

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

«СКАДИ»

(ООО «СКАДИ»)

УТВЕРЖДАЮ

**ПРОГРАММНАЯ ПЛАТФОРМА АСУ ТП И ДИАГНОСТИКИ
ДЛЯ СВБУ АЭС И ПРОМЫШЛЕННЫХ ПРИМЕНЕНИЙ**

Руководство системного программиста

34185873.425510.001.32.М

(На 43 листах)

2022

ООО «СКАДИ»	Программная платформа АСУ ТП и диагностики Руководство системного программиста	Версия 1
-------------	---	----------

РАЗРАБОТАЛ

34185873.425510.001.32.M	Стр. 2 из 43
--------------------------	--------------

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ О ПО.....	4
1.1 НАИМЕНОВАНИЕ.....	4
1.2 НАЗНАЧЕНИЕ И ФУНКЦИИ.....	4
1.3 НЕОБХОДИМЫЕ ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА	4
2 СТРУКТУРА ПО	5
2.1 Состав ПО	5
2.2 СВЯЗИ СОСТАВНЫХ ЧАСТЕЙ ПО.....	6
3 НАСТРОЙКА ПРОГРАММЫ.....	9
3.1 ИНСТАЛЛЯЦИЯ.....	9
3.1.1 Инсталляция для ОС Astra Linux, ОС Linux	9
3.1.2 Инсталляция для ОС Windows	10
3.2 НАСТРОЙКА ПРОЕКТА	10
3.2.1 Базовые файлы проекта.....	10
3.2.2 Группы сетевых узлов	13
4 ПРОВЕРКА ПРОГРАММЫ	15
4.1 Проверка наличия ПО и загрузки.....	15
4.2 Проверка загрузки глобалов общей памяти	16
5 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ	17
5.1 РАЗМЕЩЕНИЕ АРХИВНЫХ ФАЙЛОВ.....	17
6 СООБЩЕНИЯ СИСТЕМНОМУ ОПЕРАТОРУ	18
6.1 Лог системных сообщений	18
ПРИЛОЖЕНИЯ	20
Приложение 1 – Состав задач и утилит СКАДИ	20
Приложение 2 – Описание конфигурации DsCONFIG.....	22
Приложение 3 - Данные конфигурируемых реляционных таблиц.....	23
Приложение 4 – Динамически загружаемые модули	42
ЛИСТ ИЗМЕНЕНИЙ.....	43

1 ОБЩИЕ СВЕДЕНИЯ О ПО

1.1 НАИМЕНОВАНИЕ

Наименование изделия — Программная платформа средств комплексной автоматизации и диагностики «СКАДИ 1.0».

Сокращенное наименование: «СКАДИ».

Наименование документа 34185873.425510.001.32.М, обозначение по ГОСТ 34.201-89. Руководство системного программиста человеко-машинного интерфейса (ЧМИ) СКАДИ описано в документе 34185873.425510.001.32.01.М.

«СКАДИ 1.0» разрабатывается на основании документа «СКАДИ 1.0» разрабатывается на основании документа 34185873.425510.001.П2.М «Программная платформа АСУ ТП и диагностики для СВБУ АЭС и промышленных применений», Техническое задание, утвержденного 23 марта 2022г.

1.2 НАЗНАЧЕНИЕ И ФУНКЦИИ

СКАДИ предназначена для построения систем автоматизации на основе базовой платформы.

Основные функции СКАДИ:

- прием и передача данных;
- передача команд управление технологическим процессом;
- получение информационных сообщений, выработанных диагностическими и расчетными системами;
- регистрация и управление сигнализациями;
- управление архивами;
- управление протоколами;
- самодиагностика программных и аппаратных средств;
- контроль конфигурации параметров системы.

Программное обеспечение ЧМИ СКАДИ интегрировано с общим ПО СКАДИ. ПО ЧМИ СКАДИ описано отдельно в соответствующей документации.

1.3 НЕОБХОДИМЫЕ ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА

Для функционирования «СКАДИ 1.0» требуется наличие операционной системы семейств Linux или Windows. Операционное системное программное обеспечение устанавливается на все узлы, где должно работать ПО СКАДИ.

ПО «СКАДИ 1.0» предназначено для работы с является операционным программным обеспечением, требующим для своей работы 64-разрядных процессоров AMD/Intel с архитектурой x86_64, либо 32-разрядных процессоров с архитектурой x86.

Для использования средств защиты информации операционных систем рекомендуется использование ОС Astra Linux, начиная с версии 1.6.

2 СТРУКТУРА ПО

2.1 СОСТАВ ПО

Программное обеспечение СКАДИ состоит из набора исполняемых модулей, расположенных в каталоге инсталляции СКАДИ. Имена исполняемых модулей СКАДИ имеют шаблон Ds* для Linux-систем и Ds*.exe для Windows-систем. Исполняемые модули СКАДИ разбиваются на группы:

1. Рантайм-менеджер DsRM для общего мониторинга и управления процессами;
2. Задачи СКАДИ, запускаемые рантайм-менеджером;
3. Автономные исполняемые задачи СКАДИ;
4. Утилиты сервиса и отладки;
5. Динамически загружаемые проектные модули.

Рантайм-менеджер при старте запускает задачи СКАДИ из группы 2, формируя загрузку из процессов СКАДИ, рисунок 1.

```
ddag@sd2:~/n5mshu$ ps x | grep -v "sh -c" | grep Ds
52478 pts/1 S+ 0:05 DsRM -cfg_file DsConfig_dist.cfg
52543 pts/1 S+ 0:00 DsPersist -file T
52544 pts/1 S+ 0:01 DsSync
52545 pts/1 S+ 0:00 DsPinger
52546 pts/1 S+ 0:00 DsSync -task_name DsSync2
52547 pts/1 S+ 0:01 DsDVP -hl DprmOvhDI
52548 pts/1 S+ 0:00 DsAlarm -alarm_set RO_TO_TAI
52549 pts/1 S+ 0:00 DsXServ -echo_cmd TRUE -task_name DsXServImit
52550 pts/1 S+ 0:00 DsGroup -disp_set DIST_SCR
52551 pts/1 S+ 0:00 DsRBAC
52590 pts/0 S+ 0:00 grep --color=auto Ds
ddag@sd2:~/n5mshu$
```

Рисунок 1 – Рантайм—менеджер и запущенные им процессы

Рантайм-менеджер запускает процессы из состава задач СКАДИ группы 2, которые добавлены в конфигурацию. Перечень возможных задач описан в таблице 1-1 приложения 1. Задача DsHMI относится только к клиентскому ПО и описывается в рамках документации ПО ЧМИ.

Конфигурационный файл, в котором описаны задачи СКАДИ, имеет имя по умолчанию DsConfig.cfg. На основании такого файла рантайм-менеджер формирует загрузку. Конкретный файл может быть задан опцией командной строки `-cfg_file` в следующем виде.

```
DsRM -cfg_file DsConfig_<conftype>.cfg
```

В приведенном примере используется конфигурационный файл DsConfig_dist.cfg. Описание конфигурационного файла DsConfig приведено в приложении 2. Состав полей конфигурируемых реляционных таблиц приведен в приложении 3.

Список запускаемых задач соответствует записям массива TASK[] конфигурационного файла.

Автономные исполняемые задачи СКАДИ (таблица 1-2 приложения 1) запускаются независимо от рантайм-менеджера.

Утилиты сервиса и отладки (таблица 1-3 приложения 1) используются пользователями и разработчиками СКАДИ в качестве консольных приложений.

Динамические загружаемые проектные модули описаны в приложении 4. Проектные модули не входят в состав ПО СКАДИ, а относятся к конкретному проекту, как расширения по требованиям Заказчика. ПО СКАДИ реализует интерфейс для каждого типа модулей, позволяющий подключать конкретную реализацию, определяемую

конфигурацией. Динамически загружаемые проектные модули располагаются в корневом каталоге проекта, имена их каталогов соответствуют шаблону Dp*.

На рисунке 2 представлен каталог для проекта plg1.

```
ddag@sd2:~/plg1$ ls -al
total 5460
drwxrwxr-x 15 ddag ddag 4096 сен 6 08:46 .
drwxr-xr-x 19 ddag ddag 4096 июл 7 16:04 ..
drwxrwxr-x 2 ddag ddag 4096 июн 29 16:17 arfs
drwxrwxr-x 3 ddag ddag 4096 сен 5 09:09 db
drwxrwxr-x 4 ddag ddag 4096 сен 5 09:06 Dpd
drwxrwxr-x 4 ddag ddag 4096 сен 5 09:06 Dpdg
drwxrwxr-x 4 ddag ddag 4096 сен 5 09:06 Dperm
drwxrwxr-x 4 ddag ddag 4096 сен 5 09:06 Dprep
drwxrwxr-x 4 ddag ddag 4096 сен 5 09:06 Dprm
drwxrwxr-x 4 ddag ddag 4096 сен 5 09:06 Dpxch
-rw-rw-r-- 1 ddag ddag 9555 июн 29 15:47 DsConfig_arch.cfg
-rw-rw-r-- 1 ddag ddag 9559 июн 29 15:47 DsConfig.cfg
-rw-rw-r-- 1 ddag ddag 9677 сен 5 09:06 DsConfig_distarch.cfg
-rw-rw-r-- 1 ddag ddag 9678 июл 1 11:52 DsConfig_distarch.cfg.save.050922
-rw-rw-r-- 1 ddag ddag 8397 июл 1 11:51 DsConfig_dist.cfg
-rw-rw-r-- 1 ddag ddag 4827 сен 5 09:06 DsEdit.cfg
-rw-rw-r-- 1 ddag ddag 57 фев 18 2022 ds_ekp.bat
-rw-rw-r-- 1 ddag ddag 17 янв 25 2022 dsend.bat
-rwxr-xr-x 1 ddag ddag 17 мар 10 15:06 dsend.sh
-rw-rw-r-- 1 ddag ddag 2374 июн 29 15:47 DsFonts.cfg
drwxrwxr-x 4 ddag ddag 4096 июл 1 12:21 Dshm
-rw-rw-r-- 1 ddag ddag 7797 июн 29 15:57 DsHMI.cfg
-rw-rw-r-- 1 ddag ddag 2620 мар 11 07:43 DsHMI_ekp.cfg
```

Рисунок 2 – Проектный каталог с динамическими модулями

В приведенном на рисунке проектном каталоге отмечены подкаталоги Dpd, Dpdg, Dperm, Dprep, Dprm, Dpxch. Каждый из приведенных модулей на рисунке является реализацией по умолчанию. Соответствие между именем модуля и интерфейса приведено в таблице 4-1 приложения 4.

2.2 СВЯЗИ СОСТАВНЫХ ЧАСТЕЙ ПО

Программные модули СКАДИ предназначены для работы в распределенной архитектуре. Распределенная архитектура состоит из узлов, связанных между собой сетевыми протоколами. Каждый узел представляет собой адресуемый по сети компьютер с установленным ПО СКАДИ. Если на узел установлено ПО в полном составе, то должен запускаться рантайм-менеджер, который, в свою очередь, запускает задачи согласно установленной конфигурации. Это означает наличие полной загрузки ПО. Если на узел установлены отдельные части ПО, относящиеся к группе 3 автономных задач, то должны запускаться приложения агента обмена DsXCH и шлюза передачи DsXGate.

Автоматический запуск и контроль за работой рантайм-менеджера и автономных задач осуществляется средствами операционной системы. При первом старте рантайм-менеджер создает глобалы общей памяти и запускает необходимые задачи. Последующий рестарт рантайм-менеджера приводит к подсоединению рестартовавшего приложения к имеющейся загрузке.

Логическая структура ПО СКАДИ представлена на рисунке 3. В верхней части рисунка представлены узлы СКАДИ, в нижней части – узлы внешних подсистем. Верхний левый узел СКАДИ на рисунке раскрывает состав используемого ПО.

