StartWorking EndWorking	Destroy Virtual Device Get Virtual Device Status Identify Virtual Device

7.2.5 Состояние Working (В работе)

Данное состояние предназначено для запуска приложения, после окончания конфигурирования и при наличии всех ресурсов. В таком состоянии нет возможности вносить изменения в конфигурацию. Объекты связи, отмеченные как параметры в описании функциональных возможностей устройства (подробнее об этом см. ИСО 20242, часть 4), не могут записываться.

Состояние Working достигается выполнением операции StartWorking в функциональном объекте Transition в контрольном VD-устройстве. В состоянии Check допустимы все сервисы, связанные с данным виртуальным устройством и его содержанием, за исключением тех, которые изменяют конфигурацию. Локальные события VDS-сервиса будут выполняться.

Таблица 25 — Свойства состояния Working

Переходы при вводе	Переходы при выводе	Допустимые сервисы
StartDefinition ChangeDefinition	Abort EndWorking AddDefinition	Все сервисы для данного виртуального устройства, за исключением: - Remove Virtual Device - Instantiate Function Object - Remove Function Object - Instantiate Communication Object - Remove Communication Object

7.2.6 Состояние Revise (Проверка)

Данное дополнительное состояние предназначено для оптимизации конфигурации при выполнении приложения, однако при этом допустимо внесение незначительных изменений добавлением или удалением объектов связи и значений записанных параметров, но нельзя изменить возможности виртуального устройства добавлением или удалением функциональных объектов.

Состояние Revise достигается выполнением операции AddDefinition в функциональном объекте Transition контрольного VD-устройства. В этом состоянии допустимы все сервисы, связанные с данным виртуальным устройством и его содержимым, за исключением тех, которые изменяют возможности виртуального устройства. Локальные события VDS-сервиса будут выполняться.

Таблица 26 — Свойства состояния Revise

Переходы при вводе	Переходы при выводе	Допустимые сервисы
AddDefinition	Abort Start Working	Все сервисы для данного виртуального устройства, за исключением: - Remove Virtual Device - Instantiate Function Object - Remove Function Object

7.2.7 Состояние Evaluation (Оценка)

Данное состояние предназначено для реализации конфигурации, используемой в состоянии Working. При этом виртуальное устройство готовится для новой конфигурации. Будут удаляться только те объекты, которые не потребуются в последующих приложениях.

Состояние Evaluation достигается выполнением операции EndWorking в функциональном объекте Transition контрольного VD-устройства. В этом состоянии допустимы только сервисы, предназначенные для удаления объектов, а также для обработки виртуального устройства.

Таблица 27 — Свойства состояния Evaluation

Переходы при вводе	Переходы при выводе	Допустимые сервисы
EndWorking	Abort ChangeDefinition ClearAllObjects	Destroy Virtual Device Get Virtual Device Status Identify Virtual Device Remove Function Object Remove Communication Object

8 Результаты выполнения сервисов

8.1 Дополнительная информация

8.1.1 Структура полученной информации

При успешном выполнении сервиса появляется подтверждение типа Result (+), в котором во многих случаях будут содержаться полученные данные — дополнительная информация, предоставляемая вместе с подтверждением в структуре Result Information (см. таблицу 28).

Таблица 28 — Структура получаемой информации

Типы получаемой информации	Cnf
Info Group	M
Info Grade	M
Info Code	M
Info Description	M

Типы Info Group, Info Grade и Info Code выражаются целыми числами, а Info Description — текстовым описанием.

8.1.2 Типы получаемой информации

Различные типы информации, предоставляемые вместе с успешно выполненным сервисом, приведены в таблице 29.

Таблица 29 — Типы получаемой информации

Info Group	Info Grade	Info Code	Info Description	Примечание
0	0	0	Пустой	Никакой дополнительной информации не предоставляется
0	1	Локально задаваемый	Локально задаваемый	Предупреждения
0	2	Локально задаваемый	Локально задаваемый	Информация о VDSI-объекте
Не равен 0	Не равен 0	Локально задаваемый	Локально задаваемый	Специфическая для реализации информация (пример см. в приложении А)

8.2 Ошибки выполнения сервисов

8.2.1 Ошибки обращения Invocation Errors

Ошибки обращения относятся к внутреннему состоянию запрашиваемого сервиса VDSI-объекта при отсутствии какого бы то ни было внешнего сервиса, вводимого с помощью служебного интерфейса управления ресурсами (RMSI). Эти ошибки описываются параметром Invocation Error в показателе Result (–) при описаниях сервисов (см. 6.3).

8.2.2 Структура получаемых ошибок Structure Result Error

Если сервис выполняется неудачно (включая все внешние сервисы, вводимые с помощью RMSI-интерфейса), то полученная ошибка появляется вновь, являясь типовой структурой, элементы которой описаны в таблице 30.

Таблица 30 — Структура получаемых ошибок

Тип получаемой ошибки	Cnf
Error Group (группа ошибки)	M
Error Grade (класс ошибки)	M
Error Code (код ошибки)	M
Error Description (описание ошибки)	M

Типы Error Group, Error Grade и Error Code выражаются целыми числами, а Error Description — текстовым описанием.

Тип получаемой ошибки Error Group должен обеспечивать ее классификацию по следующим группам:

- Periphery Error Group (периферийных устройств);
- Execution Error Group (выполнения);
- Access Error Group (доступа);
- Application Error Group (приложения);

- GDI/DIP error Group (GDI/DIP);
- MICX error Group (MICX);
- иные.

8.2.3 Группа Periphery Error Group

Группа ошибок периферийных устройств содержит ошибки, обусловленные связью с физическими устройствами через RMSI-интерфейс и определенные в категории Error Grade (см. таблицу 31). Типу Error Code всегда присваивается 0, а тип Error Description можно использовать для более подробного анализа ошибки.

Таблица 31 — Категории ошибок периферийных устройств

Обозначение ошибки	Описание ошибок
Per_1	Соединение с физическим устройством нарушено, передача данных невозможна
Per_2	Подтверждение от RMSI-интерфейса непригодно
Per_3	Получение неизвестных данных от RMSI-сервиса
Per_4	Обработка пользователем данных с подтверждением от RMSI-интерфейса недопустима
Per_5	Открытие периферийного интерфейса невозможно
Per_6	Отбраковка данных, посылаемых через RMSI-интерфейс с использованием сервиса Write Data
Per_7	Отбраковка данных, получаемых через RMSI-интерфейс с применением сервиса Read Data
Per_8	Неудачное выполнение сервис-операций RMSI-интерфейсом
Per_9	Любые другие ошибки

8.2.4 Группа Execution Error Group

8.2.4.1 Общие сведения

Эта группа содержит ошибки, обусловленные выполнением сервиса VDSI-интерфейсом, тип Error Grade — классы ошибок, указанные в таблице 31, тогда как тип Error Code позволяет идентифицировать ошибки непосредственно. Тип Error Description можно использовать для их более подробного анализа.

Таблица 32 — Ошибки выполнения

Обозначение типа ошибки	Описание ошибок
VDstate	Связанные с состоянием виртуального устройства
AppRef	Связанные с приложением
Definition	Обусловленные определением объектов
Resource	Связанные с конфигурациями и ресурсами
Preemptive	Обусловленные блокировкой и временным контролем
Access	Связанные с доступом к данным и изменением состояния
Remove	Связанные с удалением объектов
Cancel	Обусловленные отменой сервиса

8.2.4.2 Ошибки типа VDstate

Тип ошибок кода (Error Code) должен характеризовать их по следующим признакам:
- выполнение сервиса невозможно в данном рабочем состоянии;
- иные.

8.2.4.3 Ошибки типа AppRef

Тип Error Code должен характеризовать подобные ошибки по следующим признакам:
- ошибка сигнализации, программно-контролируемые ресурсы заблокированы;
- иные.

8.2.4.4 Ошибки типа Definition

Тип Error Code должен характеризовать подобные ошибки по следующим признакам:

- идентификатор шаблона виртуального устройства недействителен;
- идентификатор шаблона функционального объекта недействителен;
- идентификатор объекта связи недействителен;
- данные недействительны;
- идентификатор объекта связи находится в употреблении;
- конкретный вариант объекта связи забракован из-за несоответствия конфигурации;
- иные.

8.2.4.5 Ошибки типа Resource

Тип Error Code должен характеризовать подобные ошибки по следующим признакам:

- проблемы с распределением ресурсов памяти;
- проблемы с временем обработки;
- исчерпано число допустимых вариантов;
- неправильное конфигурирование, рабочее состояние невозможно;
- проверка конфигурации, рабочее состояние в данный момент невозможно;
- функциональный объект Control VD не удаляется из-за наличия другого VD;
- иные.