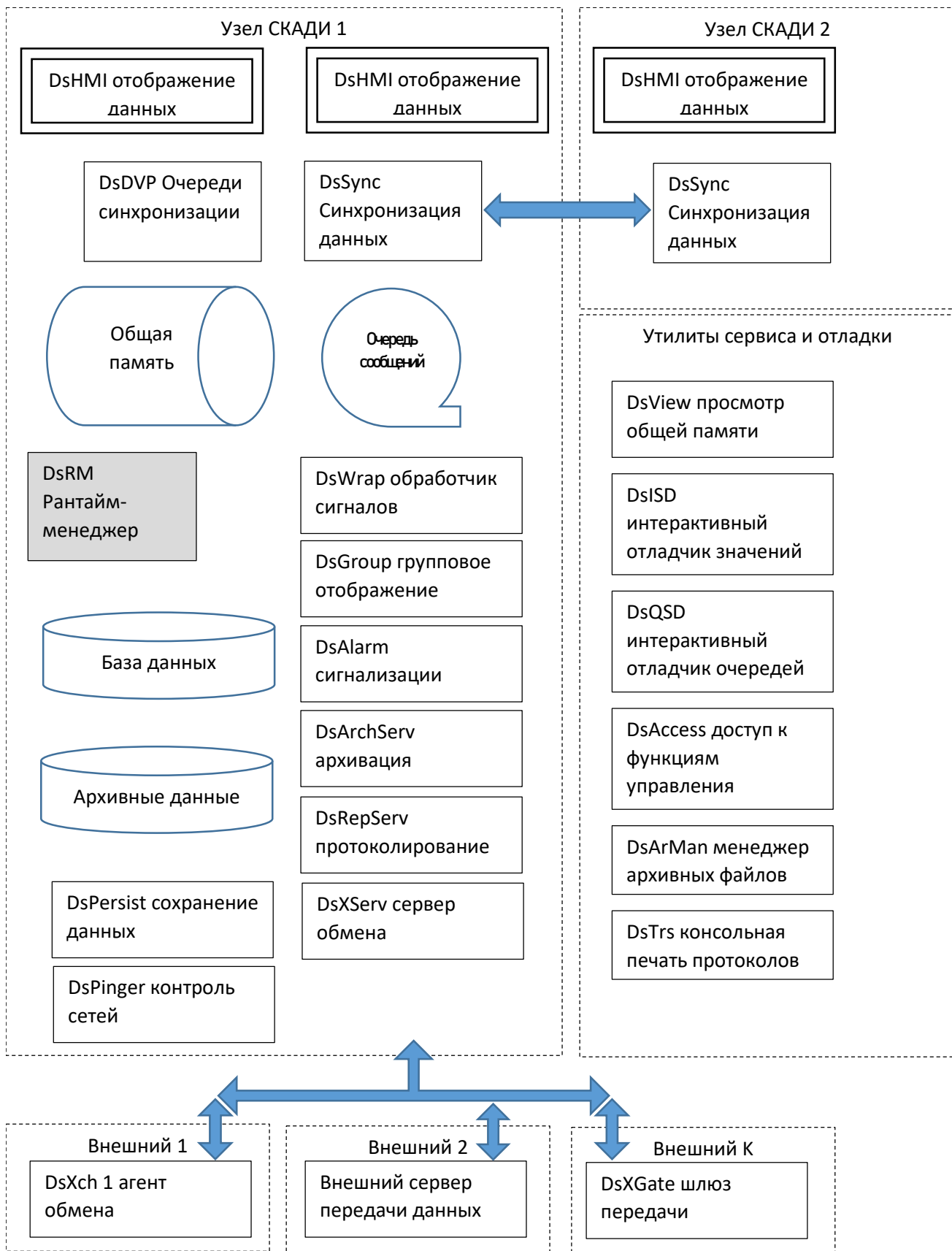


Рисунок 3 – Логическая структура ПО

На узле СКАДИ расположены рантайм-менеджер и задачи, им запускаемые. Улититы сервиса и отладки устанавливаются на каждый из узлов, независимо от его назначения. Узлы связи с внешними подсистемами представляют собой шлюзовые компьютеры с установленными на них автономными приложениями DsXCH, DsXGate. Возможен вариант с установкой внешнего сервера передачи данных (на рисунке – внешний 2) с общим протоколом обмена.

Рантайм-менеджер и запускаемые им задачи связаны по данным и очередям сообщений, расположенным в общей памяти.

Узлы с ПО СКАДИ взаимодействуют между собой по сети посредством задач DsSync.

Автономные приложения взаимодействуют с задачами серверов обмена DsXServ по сети.

Динамически загружаемые проектные модули загружаются задачами на этапе старта или по запросу, в зависимости от назначения модуля.

3 НАСТРОЙКА ПРОГРАММЫ

3.1 ИНСТАЛЛЯЦИЯ

3.1.1 Инсталляция для ОС Astra Linux, ОС Linux

Разместить пакет ПО СКАДИ для соответствующей архитектуры в корневой каталог:

- scadi_i386_1.0.0.deb для архитектуры X86;
- scadi_amd64_1.0.0.deb для архитектуры X64.

Для установки пакета scadi_i386_1.0.0.deb требуются дополнительные программные средства.

```
librt.so.1 => /lib32/librt.so.1 (0xf7ecf000)
libdl.so.2 => /lib32/libdl.so.2 (0xf7eca000)
libc.so.6 => /lib32/libc.so.6 (0xf7d13000)
libpthread.so.0 => /lib32/libpthread.so.0 (0xf7cf6000)
libm.so.6 => /lib32/libm.so.6 (0xf6f8a000)
libtinfo.so.5 => /lib32/libtinfo.so.5 (0xf7cf3000)
libpthread.so.0 => /lib32/libpthread.so.0 (0xf7cd4000)
```

Для установки ПО ЧМИ СКАДИ требуются дополнительные программные средства.

```
libgdk-x11-2.0.so.0 => /usr/local/scadi0/lib/libgdk-x11-2.0.so.0 (0xf7e90000)
libgtk-x11-2.0.so.0 => /usr/local/scadi0/lib/libgtk-x11-2.0.so.0 (0xf799c000)
libcairo.so.2 => /usr/lib32/i386-linux-gnu/libcairo.so.2 (0xf783b000)
libpango-1.0.so.0 => /usr/local/scadi0/lib/libpango-1.0.so.0 (0xf77ea000)
libglib-2.0.so.0 => /lib32/i386-linux-gnu/libglib-2.0.so.0 (0xf76be000)
libgobject-2.0.so.0 => /usr/lib32/i386-linux-gnu/libgobject-2.0.so.0
(0xf765e000)
libpangocairo-1.0.so.0 => /usr/local/scadi0/lib/libpangocairo-1.0.so.0
(0xf748b000)
libgdk_pixbuf-2.0.so.0 => /usr/local/scadi0/lib/libgdk_pixbuf-2.0.so.0
(0xf7460000)
libgio-2.0.so.0 => /usr/lib32/i386-linux-gnu/libgio-2.0.so.0 (0xf7286000)
libfontconfig.so.1 => /usr/lib32/i386-linux-gnu/libfontconfig.so.1
(0xf7243000)
libfreetype.so.6 => /usr/lib32/i386-linux-gnu/libfreetype.so.6 (0xf718f000)
libXrender.so.1 => /usr/lib32/i386-linux-gnu/libXrender.so.1 (0xf7183000)
libXinerama.so.1 => /usr/lib32/i386-linux-gnu/libXinerama.so.1 (0xf717f000)
libXi.so.6 => /usr/lib32/i386-linux-gnu/libXi.so.6 (0xf716a000)
libXrandr.so.2 => /usr/lib32/i386-linux-gnu/libXrandr.so.2 (0xf715d000)
libXcursor.so.1 => /usr/lib32/i386-linux-gnu/libXcursor.so.1 (0xf7151000)
libXcomposite.so.1 => /usr/lib32/i386-linux-gnu/libXcomposite.so.1
(0xf714d000)
libXdamage.so.1 => /usr/lib32/i386-linux-gnu/libXdamage.so.1 (0xf7149000)
libXfixes.so.3 => /usr/lib32/i386-linux-gnu/libXfixes.so.3 (0xf7142000)
libX11.so.6 => /usr/lib32/i386-linux-gnu/libX11.so.6 (0xf6ff4000)
libXext.so.6 => /usr/lib32/i386-linux-gnu/libXext.so.6 (0xf6fdf000)
libgmodule-2.0.so.0 => /usr/lib32/i386-linux-gnu/libgmodule-2.0.so.0
(0xf6f85000)
libatk-1.0.so.0 => /usr/local/scadi0/lib/libatk-1.0.so.0 (0xf6f5e000)
libpangoft2-1.0.so.0 => /usr/local/scadi0/lib/libpangoft2-1.0.so.0
(0xf6f44000)
libpixman-1.so.0 => /usr/lib32/i386-linux-gnu/libpixman-1.so.0 (0xf6e77000)
libpng16.so.16 => /usr/lib32/i386-linux-gnu/libpng16.so.16 (0xf6e3d000)
libxcb-shm.so.0 => /usr/lib32/i386-linux-gnu/libxcb-shm.so.0 (0xf6e39000)
libxcb.so.1 => /usr/lib32/i386-linux-gnu/libxcb.so.1 (0xf6e0d000)
libxcb-render.so.0 => /usr/lib32/i386-linux-gnu/libxcb-render.so.0
(0xf6dfe000)
libz.so.1 => /lib32/i386-linux-gnu/libz.so.1 (0xf6de3000)
```

```
libgthread-2.0.so.0      =>      /usr/lib32/i386-linux-gnu/libgthread-2.0.so.0
(0xf6de0000)
libthai.so.0 => /usr/local/scadi0/lib/libthai.so.0 (0xf6dd5000)
libpcre.so.3 => /lib32/i386-linux-gnu/libpcre.so.3 (0xf6d5a000)
libffi.so.6 => /usr/lib32/i386-linux-gnu/libffi.so.6 (0xf6d51000)
libselinux.so.1 => /lib32/i386-linux-gnu/libselinux.so.1 (0xf6d24000)
libresolv.so.2 => /lib32/libresolv.so.2 (0xf6d0c000)
libmount.so.1 => /lib32/i386-linux-gnu/libmount.so.1 (0xf6cb4000)
libexpat.so.1 => /lib32/i386-linux-gnu/libexpat.so.1 (0xf6c8a000)
libharfbuzz.so.0 => /usr/local/scadi0/lib/libharfbuzz.so.0 (0xf6bed000)
libXau.so.6 => /usr/lib32/i386-linux-gnu/libXau.so.6 (0xf6be9000)
libXdmp.so.6 => /usr/lib32/i386-linux-gnu/libXdmp.so.6 (0xf6be2000)
libdatrie.so.1 => /usr/local/scadi0/lib/libdatrie.so.1 (0xf6bd6000)
libblkid.so.1 => /lib32/i386-linux-gnu/libblkid.so.1 (0xf6b87000)
libgraphite2.so.3 => /usr/local/scadi0/lib/libgraphite2.so.3 (0xf6b58000)
libbsd.so.0 => /lib32/i386-linux-gnu/libbsd.so.0 (0xf6b3c000)
libuuid.so.1 => /lib32/i386-linux-gnu/libuuid.so.1 (0xf6b36000)
```

Установить пакет ПО СКАДИ средствами ОС:

```
sudo apt install ./scadi_i386_1.0.0.deb      (* X86 *)
sudo apt install ./scadi_amd64_1.0.0.deb    (* X64 *)
```

3.1.2 Инсталляция для ОС Windows

Разместить инсталлятор ПО СКАДИ Scadi_1_0_0_x86.msi для 32-bit на жестком диске.

Для установки пакета Scadi_1_0_0_x86.msi требуются дополнительные программные средства:

```
freetype6.dll
intl.dll
libatk-1.0-0.dll
libcairo-2.dll
libexpat-1.dll
libfontconfig-1.dll
libgcc_s_dw2-1.dll
libgdk_pixbuf-2.0-0.dll
libgdk-win32-2.0-0.dll
libpango-1.0-0.dll
libpangocairo-1.0-0.dll
libpangoft2-1.0-0.dll
libpangowin32-1.0-0.dll
libpng14-14.dll
pdcurses.dll
zlib1.dll
```

Установить ПО СКАДИ запуском инсталлятора.

Поместить дополнительные программные средства в каталог установки исполняемых файлов ПО СКАДИ. Должна быть обеспечена загрузка библиотек приложениями ПО СКАДИ, их использующими.

3.2 НАСТРОЙКА ПРОЕКТА

3.2.1 Базовые файлы проекта

Поскольку СКАДИ предназначена для построения систем автоматизации на основе базовой платформы, для ее использования необходимо создание проекта системы автоматизации. Для проекта системы автоматизации создается корневой каталог проекта.

При дальнейшей работе все данные и конфигурация, относящиеся к проекту, должны находиться в корневом каталоге проекта.

Установить корневой каталог проекта (вместо названия project подставляется конкретное имя разрабатываемого проекта):

```
mkdir project  
cd project
```

Корневой каталог проекта используется для размещения проектных данных и программного обеспечения, относящихся к конкретному проекту. Он должен иметь определенную структуру для обеспечения доступа к указанным данным.