8.2.4.6 Ошибки типа Preemptive

Тип Error Code должен характеризовать подобные ошибки по следующим признакам:

- истекло время исполнения данного сервиса;
- выявлена блокировка при исполнении данного сервиса;
- иные.

8.2.4.7 Ошибки типа Access

Тип Error Code должен характеризовать подобные ошибки по следующим признакам:

- неправильная обработка виртуального устройства;
- неправильная обработка функционального объекта;
- отсутствие объекта связи, к которому должен быть обеспечен доступ в данном функциональном объекте;
- отсутствие операции, к которой должен быть обеспечен доступ в данном функциональном объекте;
- невозможность доступа к режиму записи из-за рабочего состояния объекта и наличия режима только считывания данных;
- нарушение диапазона данных или рассогласование доступа к указателю;
- невозможность изменения рабочего состояния;
- неисправность в аппаратной части связанного устройства;
- иные.

8.2.4.8 Ошибки типа Remove

Тип Error Code должен характеризовать подобные ошибки по следующим признакам:

- невозможность удаления объекта из-за открытости другого сервиса для него;
- невозможность удаления Control VD из-за наличия другого VD;
- иные.

8.2.4.9 Ошибки типа Cancel

Тип Error Code должен характеризовать подобные ошибки по следующим признакам:

- обработка сервиса неизвестным пользователем;
- невозможность отмены сервиса в данный момент;
- иные.

8.2.5 Группа Application Error Group

Ошибки в приложении (применения) распространяются на специфические для реализации VDSI-интерфейса ошибки и связаны с многоязычными текстовыми элементами, определяемыми согласно ИСО 20242 (часть 4), для чего используется описанный в нем XML-элемент SubAreaError.

Эта группа ошибок включает номер области текста, задаваемый в многоязычных текстовых элементах согласно ИСО 20242 (часть 4). Тип Error Code содержит номер текста в указанной области и

тип Error Description — перечень текстовых элементов, которые могут вводиться в заданные метки-заполнители этих связанных элементов.

8.2.6 Группа GDI/DIP Error Group

Для облегчения доступа автоматизированных систем к драйверам устройств ASAM GDI их интерфейс определяет не зависящий от конкретного устройства профиль шаблона (DIP), который находится вне области рассмотрения ИСО 20242, однако сообщения об ошибках должны соответствовать этому стандарту и принадлежать данной группе ошибок.

8.2.7 Группа MICX Error Group

В приложении В ИСО 20242 (часть 4) описан профиль шаблона для MICX-возможностей устройства, зависящий от принятой методики. Если соответственно элемент VDSI-интерфейса требует специальных сообщений об ошибках, то они должны принадлежать их данной группе.

Приложение А
(справочное)

Рекомендации по реализации VDSI-интерфейса

А.1 Конфигурирование сервисов для функциональных вызовов C/C++ function calls

А.1.1 Объект приведения в соответствие

Использование данных рекомендаций должно гарантировать, что реализации VDSI-интерфейса для одной и той же операционной системы будут совместимыми с точки зрения их интерфейсов и процедур обслуживания, а также с реализациями драйверов устройств ASAM GDI версии 4.4. Наименования функций, описываемых в данном приложении, аналогичны применяемым в спецификации на ASAM GDI.

А.1.2 Использование правил конфигурирования

Правила конфигурирования несущественны для создания полноценной реализации VDSI-интерфейса. Необходимо принять во внимание описания сервисов, приведенные в 6.3 данной части стандарта.

А.1.3 Стандарты С и С++

Язык программирования С стандартизирован в ИСО/МЭК 9899:1990 (С90) и его новой редакции ИСО/МЭК 9899:1999 (С99), а язык программирования С++ — в ИСО/МЭК 14882:1998.

А.1.4 Соглашение относительно типов простых данных

Если не определено иное, то типы данных в языках C/C++ , применяемые в данном приложении, относятся к 32-битовой разрядности и поэтому в большинстве случаев не зависят от конкретной операционной системы. При использовании иной вычислительной среды (например, с разрядностью 64 бит) некоторые простые типы данных могут стать системно зависимыми. В этих случаях реализация должна документироваться в соответствии с конкретными спецификациями ASAM GDI.