Проектный каталог для каждого проекта должен представлять из себя структуру:

- project> - место для конфигурационных файлов;
- project/db> - место для файлов базы данных;
- project/vis> - место для графических данных;
- project/wrk - место для рабочих файлов;
- project/arfs - место для размещения среднесрочных архивов;
- project/lta - место для размещения долговременных архивов;
- project/tra - место для размещения аварийных архивов;
- project/rep - место для протоколов и выборок страниц;
- project/Dp* - плагины подключаемых программных модулей.

Наличие общего конфигурационного файла должно обеспечивать старт системы командой:

```
project> DsRM -cfg_file DsConfig_<conftype>.cfg
```

В качестве аргументов команды задается имя конфигурационного файла. Если имя конфигурационного файла не задано, то берется по умолчанию DsConfig.cfg. Описание содержимого конфигурационного файла приведено в приложении 2. Описание данных конфигурируемых параметров приведено в приложении 3.

Каталог базы данных проекта project/db содержит набор файлов для базы данных с расширением .tsd:

- types.tsd - проектно-зависимые метаданные;
- global_<globname>.tsd – набор параметров базы данных реального времени для globname;
- mod_*.tsd – скрипты модификации базы данных;
- imi*_*.tsd – скрипты данных имитатора внешних систем.

Каталог рабочих файлов проекта project/wrk содержит набор файлов групп параметров с расширением *.pgg.

Каталоги архивных файлов проекта project/arfs project/lta project/tra содержат наборы файлов архивов согласно таблице 1.

Название типа	Тип архива artype	Класс реализации aclass	Формат архивного файла afmt	Сохранение глобала	Примечание
Оперативный архив	OPER	PERIOD OLD QUEUE	FOA	Глобал 'а' persist_interval	Каталог arfs, файл <project>_<name>.foa
Среднесрочный апертурный по очередям	STA	QUEUE	FBA	Глобал 'с' persist_interval	Каталог arfs, файл <name><fileno>.fba Глобал 'с' <project>_<name>.fca Глобал 'с' ссылается на 'а' посредством refglobal
Среднесрочный апертурный по изменениям	STA	OLD	FBA	Аналогично	Аналогично
Среднесрочный непрерывный	STA	PERIOD	FBA	Аналогично	Аналогично
Расчетных параметров	STA	CALC	FBA	Аналогично	Аналогично
Архив сигнализаций	STA	LOG	FBA	Аналогично	Аналогично
Аварийный архив	TRA	TRAPERIOD TRAOLD	FBA	Аналогично	Каталог tra/T<num><date>, файл <name><fileno>.fba Глобал 'с' <project>_<name>.fca Глобал 'с' ссылается на 'а' посредством refglobal
Долговременный архив	LTA	LTACONV	FTA	НЕТ	Каталог lta/<date> Файл <name><fileno>.fta
Архив системных событий	SYSLOG	--	--	Нет глобала, системный Syslogs	Каталог db Файл <project>_Syslogs fla

Таблица 1. Конфигурации для разных типов архивов

Указанные файлы имеют расширения:

- fta – текстовый формат архива;
- fba – двоичный формат архива
- foa – двоичный формат оперативного архива.

При этом архив системных событий <project>_Syslogs fla содержит таблицу сообщений, располагающуюся в каталоге db.

Каталог протоколов и выборок страниц project/wrk должен быть создан для последующего использования программой DsRepServ, которая будет выкладывать в него сгенеренные протоколы и файлы страниц.

Плагины подключаемых программных модулей образует каталоги с именами систем модулей для каждого. Имя системы для модуля представляет собой префикс модуля до первой заглавной буквы, начиная со второй. Например, плагины DpdgButtons и DpdgTrends будут находиться внутри одной системы модулей с именем Dpdg. Система модулей представляет собой каталог, в нашем примере Dpdg. Внутри системы модулей производится разделение на подкаталоги с именами:

- Code – содержит динамически загружаемые коды программных модулей;
- Sym – содержит символьные файлы интерфейса программных модулей;
- Mod – содержит исходные тексты программных модулей (опционально, если исходные тексты поставляются Заказчику).

Файл внутри подкаталогов имеют имена за вычетом имени системы модулей. Для нашего примера DpdgButtons представляется файлом кода Dpdg/Code/Buttons.ocf Dpdg/Sym/Buttons.osf. При наличии исходного текста может содержаться файл Dpdg/Mod/Buttons.odc.

3.2.2 Группы сетевых узлов

СКАДИ представляет собой ПО для построения распределенных систем, состоящих из нескольких хостов. На этих хостах устанавливается программа DsSync, которая осуществляет синхронизацию данных общей памяти. Каждый экземпляр программы DsSync должен иметь адреса абонентов сети, которые участвуют в процессе обмена.

Каждый хост может участвовать в обмене данными несколькими сетевыми интерфейсами. Стандартным решением является использование по две сетевые карты на хост для нулевой и первой сетей. Для такой конфигурации возникает 2 группы сетевых узлов, нулевые интерфейсы для всех хостов и первые интерфейсы для всех хостов. Каждая группа участвует в коммуникации независимо от другой для исключения отказа по общей причине.

Возможен более сложный вариант с разбиением на сегменты независимых сетей. Тогда групп хостов может возникнуть 4 или больше. Группы сетевых узлов описываются в конфигурации массивами HGROUП[], сетевые узлы описываются массивами HOST[].

Для конфигурирования группы сетевых узлов необходимо сначала определить набор хостов с адресами. Затем следует сделать ссылки для каждого хоста о принадлежности к конкретной группе.

```
HOST[1].name=WS20BI01
```

```
HOST[1].in_groups=PINGER SYNC SYNC2
```

```
HOST[1].ip_address=10.0.5.216
```

```
HOST[1].ip_address_2=10.0.6.216
```

Данный фрагмент конфигурации устанавливает хост с именем WS20BI01, принадлежащим группам с именами PINGER, SYNC, SYNC2. В конфигурационном файле должны быть определены соответствующие имена групп. Например,

```
HGROUP[0].name=PINGER
```

```
HGROUP[0].task_name=DsPinger
```

```
HGROUP[0].hgtype=PING
```

Рисунок 4 иллюстрирует соотношение сетевых узлов и групп, к ним привязанным.

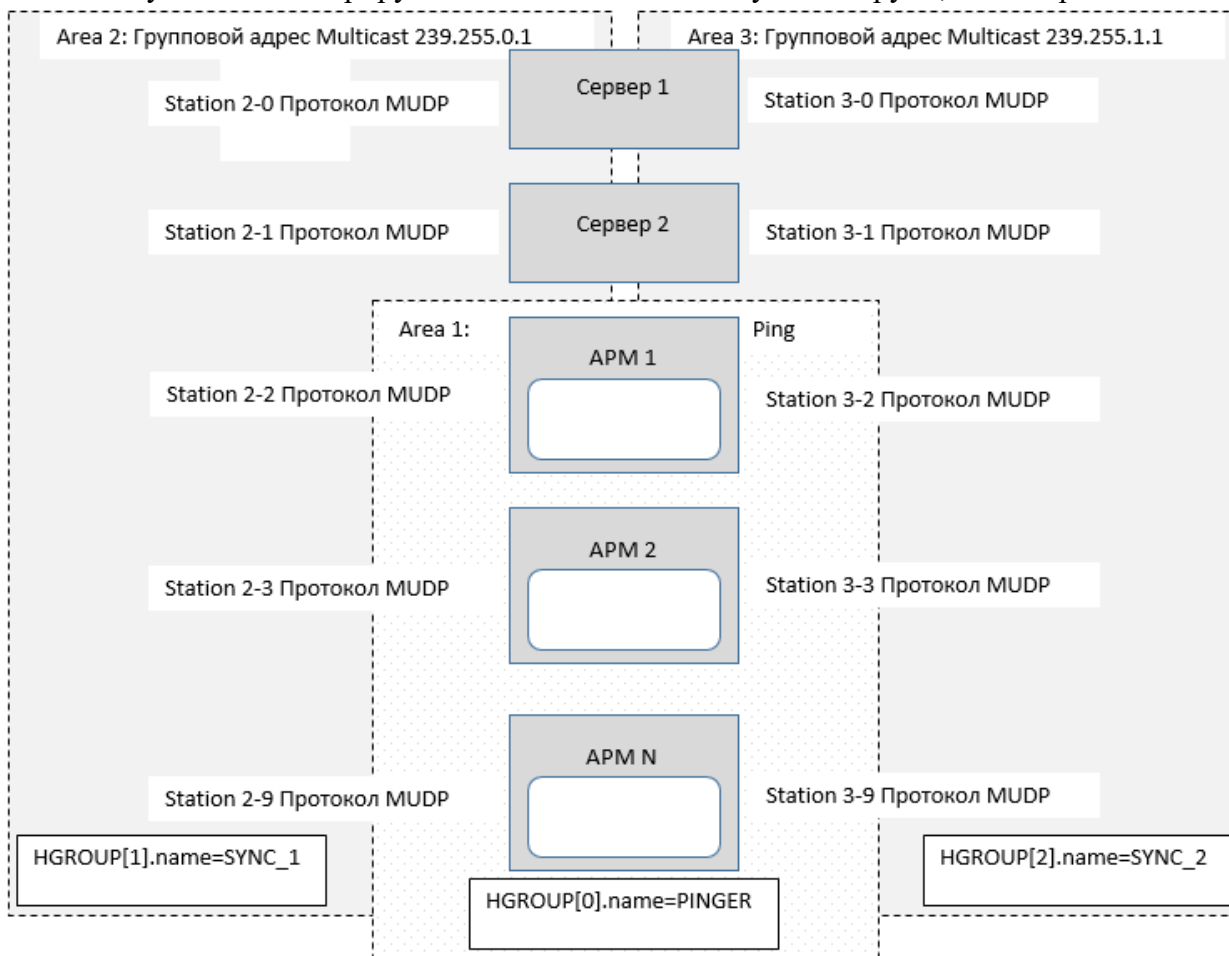


Рисунок 4 – Сетевые узлы и группы

Группа сетевых узлов определяет характер взаимодействия. Вид сетевого взаимодействия устанавливается параметром hgtype. Возможные значения параметра:

- 1-PING пингование IP-адреса;
- 2-SYNC сеть синхронизации СКАДИ;
- 3-DVP область обновления таблиц по очередям;
- 4-VALS распространение входных данных.

При заполнении групп необходимо правильно заполнять в файле DsConfig.cfg параметр hgtype групп корректными значениями.

4 ПРОВЕРКА ПРОГРАММЫ

4.1 ПРОВЕРКА НАЛИЧИЯ ПО И ЗАГРУЗКИ

Наличие проекта СКАДИ проверяется через наличие корневого каталога проекта project(вместо названия project подставляется конкретное имя разрабатываемого проекта).
cd project

В корневом каталоге проекта должны находиться каталоги и конфигурационные файлы, описанные в разделе 3.2.1.

Наличие установленного ПО СКАДИ проверяют путем наличия путей к бинарным исполняемым приложениям. Для Unix-систем можно использовать запрос пути.
which DsRM

Для проверки того, что СКАДИ запущена, можно использовать консольную утилиту из проектного каталога.

DsAccess

При отсутствии загрузки будет выведено сообщение: “SCAD I not found”. При наличии загрузки будет выведен перевод строки.