Строки — это поля, содержащие значения из 8 бит (октеты) без нуля, который должен отмечать конец строки.

А.1.5 Специальные типы простых данных

А.1.5.1 Не зависящие от используемой операционной системы простые типы данных

Для параметров сервис-примитивов должны применяться не зависящие от используемой операционной системы специальные простые типы данных.

Таблица А.1 — Не зависящие от используемой операционной системы простые типы данных

Тип данных	C/C++-определение	Диапазон значений
APICHAR	Signed char (Число со знаком)	От –128 до 127
APIBYTE	Unsigned char (Число без знака)	От 0 до 2^8
APIRET	Signed short (Укороченное число со знаком)	От –65536 до 65535
APIHND	Unsigned long (Расширенное число без знака)	От 0 до 2^{32}

А.1.5.2 Зависящие от используемой операционной системы простые типы данных

Для параметров сервис-примитивов должны использоваться зависящие от используемой операционной системы специальные простые типы данных.

Таблица А.2 — Зависящие от используемой операционной системы простые типы данных

Тип данных	WIN32 С-определение	LINUX С-определение
GDI_CALL	_stdcall	Отсутствие специального функционального типа данных
GDI_CB	_cdecl	Отсутствие специального функционального типа данных

А.1.6 Специальные типы сложных данных

А.1.6.1 Соглашение относительно сложных типов данных

Сложные типы данных должны по умолчанию быть сопряженными с 8-битовой размерностью; в противном случае это необходимо отмечать при поставке драйвера.

А.1.6.2 С-структуры данных

Для параметров сервис-примитивов должны использоваться не зависящие от операционной системы структуры данных.

Таблица А.3 — Не зависящие от операционной системы структуры данных

Тип данных	C/C++-определение	Примечания

GDIRESULT	struct
	{
	short qual;
	short grade;
	short code;
	void *addinfo;
	};
		Данный тип данных описывает результат функционального вызова в случае ошибок, предупреждающих сообщений или любой другой информации
		
		Для ознакомления с описаниями элементов см. раздел 8 (таблицы 28 и 30)

GDIIDENT	struct
	{
	unsigned long deviceVersion;
	char *driverName;
	unsigned long driverVersion;
	char*vendor;
	};
		deviceVersion:
		Это число служит для указания экземпляров виртуальных устройств и, если возможно, — экземпляров связанных с ними физических устройств
		
		driverName:
		Обозначение драйвера устройства или его части, к которому принадлежит данное VD-устройство
		driverVersion:
		Это число служит для указания экземпляра исполнения драйвера в целом и может проверяться поставщиком на соответствие
		Обозначение для изготовителя (разработчика) данного виртуального устройства
		
		Примечание — Используемые вместе с Control VD элементы рассматривают драйвер устройства в целом относительно информации, которая применима для всех типов виртуальных устройств в драйвере

GDISTATUS	struct
	{
	short log;
	short phys;
	short phase;
	GDIRESULT detail;
	};
		Данный тип данных описывает состояние виртуального устройства и неприменим к контрольному VD-устройству
		
		Подробнее см. описание параметров сервис-примитивов в 6.3.6.

А.1.7 Соглашения относительно предварительно определенных постоянных

Если в наименованиях применяются постоянные, то их необходимо указывать в заголовках файлов, которые определены для ASAM GDI-интерфейса в соответствующих спецификациях. Поскольку содержание этих заголовков может расширяться в последующих реализациях, а сочетания драйверов с различными версиями и устройствами сопряжения платформ также могут обрабатываться в файлах заголовков, то они не будут рассматриваться в настоящем стандарте.

Предварительно определяемые постоянные, которые используются в данном приложении, указаны в таблице А.4.

Таблица А.4 — Предварительно определяемые постоянные

Наименование постоянной	Значение	Описание
SYNC	(APIHND) 0	Идентификатор для синхронного функционального вызова

Примечание — Постоянная SYNC применяется вместо задаваемого пользователем дескриптора для асинхронного функционального вызова. Асинхронная связь запрашивается с помощью значения, не эквивалентного SYNC. Синоним постоянной ASYNC используется для данного дескриптора в функциональных описаниях ниже. При асинхронной связи дескриптор является параметром подтверждения с помощью обратного (повторного) вызова.

А.1.8 Соглашения относительно прототипов функций

Прототипы функций описываются как:

returnType callType functionName (list of argument types).