4.2 ПРОВЕРКА ЗАГРУЗКИ ГЛОБАЛОВ ОБЩЕЙ ПАМЯТИ

Наличие и состав глобалов общей памяти проверяется командой.

```
DsView -globtype .
```

Команда DsView -globtype . выводит на консоль список имен, типов и заголовков объектов общей памяти.

```
C:\DWDagaev\plg1>DsView -globtype .
G 0 Globals n_total_objs= 512, n_used_objs= 45, sz_used_mem= 43008, usr_data= 0
E 1 Enums n_total_objs= 200, n_used_objs= 138, sz_used_mem= 85600, usr_data= 400
T 2 Types n_total_objs= 50, n_used_objs= 41, sz_used_mem= 7200, usr_data= 1
A 3 Attrs n_total_objs= 500, n_used_objs= 396, sz_used_mem= 104000, usr_data= 100
S 4 Syslogs n_total_objs= 6000, n_used_objs= 15, sz_used_mem= 1272000, usr_data= 27580
s 5 Consts n_total_objs= 432, n_used_objs= 205, sz_used_mem= 25920, usr_data= 17
s 6 Characteristics n_total_objs= 32, n_used_objs= 10, sz_used_mem= 5120, usr_data= 18
s 7 Users n_total_objs= 64, n_used_objs= 8, sz_used_mem= 8192, usr_data= 19
s 8 Tasks n_total_objs= 15, n_used_objs= 15, sz_used_mem= 7560, usr_data= 20
s 9 HGroups n_total_objs= 6, n_used_objs= 6, sz_used_mem= 3504, usr_data= 21
s 10 Hosts n_total_objs= 4, n_used_objs= 4, sz_used_mem= 1184, usr_data= 22
s 11 XGroups n_total_objs= 1, n_used_objs= 1, sz_used_mem= 1044, usr_data= 23
s 12 Xchs n_total_objs= 3, n_used_objs= 3, sz_used_mem= 1488, usr_data= 24
s 13 CSets n_total_objs= 25, n_used_objs= 25, sz_used_mem= 35500, usr_data= 27
s 14 Cells n_total_objs= 360, n_used_objs= 360, sz_used_mem= 391680, usr_data= 26
s 15 Reports n_total_objs= 5, n_used_objs= 5, sz_used_mem= 1940, usr_data= 29
s 16 Repdatas n_total_objs= 1500, n_used_objs= 1500, sz_used_mem= 498000, usr_data= 28
s 17 WSets n_total_objs= 25, n_used_objs= 25, sz_used_mem= 1900, usr_data= 31
s 18 Wdatas n_total_objs= 140, n_used_objs= 140, sz_used_mem= 454720, usr_data= 30
s 19 VFiles n_total_objs= 100, n_used_objs= 6, sz_used_mem= 16400, usr_data= 32
s 20 FVars n_total_objs= 10000, n_used_objs= 503, sz_used_mem= 3240000, usr_data= 33
s 21 Alarms n_total_objs= 260, n_used_objs= 260, sz_used_mem= 23920, usr_data= 34
s 22 Archs n_total_objs= 3, n_used_objs= 3, sz_used_mem= 384, usr_data= 35
s 23 Gstats n_total_objs= 512, n_used_objs= 45, sz_used_mem= 77824, usr_data= 25
q 24 RBACq n_total_objs= 1200, n_used_objs= 0, sz_used_mem= 57600, usr_data= 0
q 25 Parmq n_total_objs= 1200, n_used_objs= 0, sz_used_mem= 57600, usr_data= 0
q 26 Syncq n_total_objs= 5000, n_used_objs= 0, sz_used_mem= 240000, usr_data= 0
q 27 DVPq n_total_objs= 5000, n_used_objs= 0, sz_used_mem= 240000, usr_data= 0
q 28 Parm2q n_total_objs= 1200, n_used_objs= 0, sz_used_mem= 57600, usr_data= 0
q 29 Sync2q n_total_objs= 5000, n_used_objs= 0, sz_used_mem= 240000, usr_data= 0
q 30 DVP2q n_total_objs= 5000, n_used_objs= 0, sz_used_mem= 240000, usr_data= 0
q 31 Groupq n_total_objs= 2000, n_used_objs= 0, sz_used_mem= 96000, usr_data= 0
q 32 Doperq n_total_objs= 1200, n_used_objs= 32, sz_used_mem= 57600, usr_data= 32
q 33 Doper2q n_total_objs= 1200, n_used_objs= 0, sz_used_mem= 57600, usr_data= 0
n 34 TPTs n_total_objs= 10, n_used_objs= 4, sz_used_mem= 3960, usr_data= 40
n 35 AIs n_total_objs= 14826, n_used_objs= 14826, sz_used_mem= 6167616, usr_data= 36
i 36 AII n_total_objs= 14826, n_used_objs= 14826, sz_used_mem= 533736, usr_data= 0
n 37 DIs n_total_objs= 148707, n_used_objs= 148707, sz_used_mem= 51155208, usr_data= 37
i 38 DIi n_total_objs= 148707, n_used_objs= 148707, sz_used_mem= 5353452, usr_data= 0
q 39 XImiq n_total_objs= 1200, n_used_objs= 447, sz_used_mem= 57600, usr_data= 447
q 40 Alarmq n_total_objs= 2000, n_used_objs= 586, sz_used_mem= 96000, usr_data= 589
a 41 AAI n_total_objs= 35000, n_used_objs= 30519, sz_used_mem= 2272000, usr_data= 30519
a 42 ADI n_total_objs= 50000, n_used_objs= 0, sz_used_mem= 8360000, usr_data= 0
c 43 ADIc n_total_objs= 100, n_used_objs= 0, sz_used_mem= 5600, usr_data= 19
q 44 RepSq n_total_objs= 2000, n_used_objs= 0, sz_used_mem= 96000, usr_data= 0
```

Рисунок 5 – Фрагмент списка системных глобалов

Для каждого глобала памяти выводится номер, имя, максимальное число объектов n_total_objs, текущее число объектов или номер n_used_objs, размер используемой памяти sz_used_mem, поле переменного значения usr_data.

5 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

5.1 РАЗМЕЩЕНИЕ АРХИВНЫХ ФАЙЛОВ

При создании проекта должен быть размещен каталог project/arfs - место для среднесрочных архивов. При этом архивные файлы среднесрочного архива STA автоматически не создаются. Создание файлов среднесрочного архива осуществляется архив-менеджером - утилитой ArMan.

```
DsArMan -cmd c -artype STA -arch <arch_global_name> -afmt <FBA|FTA> -number <number> -size <size>
```

Глобалу среднесрочного архива arch_global_name соответствуют несколько архивных файлов, количество определено параметром n_files (Таблица 3-13 приложения 3). Поэтому должно быть создано n_files файлов с номерами (0, n_files-1), для каждого запускается команда DsArMan. Формат файла может быть либо FBA – для двоичного формата архива, либо FTA – для текстового формата архива. Размер архива size задается в записях, при этом запись для двоичного архива равна 48 байт, а запись для текстового архива равна 108 байт. Размер файла архива вычисляется по формуле:

- filesize = size * 48 + 24 для бинарного архива FBA;
- filesize = size * 108 для текстового архива FTA.

Пример создания архива из 2 файлов приведен ниже.

```
DsArMan -cmd c -arch ADI -afmt FBA -artype STA -number 0 -size 300000
```

```
DsArMan -cmd c -arch ADI -afmt FBA -artype STA -number 1 -size 300000
```

Как результат, получаем 2 двоичных файла среднесрочного архива размером 144024 байт, рисунок 6.

Имя	Дата изменения	Тип	Размер
ADI0.fba	20.07.2022 8:15	Файл "FBA"	14 063 КБ
ADI1.fba	19.07.2022 9:32	Файл "FBA"	14 063 КБ
arch	27.06.2022 16:44	Текстовый докум...	1 КБ
arch.txt.bak	27.06.2022 15:11	Файл "BAK"	1 КБ
plg1_AAI.foa	27.06.2022 16:24	Файл "FOA"	2 219 КБ
plg1_AD1c.fca	27.06.2022 15:59	Файл "FCA"	6 КБ

Рисунок 6 – Созданные файлы среднесрочного архива

При создании файлов среднесрочного архива их содержимое заполняется нулевыми значениями, которые означают отсутствие данных.

6 СООБЩЕНИЯ СИСТЕМНОМУ ОПЕРАТОРУ

6.1 ЛОГ СИСТЕМНЫХ СООБЩЕНИЙ

Для записи системных сообщений для каждого из процессов СКАДИ сделан специальный лог – глобал S. Лог системных сообщений представляет собой кольцевой буфер, в котором сохраняются сообщения даже после завершения работы модулей. Во время работы и по окончании лог сбрасывается в файл db/<project>_Syslogs fla. При возобновлении работы общая память восстанавливается из файла с сохранением ранее сделанных сообщений. Удаление файла лога приведет к созданию нового, очищенного от предыдущих сообщений лога при очередной загрузке СКАДИ.

Просмотр лога системных сообщений осуществляется командой DsView –global 4 (либо DsView –globname Syslogs), вывод на консоль приведен на рисунке 7.

```
08/09/22 08:53:37.028 INFO 27579# DsRepServ 0@0 DszarRepServ.Restart DsRepServ task (re)started old_status=5 periods=0 1
0 load pct=80
08/09/22 08:53:57.378 ERROR 27580# DsDOPER 0@0 DshhModuls.LoadModule failed to load module: DpdUsrScreva, last imported:
DsScrEvals.FrmDataDesc
08/09/22 09:48:44.611 ERROR 27581# DsAlarm 0@0 DsrmlAlarm.UpdateSets Alarms in set #1 exceeds maximum size 20
08/09/22 11:24:04.748 AUX 27582# DsRM 0@0 DsrmlServ.AloneProcess.Create Process 'DsAlarm -alarm_set RO_TO_TAI' created wi
th return code=0
08/09/22 11:24:04.813 INFO 27583# DsAlarm 0@0 DsrmlAlarm.ReadIndices alarmsets_per_host 3 configured, 3 available
08/09/22 11:24:04.813 INFO 27584# DsAlarm 0@0 DsrmlAlarm.AlarmSet Use A0+ alarm set, dst_masks=1, enabled=0
08/09/22 11:24:04.813 INFO 27585# DsAlarm 0@0 DsrmlAlarm.AlarmSet Use B20+ alarm set, dst_masks=4, enabled=0
08/09/22 11:24:04.813 INFO 27586# DsAlarm 0@0 DsrmlAlarm.AlarmSet Use C40+ alarm set, dst_masks=16, enabled=0
08/09/22 11:24:04.813 INFO 27587# DsAlarm 0@0 DsrmlAlarm.Restart DsAlarm task (re)started old_status=5 periods=0 10
```

Рисунок 7 – Фрагмент лога системных сообщений глобала Syslogs

Сообщения системного лога представляют собой записи определенного формата Syslog. Например, можно разобрать по полям следующее сообщение.

```
08/09/22 09:48:44.611 ERROR 27581# DsAlarm 0@0 DsrmlAlarm.UpdateSets
Alarms in set #1 exceeds maximum size 20
```

Поле timestamp содержит время, в строковом представлении равное «08/09/22 09:48:44.611», поле важности severity – код ERROR, поле номера сообщения msg_id – номер 27581, поле имени процесса module – имя программы DsAlarm, строка 0@0 означает user_id = 0 (сервер), host_ind = 0 (нулевой хост), поле имени действия operation содержит «DsrmlAlarm.UpdateSets», поле сообщения message – шаблон Alarms in set \$ exceeds maximum size \$, а поле params – параметры «#1 20».

Все поля сообщений глобала S приведены в таблице 2.

Поле	Наименование	Значения	Примечание
timestamp	Метка времени	Время	
severity	Код важности	NONE, LIST, AUX, INFO, WARNING, ERROR, FATAL	0-нет, 1-список, 2-доп, 3-инфо, 4-предупреждение, 5-ошибка, 6-фатально данных
user_id	Идентификатор пользователя	0 ...	
host_ind	Номер хоста	0 ...	В таблице Hosts
msg_id	Номер сообщения	0 ...	
operation	Имя действия	<Имя>	
module	Имя процесса	<Имя>	В таблице Tasks
message	Шаблон сообщения	<Строка>	
params	Параметры к шаблону	<Строка>	

Таблица 2. Поля сообщений глобала S

Глобал S системных сообщений разбит на непересекающиеся области из фиксированного количества сообщений для каждого процесса. Количество сообщений задается параметром `DsConfig.cfg:ldr_task_msgs=400` и не меняется после конфигурирования. Для каждого процесса сформирован кольцевой буфер из 400 сообщений, при заполнении которого перезаписываются наиболее сообщения данного процесса. Параметр `user_data` в заголовке глобала Syslogs определяет идентификатор следующего сообщения.