В приведенных описаниях значения типа returnType называются «возвращаемыми». Аргументы обозначаются как arg и нумеруются начиная с 1, например arg1, arg2 и т. д.

А.1.9 Возвращаемые значения

Указанные числа используются для возвращаемых значений функций:

Таблица А.5 — Числа для возвращаемых значений

Число	Наименование	Описание
< –1	Invocation Error (Ошибка обращения)	Функция не может правильно выполняться, возвращаемое значение описывает ошибку в соответствии с перечнем ошибок обращения. Элементы структуры GDIRESULT устанавливаются на нуль
–1	COM_ERR	Функция не может правильно выполняться, ошибка описывается в структуре VDSIRESULT
0	COM_FIN	Функция успешно завершена в синхронном режиме работы
1	COM_BUSY	Функция активирована в асинхронном режиме работы

А.1.10 Значения для ошибок обращения

Данные отрицательные значения используются для указания ошибок в возвращаемом значении функции:

Таблица А.6 — Значения для указания ошибок

Значение для ошибки	Описание
– 2	Функция VDSI_Attach запрошена, хотя VDSI-интерфейс уже открыт
– 3	Запрос на сервис возникает до того, как VDSI-интерфейс открыт с помощью функции VDSI_Attach
– 4 ... –8	Зарезервированы для последующих применений
– 9	Отсутствие ресурсов для выполнения последующих связей в асинхронном режиме
– 10, –11	Зарезервированы для последующих применений
– 12	Асинхронный вызов функции не поддерживается
– 13	VDSI-интерфейс не поддерживает созданные экземпляры в рассматриваемом классе
– 14	Зарезервировано для последующих применений
– 15	Функция не может выполняться из-за нарушения последовательности или неверно выбранных параметров

А.1.11 Сервисы базового управления

Функции сервисов базового управления указаны в таблице А.7.

Таблица А.7 — Сервисы базового управления

Сервис	Прототип функции	Связь со служебными параметрами
Attach VDSI Entity (Прикрепление VDSI-объекта)	short GDI_CALL GDI Attach (GDI_CB, GDI_CB, GDI_CB)	arg1: Указатель функции, вызываемый провайдером VDSI-интерфейса для подтверждения arg2: Указатель функции, вызываемый провайдером VDSI-интерфейса для индикации сервиса VDSI_InfReport arg3: Указатель функции, вызываемый провайдером VDSI-интерфейса для индикации сервиса VDSI_Accept return: COM_FIN, если VDSI-интерфейс может использоваться; в противном случае выдается сообщение об ошибке (см. таблицу А.6)

Окончание таблицы А.7

Сервис	Прототип функции	Связь со служебными параметрами
Cancel Service (Отмена сервиса)	short GDI CALL GDI Cancel (APIHND, APIHND, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: SYNC или ASYNC (см. примечание к таблице А.4) arg3: Определяемый пользователем VDSI-интерфейса дескриптор сервиса, который должен быть отменен arg4: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно этого сервиса return: (см. таблицу А.5)

А.1.12 Сервисы обработки виртуального устройства

Функции для сервисов обработки виртуального устройства указаны в таблице А.8.

Таблица А.8 — Сервисы виртуального устройства

Сервис	Прототип функции	Связь со служебными параметрами
Instantiate Virtual Device (Реализация виртуального устройства)	short GDI_CALL GDI_Initiate (APIHND, APIHND*, void *, APIHND, GDIRESULT*)	arg1: Идентификационный номер для рассматриваемого типа виртуального устройства, определяемый в описании его функциональных возможностей arg2: Указатель на хранение для провайдера VDSI-интерфейса, определяющий дескриптор данного экземпляра виртуального устройства arg3: Указатель на структуру, содержащую созданные параметры для виртуального устройства в описании его функциональных возможностей arg4: SYNC или ASYNC (см. примечание к таблице А.4) arg5: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)
Conclude Virtual Device (Удаление виртуального устройства)	short GDI_CALL GDI_Conclude (APIHND, APIHND, GDIRESULT *)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: SYNC или ASYNC (см. примечание в А.7) arg3: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)
Abort Virtual Device (Нарушение виртуального устройства)	short GDI CALL GDI Abort (APIHND)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса VDSI_Initiate return: COM_FIN, если VD-устройство разрушено; в противном случае возникает ошибка (см. таблицу А.6)
Get Virtual Device Status (Получение состояния виртуального устройства)	short GDI_CALL GDI_Status (APIHND, GDISTATUS*, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Указатель на структуру GDISTATUS для информации относительно состояния VD-устройства arg3: SYNC или ASYNC (см. примечание к таблице А.4) arg4: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)