Команда `DsView -global 4 -fld t` делает прямую распечатку глобала S. Фрагмент распечатки представлен на рисунке 8.

```
Syslog 5312 -msg_id 27576 08/09/22 08:53:37.019 -severity INFO -user_id 0 -host_ind 0 -operation DszarArchServ.InitArch
s -module DsArchServ -message $ archivers created -params 2$ -text 2 archivers created
Syslog 5313 -msg_id 27577 08/09/22 08:53:37.019 -severity INFO -user_id 0 -host_ind 0 -operation DszarArchServ.Restart
-module DsArchServ -message DsArchServ task (re)started old_status=$ periods=0 $ -params 5$1$ -text DsArchServ task (re)
started old_status=5 periods=0 1
----- SyslogList task_id= 15 current= 46 old= 0
Syslog 5601 -msg_id 83 01/07/22 10:27:46.622 -severity INFO -user_id 0 -host_ind 0 -operation DszarRepServ.InitReps -mo
dule DsRepServ -message $ reporters created from $($) to $($) -params 5$comp-dagaevdv_0$0$s1bb_0$4$ -text 5 reporters cr
eated from comp-dagaevdv_0(0) to s1bb_0(4)
Syslog 5602 -msg_id 84 01/07/22 10:27:46.622 -severity INFO -user_id 0 -host_ind 0 -operation DszarRepServ.Restart -mod
ule DsRepServ -message DsRepServ task (re)started old_status=$ periods=0 $ load_pct=$ -params 5$10$80$ -text DsRepServ t
ask (re)started old_status=5 periods=0 10 load_pct=80
Syslog 5603 -msg_id 166 01/07/22 10:28:18.657 -severity INFO -user_id 0 -host_ind 0 -operation DszarRepServ.InitReps -m
odule DsRepServ -message $ reporters created from $($) to $($) -params 5$comp-dagaevdv_0$0$s1bb_0$4$ -text 5 reporters c
reated from comp-dagaevdv_0(0) to s1bb_0(4)
Syslog 5604 -msg_id 167 01/07/22 10:28:18.657 -severity INFO -user_id 0 -host_ind 0 -operation DszarRepServ.Restart -mo
dule DsRepServ -message DsRepServ task (re)started old_status=$ periods=0 $ load_pct=$ -params 5$10$80$ -text DsRepServ
task (re)started old_status=5 periods=0 10 load_pct=80
Syslog 5605 -msg_id 168 01/07/22 10:31:43.658 -severity INFO -user_id 0 -host_ind 0 -operation DszarRepServ.InitReport
-module DsRepServ -message $ in ($,$) order:$ artype:$ globname:$ -params DprepArch$01/07/22 10:31:43.658$01/07/22 10:31
```

Рисунок 8 – Фрагмент распечатки глобала Syslogs

На рисунке видно, как разные модули `DsArchServ` и `DsRepServ` делают записи в смежные области с индексами 5201-5600 и 5601-6000 соответственно. Распечатку с опцией `fld t` следует делать для контроля сообщений определенного модуля отдельно от остальных.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1 – СОСТАВ ЗАДАЧ И УТИЛИТ СКАДИ

В таблице 1-1 приведен состав задач СКАДИ, запускаемые рантайм-менеджером.

Имя	Описание	Положение	Назначение
DsSync	Синхронизация данных	все	обновление табличных данных узлов по очередям сообщений
DsDVP	Очереди синхронизации	все	прием/передачу данных синхронизации узлов по сетевому интерфейсу
DsPersist	Сохранение данных	сервер	сброс пользовательских данных в постоянное хранение
DsPinger	Контроль сетей	сервер	Периодическое пингование адресов из набора
DsWrap	Обработчик сигналов	сервер	обработка сигналов переменных процесса
DsGroup	Групповое отображение	все	подготовка данных для группового отображения
DsAlarm	Сигнализации	сервер	подсистема регистрации и управления сигнализациями
DsArchServ	Архивация	арх сервер	ПО для архивации текущих данных
DsRepServ	Протоколирование	сервер	ПО для доступа к архивным и текущим данным
DsXServ	Сервер обмена	сервер	сервер обмена данными
DsHMI	Отображение данных	клиент	Вьюер ЧМИ СКАДИ

Таблица 1-1 – Состав запускаемых задач СКАДИ

В таблице 1-2 приведен состав автономных задач СКАДИ, запускаемых как серверные приложения.

Имя	Описание	Положение	Назначение
DsXCH	Агент обмена	автономно	Передача/прием данных для сервера обмена
DsXGate	Шлюз передачи	автономно	Ретрансляция данных к серверу обмена

Таблица 1-2 – Состав автономных задач СКАДИ

Автономные задачи СКАДИ запускаются и контролируются автономно, средствами операционной системы.

В таблице 1-3 приведен состав утилит сервиса и отладки, запускаемых как консольные приложения.

Имя	Описание	Необходимое условие	Назначение
DsView	просмотр общей памяти	Общая память DsRM или DsXCH	Просмотр глобалов или локальных сегментов разделяемой памяти

DsISD	интерактивный отладчик значений	Общая память DsRM	Чтение/запись значений глобалов разделяемой памяти
DsQSD	интерактивный отладчик очередей	Очереди сообщений DsRM или DsXCH	Чтение данных из очередей глобалов или локальных сегментов разделяемой памяти
DsAccess	доступ к функциям управления	Общая память DsRM	Чтение, запись, управление, параметризация системных и пользовательских глобалов
DsArMan	менеджер архивных файлов	Архивные каталоги СКАДИ	Администрирование и просмотр архивов
DsTrs	консольная печать протоколов	Общая память DsRM и таблицы групп и протоколов	Запуск генерации протоколов в виде текстовых файлов по запросу

Таблица 1-3 – Состав утилит сервиса и отладки

Утилиты сервиса и отладки запускаются как консольные приложения. Они осуществляют вывод текстовых данных на консоль или перенаправляются в текстовый файл при использовании соответствующих команд оболочки операционной системы.

ПРИЛОЖЕНИЕ 2 – ОПИСАНИЕ КОНФИГУРАЦИИ DSCONFIG

Конфигурация представляет собой файл вида `DsConfig_<conftype>.cfg`. Данный файл используется рантайм-менеджером DsRM при создании глобалов общей памяти на этапе старта. Конфигурация состоит из одиночных опций и массивов опций. Одиночные опции могут задаваться в командной строке или в конфигурационном файле. Пример одиночной опции.

```
trace_level=940
```

Массивы опций задаются только в конфигурационном файле. Пример массива опций.

```
CONSTA[0].name=pers_tsd_delay_sec
CONSTA[0].val=10
```

Одиночные опции приведены в таблице 2-1. Сокращения в поле «Диап» означают минимальное значение, значение по умолчанию, максимальное значение.

Опция	Тип	Диап	Описание	Примечание
trace_level	I4	-	Уровень трассировки	
ldr_n_enums	I4	-	Размер массива констант Enumerations	Рекомендуется 200
ldr_n_types	I4	-	Размер массива констант Types	Рекомендуется 50
ldr_n_attrs	I4	-	Размер массива констант Attributes	Рекомендуется 500
ldr_task_messages	I4	-	Размер массива констант Task Messages	Рекомендуется 400
ldr_syslogs_persist_sec	I4	-	Интервал сохранения данных Syslogs	Рекомендуется 60

Таблица 2-1 – Список одиночных опций

Массивы опций приведены в таблице 2-2. Все ссылки на таблицы даны для приложения 3.

Опция	Размер	Тип	Описание	Примечание
CONSTA	256	S32+I4	Строка описания константы	Таблица 3-1
TASK	100	Task	Структура описания задания	Таблица 3-4
HOST	256	Host	Структура описания хоста	Таблица 3-6
HGROUP	32	HGroup	Структура описания группы	Таблица 3-5
GLOBAL	512	Global	Структура описания глобала	
XGROUP	32	XGroup	Не используется	Таблица 3-7
XCH	256	Xch	Номера групп глобалов	Таблица 3-8

Таблица 2-2 – Список массивов опций

ПРИЛОЖЕНИЕ 3 - ДАННЫЕ КОНФИГУРИРУЕМЫХ РЕЛЯЦИОННЫХ ТАБЛИЦ

Тип Const предназначен для хранения целочисленных констант. Структура Const – это запись следующего вида, наследуемая от OGeneric.

```
Const* = RECORD (OGeneric)
    val-, vdef-, vmin-, vmax-: Tb.Int32;
END;
```

Список полей структуры приведен в таблице 3-1.

Поле	Наименование	Значения	Примечание
val	Текущее значение	<Число>	Все значение целочисленные
vdef	Значение по умолчанию	<Число>	-1 если не задано
vmin	Минимально возможное значение	<Число>	-1 если не задано
vmax	Минимально возможное значение	<Число>	-1 если не задано

Таблица 3-1. Список полей структуры Const

Значения Const представляют собой именованные целочисленные константы. Константы имеют также диапазон и значение по умолчанию. Доступ к значениям структуры Const осуществляется по именам, заданным в поле name структуры OGeneric.

Тип Characteristic предназначен для хранения характеристик системы взаимодействия. Структура Characteristic - это запись следующего вида, наследуемая от OGeneric.

```
Characteristic* = RECORD (OGeneric)
    parent-: Tb.Int32;
    cname-: Sb.STag;
    descr-: Sb.SDescr;
END;
```

Список полей структуры приведен в таблице 3-2.

Поле	Наименование	Значения	Примечание
parent	Идентификатор родительской характеристики	<Число>	
cname	Имя родительской характеристики	<Имя>	Значения из predetermined
descr	Описание характеристики	<Строка>	

Таблица 3-2. Список полей структуры Characteristic

Значения Characteristic представляют собой именованные характеристики системы взаимодействия. Характеристики определяют физически возможное в данной архитектуре взаимодействие (управление, обновление данных, синхронизацию), подробно описанное в Приложении 6. Доступ к тем или иным возможностям системы характеристиками не определяется. Доступ к значениям структуры Characteristic осуществляется по именам, заданным в поле name структуры OGeneric.

Тип User предназначен для хранения номеров пользователей. Структура User - это запись следующего вида, наследуемая от OGeneric.

```
User* = RECORD (OGeneric)
    chids-: Tb.Int32;
    descr-: Sb.SDescr;
```

END;

Список полей структуры приведен в таблице 3-3.

Поле	Наименование	Значения	Примечание
chids	Набор идентификаторов характеристик	Приложение 6	Множество oid из таблицы Characteristics
descr	Описание пользователя	<Строка>	

Таблица 3-3. Список полей структуры User

Значения User представляют собой пользователей, имеющих набор ролей пользовательского доступа. Доступ к значениям структуры User осуществляется по именам, заданным в поле name структуры OGeneric.

Тип Task предназначен для хранения задач СКАДИ. Структура Task - это запись следующего вида, наследуемая от OGeneric.

```

Task* = RECORD @UNTAG (OGeneric)
  process_id-, program_id-: Tb.Int32;
  exe_path-: Sb.String64;
  command_line-: Sb.String128;
  dependencies-: Sb.STag;
  watch_limit-, watch_count-, restart_limit-,
    restart_count-: Tb.Int32;
  is_disabled-, is_debug-, is_master-, must_start-: BOOLEAN;
  run_at-: Tb.Int32;
  command_result-, task_status-: Tb.Int32;
  update_ms-: Tb.Int32;
  start_time-, stop_time-, current_time-, cycle_period-,
    delta_time-: Tb.Time;
  error_count-, init_trace_level-, trace_level-, init_priority-,
    priority-: Tb.Int32;
  error_time_first-, error_time_last-, process_start-,
    process_end-: Tb.Time;
  current_load-, maximum_load-, cpu_load-,
    statistics_interval-: Tb.Int16;
  write_q-, counter_n-: Tb.Int32;
  for_rcv_q-, rcv_q-, for_send_q-, send_q-, resend_q-: Tb.Int32;

```

END;

Список полей структуры приведен в таблице 3-4.

Поле	Наименование	Значения	Примечание
process_id	Идентификатор процесса	<Число>	getpid()
program_id	Идентификатор задачи	<Роли>	
exe_path	Имя исполняемой задачи	<Строка>	
command_line	Аргументы командной строки	<Строка>	Запуск: <exe_path> <command_line>
dependencies	Зависимости – загруженные задачи	<Строка>	
watch_limit	Предел счетчика исправности	<Число>	
watch_count	Значение счетчика исправности	<Число>	Перезапуск при 0
restart_limit	Предел счетчика рестартов	<Число>	
restart_count	Значение счетчика рестартов	<Число>	Прекращение при 0
is_disabled	Задача отключена	BOOL	

is_debug	Задача в отладке	BOOL	
is_master	Задача является основной	BOOL	
must_start	Задача стартует автоматически	BOOL	
run_at	Уровень запуска	<Число>	
command_result	Результат выполнения команды	<Число>	Код возврата процесса
task_status	Статус задачи	0-OFF 1-INIT 2-LOAD 3-ERROR 4-END 5-WAITING 6-STARTING 7-SINGLE 8-PRIMARY 9-BACKUP 10-FAILOVER	Enum SRV_STAT
update_ms	Время обновления	<Интервал>	Миллисекунд
start_time	Время старта	<Время>	
stop_time	Время останова	<Время>	
current_time	Текущее время	<Время>	
cycle_period	Период цикла	<ИнтервалВрем>	
delta_time	Время обработки	<ИнтервалВрем>	
error_count	Счетчик ошибок	<Число>	
init_trace_level	Начальное значение уровня трассировки	<Число>	
trace_level	Текущее значение уровня трассировки	<Число>	
init_priority	Начальное значение приоритета	<Число>	
priority	Текущее значение приоритета	<Число>	
error_time_first	Время первой ошибки	<Время>	
error_time_last	Время последней ошибки	<Время>	
process_start	Время старта процесса	<Время>	
process_end	Время окончания процесса	<Время>	
current_load	Текущая загрузка процесса	<Число>	
maximum_load	Максимальная загрузка процесса	<Число>	
cpu_load	Загрузка ЦПУ	<Число>	
statistics_interval	Интервал сбора статистики	<Число>	
write_q	Очередь сообщений для записи	<Номер>	Номер глобала
counter_n	Счетчик изменений глобала n	<Число>	
for_recv_q	Счетчик изменений для очереди приема	<Число>	

recv_q	Счетчик изменений очереди приема	<Число>	
for_send_q	Счетчик изменений для очереди передачи	<Число>	
send_q	Счетчик изменений очереди передачи	<Число>	
resend_q	Счетчик изменений очереди повторной передачи	<Число>	

Таблица 3-4. Список полей структуры Task

Значения Task представляют данные, относящиеся к функционированию текущей задачи. Идентификация задачи осуществляется по имени, заданном в поле name структуры OGeneric.

Тип HGroup предназначен для хранения групп хостов. Каждой группе хостов соответствует область обмена внутри ПО СКАДИ. Или, по другому, каждой области обмена соответствует группа задействованной в ней хостов. Структура HGroup (группы хостов) - это запись следующего вида, наследуемая от OGeneric.

```

HGroup* = RECORD (OGeneric)
  task_id-: Tb.Int32;
  multicast_address-: Sb.String32;
  multicast_address_2-: Sb.String32;
  pl_polling-: Sb.String64;
  pl_protocol-: Sb.String64;
  pl_texas-: Sb.String64;
  cu_polling-: Sb.String32;
  cu_protocol-: Sb.String32;
  cu_texas-: Sb.String32;
  cu_texas_params-: Sb.String128;
  protocol-, rtodb_port-, rtodb_port_2-: Tb.Int16;
  netnum-: Tb.Int16;
  hgtype-: Tb.Int16;
  tp_discovery_timeout-: Tb.Int32;
  fix_port-: Tb.Int16;
  max_resend_counter-: Tb.Int16;
  tp_resend_delay-: Tb.Int32;
  max_send_polls-: Tb.Int16;
  so_keepalive-: Tb.Int16;
  so_reuseaddr-: Tb.Int16;
  so_sndbuf-: Tb.Int32;
  so_rcvbuf-: Tb.Int32;
  tp_connect_timeout-: Tb.Int32;
  tp_disconnect_timeout-: Tb.Int32;
  tcp_nodelay-: Tb.Int16;
  listen_backlog-: Tb.Int16;
  max_send_length-: Tb.Int32

```

END;

Список полей структуры приведен в таблице 3-5.

Поле	Наименование	Значения	Примечание
task_id	Номер в таблице Tasks	<Число>	
multicast_address	Адрес групповой передачи	<IP>	
multicast_address2	Адрес групповой передачи 2	<IP>	
pl_polling	Плагин поллинга	<Имя>	Имя динамически загружаемого модуля

pl_protocol	Плагин протокола	<Имя>	Аналогично
pl_texas	Плагин технологического закрытия	<Имя>	Аналогично
cu_polling	Имя поллинга	<Имя>	
cu_protocol	Имя протокола	<Имя>	
cu_texas	Имя закрытия	<Имя>	
cu_texas_params	Параметры закрытия	<Строка>	
protocol	Протокол обмена	1-ICMP 2-UDP 3-FTUDP 4-MUDP 5-FTMUDP 6-TCP 7-FTTCP	Enum PROTOCOL FT Означает дублированное Fault Tolerant расширение
rtdb_port	Порт синхронизации БДРВ	<Число>	
rtdb_port_2	Порт 2 синхронизации БДРВ	<Число>	
netnum	Номер подсети	<Число>	
hgtype	Тип группы хостов	1-PING 2-SYNC 3-DVP	Определяет способ синхронизации
tp_discovery_timeout	Таймаут системы обнаружения	<Время>	
fix_port	Фиксированные порт соединения	<Число>	
max_resend_counter	Максимальное значение счетчика повтора посылок	<Число>	
tp_resend_delay	Задержка повтора посылок	<Время>	
max_send_polls	Максимальное значение посылок поллинга	<Число>	
so_keepalive	Параметр связи keepalive	<Число>	POSIX параметры сокета
so_reuseaddr	Параметр связи reuseaddr	<Число>	Аналогично
so_sndbuf	Параметр связи sndbuf	<Число>	Аналогично
so_recvbuf	Параметр связи sndbuf	<Число>	Аналогично
tp_connect_timeout	Таймаут соединения	<Время>	
tp_disconnect_timeout	Таймаут разрыва соединения	<Время>	
tcp_nodelay	Параметр tcp_nodelay	<Число>	Аналогично
listen_backlog	Параметр listen_backlog	<Число>	Аналогично
max_send_length	Максимальная длина передачи	<Число>	

Таблица 3-5. Список полей структуры HGroup

Значения HGroup представляют собой группы хостов для внутренней синхронизации. Доступ к значениям структуры HGroup осуществляется по именам, заданным в поле name структуры OGeneric.

Тип Host предназначен для хранения хостов, обменивающихся в рамках групп HGroup. Структура Host - это запись следующего вида, наследуемая от OGeneric.

```
Host* = RECORD @UNTAG (OGeneric)
    host_id-, user_id-, old_user_id-: Tb.Int32;
```

```

chids-: Tb.Int32;
in_groups-: Tb.Int32;
ip_address-: Sb.String32;
ip_address_2-: Sb.String32;
watch_limit-, watch_count-, restart_limit-,
    restart_count-: Tb.Int32;
start_time-, current_time-, monitor_time-, version_date-: Tb.Time;
version-: Tb.Int32;
is_std_server-, is_server-, is_master-, by_wdata-: BOOLEAN;
host_status-, data_status-: Tb.Int32;
cpu_load-, net_mask-: Tb.Int16;
ping_state-, ping_state_2-: Tb.Int32;
ping_last-, ping_last_2-: Tb.Time;
master_time-: Tb.Time;
master_id-: Tb.Int32;
comm_status-, comm_status_2-: Tb.Int32;
ta_flags-, ta_flags_2-: Tb.Int32;
ta_trans_flags-, ta_trans_flags_2-: Tb.Int32;
ta_poll_events-, ta_poll_events_2-: Tb.Int32;
ta_link_stat-, ta_link_stat_2-: Tb.Int32;
t_rcv-, last_rcv-: Tb.Time;
n_rcv-, n_rejected-: Tb.Int32;

```

END;

Список полей структуры приведен в таблице 3-6.

Поле	Наименование	Значения	Примечание
host_id	Идентификатор хоста	<Число>	
user_id	Идентификатор пользователя	<Число>	Из таблицы Users
old_user_id	Идентификатор предыдущего пользователя	<Число>	Из таблицы Users
chids	Набор идентификаторов характеристик	Приложение 6	Множество oid из таблицы Characteristics
in_groups	Множество групп хостов	<Множество>	Число представляет собой множество идентификаторов oid
ip_address	IP-адрес	<IP>	
ip_address2	IP-адрес 2	<IP>	
watch_limit	Предел счетчика исправности	<Число>	
watch_count	Значение счетчика исправности	<Число>	Перезапуск при 0
restart_limit	Предел счетчика рестартов	<Число>	
restart_count	Значение счетчика рестартов	<Число>	Прекращение при 0
start_time	Время старта	<Время>	
current_time	Текущее время	<Время>	
monitor_time	Время мониторинга	<Время>	
version_date	Дата обновления	<Время>	
version	Номер версии	<Число>	
is_std_server	Хост является стандартным сервером	BOOL	

is_server	Хост является сервером	BOOL	
is_master	Хост является основной	BOOL	
by_wdata	Синхронизация данными окна	BOOL	
host_status	Статус хоста	0-OFF 1-INIT 2-LOAD 3-ERROR 4-END 5-WAITING 6-STARTING 7-SINGLE 8-PRIMARY 9-BACKUP 10-FAILOVER	Enum SRV_STAT
data_status	Статус данных	<Число>	
cpu_load	Текущая загрузка ЦПУ	<Число>	
net_mask	Текущая маска сети	<Число>	
ping_state	Текущее состояние пинга	<Число>	
ping_state_2	Текущее состояние пинга 2	<Число>	
ping_last	Время последнего пинга	<Время>	
ping_last_2	Время последнего пинга 2	<Время>	
master_time	Время основного сервера	<Время>	
master_id	Идентификатор основного сервера	<Число>	oid таблицы Hosts
comm_status	Статус коммуникации	См host_status	Enum SRV_STAT
comm_status_2	Статус коммуникации 2	См host_status	Enum SRV_STAT
ta_flags	Флаги уровня линка канала	<Число>	Число представляет собой множество флагов
ta_flags_2	Флаги уровня линка канала 2	<Число>	Аналогично
ta_trans_flags	Флаги транспортного уровня канала	<Число>	Аналогично
ta_trans_flags_2	Флаги транспортного уровня канала 2	<Число>	Аналогично
ta_poll_events	События поллинга канала	<Число>	
ta_poll_events_2	События поллинга канала 2	<Число>	
ta_link_stat	Статус линка канала	<Число>	
ta_link_stat_2	Статус линка канала 2	<Число>	
t_recv	Время приема на соquete	<Время>	
last_recv	Время приема от станции	<Время>	
n_recv	Счетчик принятых сообщений	<Число>	
n_rejected	Счетчик отклоненных сообщений	<Число>	

Таблица 3-6. Список полей структуры Host

Значения Host представляют собой описания хостов по отношению к группе обмена и синхронизации. Доступ к значениям структуры Host осуществляется по именам, заданным в поле name структуры OGeneric.

Тип XGroup предназначен для хранения хранения групп хостов, включенных в области связи с агентами обмена. Структура XGroup – это запись следующего вида, наследуемая от OGeneric.

```
XGroup* = RECORD (OGeneric)
  task_id-: Tb.Int32;
  chids-: Tb.Int32;
  multicast_address-: Sb.String32;
  multicast_address_2-: Sb.String32;
  pl_prot_format-: Sb.String64;
  pl_polling-: Sb.String64;
  pl_protocol-: Sb.String64;
  pl_texas-: Sb.String64;
  cu_prot_format-: Sb.String32;
  cu_polling-: Sb.String32;
  cu_protocol-: Sb.String32;
  cu_texas-: Sb.String32;
  cu_texas_params-: Sb.String128;
  protocol-: Tb.Int16;
  netnum-: Tb.Int16;
  max_ndatas-, nack_history-: Tb.Int32;
  tp_ping_timeout-: Tb.Int32;
  enable_ping-: BOOLEAN;
  update_parm-: BOOLEAN;
  with_crc_ref-: BOOLEAN;
  tp_discovery_timeout-: Tb.Int32;
  fix_port-: Tb.Int16;
  max_resend_counter-: Tb.Int16;
  tp_resend_delay-: Tb.Int32;
  max_send_polls-: Tb.Int16;
  so_keepalive-: Tb.Int16;
  so_reuseaddr-: Tb.Int16;
  so_sndbuf-: Tb.Int32;
  so_rcvbuf-: Tb.Int32;
  tp_connect_timeout-: Tb.Int32;
  tp_disconnect_timeout-: Tb.Int32;
  tcp_nodelay-: Tb.Int16;
  listen_backlog-: Tb.Int16;
  max_send_length-: Tb.Int32;
  pl_usercom-: Sb.String64;
  cu_usercom-: Sb.String32;
  params_usercom-: Sb.String256;
END;
```

Список полей структуры приведен в таблице 3-7.