Окончание таблицы А.8

Сервис	Прототип функции	Связь со служебными параметрами
Identify Virtual Device (Идентификация виртуального объекта)	short GDI CALL GDI Identify (APIHND, GDIIDENT*, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Указатель на структуру GDIIDENT для идентификационной информации относительно VD-устройства arg3: SYNC или ASYNC (см. примечание к таблице А.4) arg4: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)

А.1.13 Сервисы обработки функциональных объектов

Функции сервисов обработки функциональных объектов указаны в таблице А.9.

Таблица А.9 — Сервисы функционального объекта

Сервис	Прототип функции	Связь со служебными параметрами
Instantiate Function Object (Реализация функционального объекта)	short GDI_CALL GDI_CreateFuncObject (APIHND,APIHND, void*, APIHND*, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Идентификационный номер для рассматриваемого типа функционального объекта, определяемый в описании функциональных возможностей устройства arg3: Указатель на структуру, содержащую созданные параметры для функционального объекта и указанную в описании функциональных возможностей устройства arg4: Указатель на хранение для провайдера VDSI-интерфейса, определяющий дескриптор данного экземпляра функционального объекта arg5: SYNC или ASYNC (см. примечание к таблице А.4) arg6: GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)
Remove Function Object (Удаление функционального объекта)	short GDI_CALL GDI_DeleteFuncObject (APIHND, APIHND, APIHND,GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Дескриптор рассматриваемого FO, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg3: SYNC или ASYNC (см. примечание к таблице А.4) arg4: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)
Execute Operation (Выполнение операции)	short GDI CALL GDI Execute (APIHND, APIHND, APIHND, void*, void *, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Дескриптор рассматриваемого функционального объекта, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Function Object arg3: Идентификационный номер для рассматриваемой операции, определяемый в описании функциональных возможностей устройства arg4: Указатель на структуру, содержащую входные параметры для операции, рассматриваемой в описании функциональных возможностей устройства; должен быть равным NULL при отсутствии входных параметров

Окончание таблицы А.9

Сервис	Прототип функции	Связь со служебными параметрами
		arg5: Указатель на структуру, содержащую выходные параметры для операции, рассматриваемой в описании функциональных возможностей устройства; должен быть равным NULL при отсутствии выходных параметров arg6: SYNC или ASYNC (см. примечание к таблице А.4) arg7: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)

А.1.14 Сервисы обработки объекта связи

А.1.14.1 Функции, связанные с описаниями сервисов в 6.3.

Функции для сервисов обработки объекта связи указаны в таблице А.10.

Таблица А.10 — Сервисы объектов связи

Сервис	Прототип функции	Связь со служебными параметрами
Instantiate Communication Object (Реализация объекта связи)	short GDI_CALL GDI_CreateCommObject (APIHND, APIHND, APIHND, APIHND, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Дескриптор рассматриваемого FO, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg3: Идентификационный номер для рассматриваемого объекта связи, равный его положению в перечне функциональных объектов (начиная с 1) arg4: Задаваемый пользователем глобальный идентификатор объекта для незапрашиваемых сервисов, начиная с данного объекта arg5: SYNC или ASYNC (см. примечание к таблице А.4) arg6: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)
Remove Communication Object (Удаление объекта связи)	short GDI_CALL GDI_DeleteCommObject (APIHND, APIHND, APIHND, APIHND*, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Дескриптор рассматриваемого функционального объекта, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg3: Идентификационный номер для рассматриваемого объекта связи arg4: Указатель на хранилище пользователя для глобального идентификатора объекта, предоставляемого с сервисом Create Communication Object, позволяющим записывать идентификатор в хранилище arg5: SYNC или ASYNC (см. примечание к таблице А.4) arg6: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)
Write Data to Communication Object (Запись данных в объект связи)	short GDI_CALLGDI_Write (APIHND,APIHND, APIHND, void*, APIHND, GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Дескриптор рассматриваемого функционального объекта, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg3: Идентификационный номер для рассматриваемого объекта связи