Поле	Наименование	Значения	Примечание
task_id	Номер в таблице Tasks	<Число>	
chids	Набор идентификаторов характеристик	Приложение 6	Множество oid из таблицы Characteristics
multicast_address	Адрес групповой передачи	<IP>	
multicast_address2	Адрес групповой передачи 2	<IP>	
pl_prot_format	Плагин логического формата протокола	<Имя>	Имя динамически загружаемого модуля
pl_polling	Плагин поллинга	<Имя>	Аналогично
pl_protocol	Плагин протокола	<Имя>	Аналогично

pl_tezas	Плагин технологического закрытия	<Имя>	Аналогично
cu_prot_format	Имя логического формата протокола	<Имя>	
cu_polling	Имя поллинга	<Имя>	
cu_protocol	Имя протокола	<Имя>	
cu_tezas	Имя закрытия	<Имя>	
cu_tezas_params	Параметры закрытия	<Строка>	
protocol	Протокол обмена	1-ICMP 2-UDP 3-FTUDP 4-MUDP 5-FTMUDP 6-TCP 7-FTTCP	Enum PROTOCOL FT Означает дублированное Fault Tolerant расширение
netnum	Номер подсети	<Число>	
max_ndatas	Максимальное число структур Data в сообщении	<Число>	
nack_history	История несквитированных сигналов	<Число>	
tp_ping_timeout	Таймаут посылки пингования	<Время>	В рамках протокола PROTOCOL
enable_ping	Включение пингования	BOOL	
update_parm	Обновление параметров	BOOL	
with_crc_ref	С контрольными суммами	BOOL	
tp_discovery_timeout	Таймаут системы обнаружения	<Время>	
fix_port	Фиксированные порт соединения	<Число>	
max_resend_counter	Максимальное значение счетчика повтора посылок	<Число>	
tp_resend_delay	Задержка повтора посылок	<Время>	
max_send_polls	Максимальное значение посылки поллинга	<Число>	
so_keepalive	Параметр связи keepalive	<Число>	POSIX параметры сокета
so_reuseaddr	Параметр связи reuseaddr	<Число>	Аналогично
so_sndbuf	Параметр связи sndbuf	<Число>	Аналогично
so_recvbuf	Параметр связи sndbuf	<Число>	Аналогично
tp_connect_timeout	Таймаут соединения	<Время>	
tp_disconnect_timeout	Таймаут разрыва соединения	<Время>	
tcp_nodelay	Параметр tcp_nodelay	<Число>	Аналогично
listen_backlog	Параметр listen_backlog	<Число>	Аналогично
max_send_length	Максимальная длина передачи	<Число>	
pl_usercom	Плагин пользовательского протокола	<Имя>	Имя динамически загружаемого модуля

cu_usercom	Имя пользовательского протокола	<Имя>	
params_usercom	Параметры подключения пользовательского протокола	<Строка>	

Таблица 3-7. Список полей структуры XGroup

Значения XGroup представляют собой группы хостов для связи с агентами обмена. Константы имеют также диапазон и значение по умолчанию. Доступ к значениям структуры XGroup осуществляется по именам, заданным в поле name структуры OGeneric.

Тип Xch предназначен для хранения хостов агентов обмена, обменивающихся в рамках групп XGroup. Структура Xch - это запись следующего вида, наследуемая от OGeneric.

```

Xch* = RECORD (OGeneric)
  xgroup-: Tb.Int32;
  serv_host_id-: Tb.Int32;
  area_id-, station_id-: Tb.Int32;
  ip_address-: Sb.String32;
  ip_address_2-: Sb.String32;
  rtdb_port-, rtdb_port_2-: Tb.Int16;
  comm_status-, comm_status_2-: Tb.Int32;
  ta_flags-, ta_flags_2-: Tb.Int32;
  ta_trans_flags-, ta_trans_flags_2-: Tb.Int32;
  ta_poll_events-, ta_poll_events_2-: Tb.Int32;
  ta_link_stat-, ta_link_stat_2-: Tb.Int32;
  t_recv-, last_recv-: Tb.Time;
  n_recv-, n_rejected-: Tb.Int32;
  n_datas-, n_used-: Tb.Int32;
  srvstatus-: Tb.Int16;
  data_err-, last_data_err-: Tb.Int16;
  detail-: Tb.Int32;
  flags-: Tb.Int32;
  clock-, srvclock-, lastclock-: Tb.Time;
  params_usercom-: Sb.String256;
END;

```

Список полей структуры приведен в таблице 3-8.

Поле	Наименование	Значения	Примечание
xgroup	Идентификатор группы хостов	<Число>	Из таблицы XGroup
serv_host_id	Идентификатор сервера	<Число>	Из таблицы Hosts
area_id	Идентификатор области обмена	<Число>	Область обмена Area
station_id	Номер станции	<Число>	Station для данной Area
ip_address	IP-адрес	<IP>	
ip_address2	IP-адрес 2	<IP>	
rtdb_port	Порт синхронизации БДРВ	<Число>	
rtdb_port_2	Порт 2 синхронизации БДРВ	<Число>	
comm_status	Статус коммуникации	См Host.host_status	Enum SRV_STAT

comm_status_2	Статус коммуникации 2	См Host.host_status	Enum SRV_STAT
ta_flags	Флаги уровня линка канала	<Число>	Число представляет собой множество флагов
ta_flags_2	Флаги уровня линка канала 2	<Число>	Аналогично
ta_trans_flags	Флаги транспортного уровня канала	<Число>	Аналогично
ta_trans_flags_2	Флаги транспортного уровня канала 2	<Число>	Аналогично
ta_poll_events	События поллинга канала	<Число>	
ta_poll_events_2	События поллинга канала 2	<Число>	
ta_link_stat	Статус линка канала	<Число>	
ta_link_stat_2	Статус линка канала 2	<Число>	
t_recv	Время приема на соquete	<Время>	
last_recv	Время приема от станции	<Время>	
n_recv	Счетчик принятых сообщений	<Число>	
n_rejected	Счетчик отклоненных сообщений	<Число>	
n_datas	Полное количество данных	<Число>	
n_used	Количество использованных данных	<Число>	
srvstatus	Статус сервера	<Число>	
data_err	Ошибка получения данных	<Число>	
last_data_err	Последняя ошибка получения данных	<Число>	
detail	Детализация ошибки данных	<Число>	
flags	Флаги очереди	<Число>	Число представляет собой множество флагов, относящиеся к очереди сообщений
clock	Время сообщения Xch	<Время>	
srvclock	Время хоста Xch	<Время>	
lastclock	Время последнего сообщения Xch	<Время>	
params_usercom	Параметры подключения пользовательского протокола	<Строка>	

Таблица 3-8. Список полей структуры Xch

Значения Xch представляют собой описания хостов агентов обмена по отношению к группе XGroup. Доступ к значениям структуры Xch осуществляется по именам, заданным в поле name структуры OGenetic.

Тип CSet предназначен для хранения групп, отображаемых в экранные области видеокладов. Каждому объекту соответствует группа для данной области, номера

программы, имени хоста. Структура CSet - это запись следующего вида, наследуемая от OGeneric.

```

CSet* = RECORD @UNTAG (OGeneric)
  ndpr-, maxdpr-, ngrp-, maxgrp-: Tb.Int32;
  icell-, max_cell-: Tb.Int32;
  small-: BOOLEAN;
  dispcmd-: Sb.String256;
  disp_req-: Sb.String256;
  inips-: Sb.String256;
  subscr_id-, subscr_crt-: Tb.Int32;
  colnames-: Sb.String256;
  cds-: Sb.String256;
  cds_id-, cds_crt-: Tb.Int32;
  cp-, fp-, np-: Tb.Int32;
  n_found-, pagesize-, n_cols-: Tb.Int32;
  nxt-: Tb.Int32;
  more_pages-: BOOLEAN;
  res-: Tb.Int32;
  fst-, fen-: Tb.Time;

```

END;

Список полей структуры приведен в таблице 3-9.

Поле	Наименование	Значения	Примечание
ndpr	Номер программы-вьюера	<Число>	
maxdpr	Число программ-вьюеров	<Число>	ndpr<maxdpr
ngrp	Номер группы визуализации	<Число>	
maxgrp	Число групп визуализации	<Число>	ngrp<maxgrp
icell	Идентификатор первой ячейки	<Число>	
max_cell	Число ячеек в наборе	<Число>	
small	Признак уменьшенного набора	BOOL	
dispcmd	Команда текущего отображения	<Строка>	
disp_req	Заданная команда текущего отображения	<Строка>	Установленное задание на отображение видеокadra
inips	Параметры инициализации группы	<Строка>	
subscr_id	Идентификатор требуемой подписки	<Число>	
subscr_crt	Идентификатор текущей подписки	<Число>	
colnames	Набор имен столбцов	<Строка>	
cds	Команда установки данных	<Строка>	Фиксированный перечень команд, обрабатываемый групповым отображением
cds_id	Идентификатор требуемой команды установки	<Число>	

cds_crt	Идентификатор текущей команды установки	<Число>	
cp	Текущая страница	<Число>	
fp	Первая страница	<Число>	
np	Число страниц	<Число>	
n_found	Число найденных записей для группы	<Число>	
pagesize	Заданный размер страницы	<Число>	
n_cols	Число столбцов	<Число>	
nxt	Следующая запрашиваемая страница	<Число>	
more_pages	Есть еще страницы	BOOL	
res	Результат обработки задания подписки	<Число>	0 или код ошибки
fst	Время начала	<Время>	
fen	Время окончания	<Время>	

Таблица 3-9. Список полей структуры CSet

Значения CSet представляют собой параметры отображаемых групп. Доступ к значениям структуры CSet осуществляется по именам, заданным в поле name структуры OGeneric.

Тип Cell предназначен для хранения записей, входящих в группы CSet. Структура Cell - это запись следующего вида, наследуемая от OGeneric.

```
Cell* = RECORD (OGeneric)
    icset-: Tb.Int32;
    crec-: VariableRef;
    rfs-: ARRAY @UNTAG N_CELL_MAX_COLS OF VariableRef;
END;
```

Список полей структуры приведен в таблице 3-10.

Поле	Наименование	Значения	Примечание
icset	Идентификатор группы CSet	<Число>	Номер в таблице CSet
crec	Ссылка на текущую запись	<Ссылка>	
rfs	Массив ссылок на значения	<Массив Ссылок>	Ссылки на значения переменных в столбцах CSet

Таблица 3-10. Список полей структуры Cell

Значения Cell представляют собой ячейки, в которых хранятся относящиеся к группам записи. Доступ к значениям структуры Cell осуществляется по именам, заданным в поле name структуры OGeneric.

Тип VFile предназначен для файлов разделенного доступа, состоящих из переменных/параметров. Структура VFile - это запись следующего вида, наследуемая от OGeneric.

```
VFile* = RECORD (OGeneric)
    fname-: Sb.STag;
    descr-: Sb.SDescr;
    ftype-: Tb.Int32;
    n_vars-: Tb.Int32;
END;
```

Список полей структуры приведен в таблице 3-11.

Поле	Наименование	Значения	Примечание
fname	Имя файла	<Имя>	
descr	Описание файла	<Строка>	
ftype	Тип файла	<Число>	1-VFILE 2-EKP 3-ALARM 4-DATABOOK 5-TS 6-DOV
n_vars	Количество относящихся к файлу переменных	<Число>	

Таблица 3-11. Список полей структуры VFile

Значения VFile представляют собой именованные наборы переменных/параметров, условно названные файлами. Доступ к значениям структуры VFile осуществляется по именам, заданным в поле name структуры OGeneric.

Тип FVar предназначен для хранения переменных/параметров, относящихся к файлам VFile. Структура FVar - это запись следующего вида, наследуемая от OGeneric.

```
FVar* = RECORD (OGeneric)
    vref-: VariableRef;
    params-: Sb.String256;
    lf-: Tb.Real32;
    hf-: Tb.Real32;
```

END;

Список полей структуры приведен в таблице 3-12.

Поле	Наименование	Значения	Примечание
vref	Ссылка на элемент файла	<Ссылка>	Ссылка на переменную процесса или объект
params	Параметры, устанавливаемые для переменной	<Строка>	Например, требуемые значения
lf	Нижняя граница	<Предел>	Действительное число или ~ при отсутствии
hf	Верхняя граница	<Предел>	Аналогично

Таблица 3-12. Список полей структуры FVar

Значения FVar представляют собой переменные/параметры, относящиеся к файлам. Доступ к значениям структуры FVar осуществляется по именам, заданным в поле name структуры OGeneric.

Тип Arch предназначен для хранения типов архивов. Структура Arch - это запись следующего вида, наследуемая от OGeneric.

```
Arch* = RECORD (OGeneric)
    nglobal-: Tb.Int32;
    cglobal-: Tb.Int32;
    afmt-: Tb.Int16;
    aclass-: Tb.Int16;
    artype-: Tb.Int16;
    n_files-: Tb.Int16;
    t_hh-: Tb.Int32;
    t_mm-: Tb.Int32;
    t_ss-: Tb.Int32;
    delta_t-: Tb.Int32;
    aperture-: Tb.Real32;
```

```

period-: Tb.Int32;
time_lim-: Tb.Int32;
update_time-: Tb.Time;
record_time-: Tb.Int32;
n_updated-: Tb.Int32;
n_recorded-: Tb.Int32;
update_res-: Tb.Int32;
record_res-: Tb.Int32;
mp_fileno-: Tb.Int32;
mp_mpos-: Tb.Int32;

```

END;

Список полей структуры приведен в таблице 3-13.

Поле	Наименование	Значения	Примечание
nglobal	Номер архивного глобала	<Число>	Глобал а
cglobal	Номер заголовочного глобала архива	<Число>	Глобал с
afmt	Формат файла архива	0-нет 1-fta 2-fba 3-foa	fta – текстовый формат архива; fba – двоичный формат архива foa – двоичный формат оперативного архива
aclass	Класс реализации алгоритма архивации	0-нет 1-QUEUE 2-OLD 3-CALC 4-LOG 5-PERIOD	QUEUE – архив по очередям сообщений; OLD – архив изменений значения/качества; CALC – архив расчетных параметров; LOG – архив сигнализаций; PERIOD – непрерывный архив
artype	Тип архива	0-нет 1-NGLOB 2-SYSLOG 3-OPER 4-STA 5-LTA 6-TRA	NGLOB – текущие данные глобала n; SYSLOG – архив системных событий; OPER – оперативный архив; STA – среднесрочный архив; LTA – долговременный архив; TRA – аварийный архив
t_hh	Время в часах	<Число>	
t_mm	Время в минутах	<Число>	
t_ss	Время в секундах	<Число>	
delta_t	Интервал времени	<Число>	
aperture	Апертура по умолчанию	<Значение>	
period	Период непрерывной архивации	<Число>	
time_lim	Обеспечиваемый предел времени	<Число>	
update_time	Время формирования архивной записи	<Время>	
record_time	Время сброса архивной записи на носитель	<Время>	
n_updated	Количество сформированных данных	<Число>	

n_recorded	Количество записанных данных	<Число>	
update_res	Результат формирования архива	<Число>	
record_res	Результат записи архива на носитель	<Число>	
mp_fileno	Номер файла позиции	<Число>	Адрес записи: (fileno,mpos)
mp_mpos	Номер записи позиции	<Число>	Адрес записи: (fileno,mpos)

Таблица 3-13. Список полей структуры Arch

Значения Arch представляют собой параметры, относящиеся к архивным файлам в целом. Доступ к значениям структуры Arch осуществляется по именам, заданным в поле name структуры OGeneric.

Тип Gstat предназначен для хранения ролей с правами пользователей. Структура Gstat - это запись следующего вида, наследуемая от OGeneric.

```
Gstat* = RECORD (SO.OGeneric)
  globtype-: Tb.Char8;
  t_modified-, t_saved-: Tb.Time;
  task_modified-, task_saved-, persist_interval-,
  persist_count-: Tb.Int32;
  done_saved-, parm_modified-: BOOLEAN;
  refresh_counter-: Tb.Int32;
  in_groups-, in_xgroups-, sync_class-: Tb.Int32;
  snd_done-, snd_missed-, snd_redone-: Tb.Int32;
  rcv_done-, rcv_missed-, rcv_redone-, rcv_lost-,
  rcv_err-: Tb.Int32;
  ncadr_sum-, netcadr_sum-: Tb.Int32;
  ncadr_last-, cnt_val_sync-: Tb.Int32;
  chids-: Tb.Int32;
END;
```

Список полей структуры приведен в таблице 3-14.

Поле	Наименование	Значения	Примечание
globtype	Тип глобала	<Символ>	Перечень типов приведен в таблице 1
t_modified	Время модификации	<Время>	
t_saved	Время сохранения	<Время>	
task_modified	task_id модифицирующей задачи	<Число>	Номер в таблице Tasks
task_saved	task_id сохраняющей задачи	<Число>	Аналогично
persist_interval	Интервал сохранения в файл	<Число>	
persist_count	Счетчик сохранения в файл	<Число>	
done_saved	Сохранение выполнено	BOOL	
parm_modified	Параметр модифицирован	BOOL	
refresh_counter	Счетчик обновлений	<Число>	
in_groups	Входит во множество серверных групп	<Число>	Число представляет собой множество идентификаторов групп

in_xgroups	Входит во множество групп агентов обмена	<Число>	Аналогично
sync_class	Класс синхронизации	<Число>	
snd_done	Счетчик отправленных	<Число>	
snd_missed	Счетчик пропущенных при отправке	<Число>	
snd_redone	Счетчик повторенных при отправке	<Число>	
rcv_done	Счетчик принятых	<Число>	
rcv_missed	Счетчик пропущенных при приеме	<Число>	
rcv_redone	Счетчик повторенных при приеме	<Число>	
rcv_lost	Счетчик потерянных при приеме	<Число>	
rcv_err	Ошибка приема	<Число>	Согласно таблице кодов ошибок
ncadr_sum	Суммарный номер кадра	<Число>	
netcadr_sum	Сетевой номер кадра	<Число>	
ncadr_last	Последний номер кадра	<Число>	
cnt_val_sync	Значение счетчика синхронизации	<Число>	
chids	Набор идентификаторов характеристик	Приложение 6	Множество oid из таблицы Characteristics

Таблица 3-14. Список полей структуры Gstat

Значения Gstat представляют собой именованные роли пользовательского доступа. Доступ к значениям структуры Gstat осуществляется по именам, заданным в поле name структуры OGeneric.

Gstat предоставляет программам дополнительную информацию о глобалах, доступных из самих глобалов. Следует отметить, программа, получающая доступ к глобалам, уже имеет информацию общего характера о подключенных глобалах.

Тип Alarm предназначен для хранения значений сигнализаций. Структура Alarm - это запись следующего вида, наследуемая от OGeneric.

```
Alarm* = RECORD @UNTAG (OGeneric)
    e16-: E16B;
    vref-: VariableRef;
    timestamp-: Tb.Time;
    t_modified-: Tb.Time;
END;
```

Список полей структуры приведен в таблице 3-15.

Поле	Наименование	Значения	Примечание
e16	Структура сигнализации	<Сигнализация>	
vref	Ссылка на элемент, вызвавший сигнализацию	<Ссылка>	Ссылка на переменную объекта со значением
timestamp	Метка времени сигнализации	<Время>	Время возникновения
t_modified	Время изменения	<Время>	Меняется при любой модификации

Таблица 3-15. Список полей структуры Alarm

Значения Alarm представляют собой массив сигнализаций, разбитый на области, в зависимости от условий выборки. Доступ к значениям структуры Alarm осуществляется по индексам, соответствующим областям выборки.

Тип Repdata предназначен для хранения данных отчетов в текстовом и бинарном виде. Структура Repdata - это запись следующего вида, наследуемая от OGeneric.

```
Repdata* = RECORD @UNTAG (OGeneric)
  atype-: Sb.VarType;
  srvstatus-: Tb.Int16;
  source_id-, subscr_id-, flags-: Tb.Int32;
  srec-: StringSRec;      (* 272 bytes *)
END;
```

Список полей структуры приведен в таблице 3-16.

Поле	Наименование	Значения	Примечание
atype	Тип атрибута	<Тип>	
srvstatus	Статус сервера	<Число>	Enum SRV_STAT
source_id	Идентификатор источника данных	<Число>	
subscr_id	Идентификатор подписки	<Число>	
flags	Флаги передачи	<Число>	Множество
srec	Данные текстовые или двоичные		

Таблица 3-16. Список полей структуры Repdata

Значения Repdata представляют собой текстовые или бинарные данные, в зависимости от значения atype, что иллюстрируется таблицей 3-17.

Имя	Значение	Описание
SG.SEQ_DATA	-1	Каждая структура Repdata содержит 1-6 записей SG.ARecSeq (Приложение 11), начиная от адреса Repdata.atype
SG.SEQ_NACK	-2	
SG.SEQ_TEXT	-3	Поле Repdata.srec представляет собой отформатированный текст протокола
SG.SEQ_PROT_PAGE	-4	Разрыв страницы протокола
SG.SEQ_PROT_END	-5	Окончание данных протокола
--	0	Нет данных
	>0	Данные бинарной последовательности

Таблица 3-17. Список значений Repdata.atype

Доступ к значениям структуры Repdata осуществляется по индексам, соответствующим строкам протокола.

Тип Report предназначен для хранения управляющих данных отчетов, по записи на каждый отчет. Структура Report - это запись следующего вида, наследуемая от OGeneric.

```
Report* = RECORD @UNTAG (OGeneric)
  cset_id-, task_id-: Tb.Int32;
  irepdata-, max_repdata-: Tb.Int32;
  inips-: Sb.String256;
  rep_id-, rep_crt-: Tb.Int32;
  irow-, ystate-: Tb.Int32;
  n_found-: Tb.Int32;
```



```

page_rec_max-, page_rec_count-, pages_max-, pages_count-:
Tb.Int32;
nset_id, nset_crt-: Tb.Int32;
pct_done-: Tb.Int32;
res-: Tb.Int32;
st-, en-: Tb.Time;

```

END;

Список полей структуры приведен в таблице 3-18.

Поле	Наименование	Значения	Примечание
cset_id	Номер записи CSet	<Число>	
task_id	Имя задачи	<Число>	Динамически присваивается по нагрузке
irepdata	Номер записи Repdata	<Число>	
max_repdata	Максимальное количество записей Repdata	<Число>	
rep_id	Идентификатор подписки	<Число>	subscr_id в структуре CSet
rep_crt	Идентификатор текущей подписки	<Число>	subscr_crt
irow	Память строки протокола	<Число>	При многостраничном останове
istate	Память состояния протокола	<Число>	При многостраничном останове
n_found	Число найденных строк	<Число>	
page_rec_max	Максимальное число строк на странице	<Число>	
page_rec_count	Число строк на странице	<Число>	
pages_max	Максимальное число страниц	<Число>	
pages_count	Число страниц	<Число>	
nset_id	Идентификатор набора страниц	<Число>	
nset_crt	Идентификатор текущего набора страниц	<Число>	
pct_done	Процент готовности	<0-100>	
res	Результат выполнения	<Число>	0, <0 при ошибке
st	Время начала	<Время>	
en	Время окончания	<Время>	

Таблица 3-18. Список полей структуры Report

Значения Repdata представляют собой массив отчетов, разбитый на области, определяющие состояние отчета. Каждый отчет связан с элементом CSet и частью массива Repdata.

ПРИЛОЖЕНИЕ 4 – ДИНАМИЧЕСКИ ЗАГРУЖАЕМЫЕ МОДУЛИ

Список динамически загружаемых модулей СКАДИ приведен в таблице 4-1. Фабричные модули определяют интерфейс, который запрашивается у плагина с проверкой соответствия версии.

Фабрика	Версия	Реализация	Описание	Примечание
DsdVltFuns*	VltFunctions 101,1.0	DpdVltfStd	Набор функций-обработчиков	DsHMI.cfg: vltf_lib_name
DsdScrEvals*	ScriptEvaluators 102,1.0	DpdUsrScreva	Расширенные скриптовые функции	DsHMI.cfg: screva_lib_name
DshmOVHandlers	ObjectValueHandlers 103,1.0	DprmOvhAI	Обработка аналоговых сигналов	TASK[].command_line=-hl DprmOvhAI
		DprmOvhDI	Обработка дискретных сигналов	TASK[].command_line=-hl DprmOvhDI
		DprmOvhNI	Обработка диагностических сигналов	TASK[].command_line=-hl DprmOvhNI
DsxchXHandlers	XchHandlers 104,1.0	DpxchHndImit	Плагин данных обмена	HND.plugin=DpxchHndImit
DsxchProtFactory	ProtFactory 105,1.0	DpxchMUDP2Channel	Плагин реализации протокола	XGROUP[.].pl_protocol=DpxchUDP2Channel
DsxchUsercomFactory	UserCommunication 106,1.0	DpxchUctest	Плагин реализации пользовательского механизма коммуникации	XGROUP[0].pl_usercom=DpxchUctest
DsdDlgrBase*	DlgrWidgets 107,1.0	DpdgButtons	Загружаемые модули визуализации	Тип Ldgrp dov-файла
DsdDlgrBase*	DlgrTrends 108,1.0	DpdgTrends	Загружаемые модули трендов	Тип Ldtrend dov-файла
DshmTypers	ReportTypers 109,1.0	DprepArch	Отчет по выборке из архива по времени	Запрос группового отображения с именем протокола: -report DprepArch
DshmPermits*	Permits 110,1.0	DpermHMI	Плагин предоставления доступа	Всегда DpermHMI

Таблица 4-1 Загружаемые модули СКАДИ

ЛИСТ ИЗМЕНЕНИЙ

Изм.	Номера листов				Всего листов в измененном документе	Номер документа, утверждающего изменение	Подп.	Дата
	измененных	замененных	новых	аннулированных				