Окончание таблицы А.10

Сервис	Прототип функции	Связь со служебными параметрами
		arg4: Указатель на хранилище пользователя для данных этого объекта связи arg5: SYNC или ASYNC (см. примечание к таблице А.4) arg6: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: (см. таблицу А.5)
Read Data from Communication Object (Считывание данных из объекта связи)	short GDI CALL GDI Read (APIHND, APIHND, APIHND, void *, APIHND,GDIRESULT*)	arg1: Дескриптор рассматриваемого VD-устройства, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg2: Дескриптор рассматриваемого функционального объекта, определяемый провайдером VDSI-интерфейса для сервиса Instantiate Virtual Device arg3: Идентификационный номер для рассматриваемого объекта связи arg4: Указатель на хранилище пользователя для получаемых данных данного объекта связи arg5: SYNC или ASYNC (см. примечание к таблице А.4) arg6: Указатель на структуру GDIRESULT для сообщений об ошибках или другой информации относительно данного сервиса return: См. таблицу А.5
Report Data to Application (Сообщение данных в приложение)	short GDI_CB GDI_InfReport (APIHND, void*)	arg1: Определяемый пользователем глобальный идентификатор объекта для незапрашиваемых сервисов, начиная с данного объекта, закрепленный за сервисом Instantiate Communication Object arg2: Указатель на данные для приложения return: COM_FIN, если передача данных была успешной: – 1, если глобальный идентификатор объекта не действителен; – 2, если передача данных временно невозможна.
Request Data from Application (Запрос данных из приложения)	short GDI_CB GDI_Accept (APIHND,void *)	arg1: Определяемый пользователем глобальный идентификатор объекта для незапрашиваемых сервисов, начиная с данного объекта, закрепленный за сервисом Instantiate Communication Object arg2: Указатель на данные из приложения return: COM_FIN, если передача данных была успешной, – 1, если глобальный идентификатор объекта не действителен; – 2, если передача данных временно невозможна

Функции GDI_InfReport и GDI_Accept должны быть предусмотрены пользователем VDSI-интерфейса при локальных событиях для передачи незапрашиваемых данных. Реализация на языках C/C++ должна использовать функции обратного (повторного) вызова для поставленной задачи. Адрес этих функций представлен в VDS-сервисе провайдера при вызове GDI_Attach.

А.1.14.2 Дополнительная функция для обработки асинхронной связи

Описание сервисов виртуального устройства (Virtual Device Services) в 6.3 данного стандарта является нейтральным по отношению к специальным сценариям типа связи в синхронном и асинхронном режимах. Реализация на языках C/C++ должна применять функции обратного (повторного) вызова для обработки асинхронных связей.

Пользователь VDSI-интерфейса обеспечивает функцию обратного вызова согласно таблице А.11, который должен запрашиваться провайдером RMS-сервиса в случае завершения процесса связи, запускаемого при вызове служебной функции. Адрес функции обратного вызова представлен в VDS-сервисе провайдера при вызове GDI_Attach.

Таблица А.11 — Дополнительная функция для связи в асинхронном режиме

Сервис	Прототип функции	Связь со служебными параметрами
Никакого точного сервиса, определенного для обработки данных в асинхронном режиме	short PA_CB io_complete (APIHND, IO_STAT *)	arg1: Идентификатор процесса связи, предоставляемый пользователем RMS-сервиса при вызове io_read, io_write или io_execute и обработке назад этим провайдером при вызове io_complete arg2: (см. таблицу А.3 для структуры данных) return: COM_FIN

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО 20242-1	IDT	ГОСТ Р ИСО 20242-1—2010 «Системы промышленной автоматизации и интеграция. Служебный интерфейс для испытательных прикладных программ. Часть 1. Общие положения»
ИСО 20242-2	—	*
ИСО/МЭК 10731	—	*
ИСО/МЭК 19501	—	*

* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта, который находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандарта:
- IDT — идентичный стандарт.

УДК 658.52.011.56 ОКС 25.040.40 Т58

Ключевые слова: автоматизированные промышленные системы, интеграция, жизненный цикл систем, управление производством

Редактор *К.Э. Маржа*
Технический редактор *В.Н. Прусакова*
Корректор *В.Е. Нестерова*
Компьютерная верстка *А.В. Бестужевой*

Сдано в набор 24.01.2014. Подписано в печать 17.02.2014 Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 5,12. Уч.-изд. л. 4,80. Тираж 62 экз. Зак. 236.

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru