

**ОИК «СИСТЕЛ». Чтение данных АИИС УЭ
«ПИРАМИДА Сети»»
Руководство администратора
Листов 21**

АННОТАЦИЯ

Данный документ содержит описание программы «ОИК «СИСТЕЛ». Чтение данных из АИИС УЭ «ПИРАМИДА Сети» (далее – ПИРАМИДА).

В состав документа входят общие сведения о программе, порядок ее установки и настройки. Так же документ включает описание способов проверки работоспособности программы.

СОДЕРЖАНИЕ

ОБЩИЕ СВЕДЕНИЯ	4
УСТАНОВКА ПРОГРАММЫ	5
АЛГОРИТМ РАБОТЫ ПРОГРАММЫ.....	7
КОНФИГУРИРОВАНИЕ ИНТЕГРАЦИОННОГО МОДУЛЯ	8
ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММЫ.....	11
Прослушивание порта подключения Пирамиды	12
Служебная База данных.....	12
Архивная База данных.....	16
База данных Прогнозирования.....	19

ОБЩИЕ СВЕДЕНИЯ

Программа «ПИРАМИДА» предназначена для обеспечения информационного взаимодействия ОИК «СИСТЕЛ» с ПО «Пирамида Сети». Данная программа является интеграционным модулем, предназначенным для получения данных АИИС УЭ из ПК «Пирамида-Сети» производства АО группы компаний «СИСТЕМЫ И ТЕХНОЛОГИИ» по стандарту МЭК 61968-9. ПИРАМИДА содержит в своем составе один запускаемый файл с единственной зависимостью от библиотек Postgres.

УСТАНОВКА ПРОГРАММЫ

Установка программы состоит из трех этапов:

компиляция исполняемого файла из исходных кодов;

создание структур и баз данных в СУБД Postgres;

создание сервиса Linux, выполняющего автоматическую загрузку программы.

Все три этапа выполняются запуском скрипта `install.sh` с единственным аргументом `build`

Также перед запуском скрипта необходимо заполнить файл `install_config.json`, в данном файле конфигурируются параметры создаваемых структур и баз данных. Этот файл расположен в той же директории, что и `install.sh`.

Структура файла `install_config.json` приведена в таблице 1.

Таблица 1 - Поля конфигурационного файла инсталлятора `install_config.json`

ARCHIVE				
	host		строка	ip сервера БД
	port		целое число	порт сервера БД
	dbname		строка	имя архивной БД
	user		строка	имя пользователя для подключения к БД
	password		строка	пароль пользователя для подключения к БД
	schema		строка	схема БД, в которой находятся все таблицы
	tablespace		строка	пространство таблиц
TECHNOLOGY				
	host		строка	ip сервера БД
	port		целое число	порт сервера БД
	dbname		строка	имя технологической БД
	user		строка	имя пользователя для подключения к БД
	password		строка	пароль пользователя для подключения к БД

	schema		строка	схема БД, в которой находятся все таблицы
	tablespace		строка	пространство таблиц
FORECAST				
	host		строка	ip сервера БД
	port		целое число	порт сервера БД
	dbname		строка	имя БД модуля прогнозирования
	user		строка	имя пользователя для подключения к БД
	password		строка	пароль пользователя для подключения к БД
	schema		строка	схема БД, в которой находятся все таблицы
	tablespace		строка	пространство таблиц

В данном файле указываются данные для подключения к СУБД и схемы, в которых будет происходить развертывание интеграционного модуля внутри СУБД.

Внутри СУБД будут развернуты следующие части интеграционного модуля:

1) Внутри архивной базы данных (раздел ARCHIVE): хранимая процедура `add_to_arch(integer, integer)`

2) Внутри базы данных модуля прогнозирования (раздел FORECAST): хранимая процедура `add_to_arch()` и таблицы: `measurement`, `measurement_type` и `point`

3) Внутри технологической базы данных (раздел TECHNOLOGY): таблицы `tb_connectclosetime`, `tb_datasize`, `tb_db2discnttime`, `tb_dbconnecttime`, `tb_dbinserttime`, `tb_elemscount`, `tb_events`, `tb_srv_avbl`, `tb_connectopentime`, `tb_db2connecttime`, `tb_db2inserttime`, `tb_dbdiscnttime`, `tb_description`, `tb_endparse`, `tb_ip`, `tb_startparse`

Создание сервиса Linux

В комплекте поставки присутствует файл `converter.service` Он нужен для создания сервиса `systemd`.

Данный файл имеет следующее содержимое:

```
[Install]
```

```
WantedBy=multi-user.target
```

```
[Unit]
```

```
Description=converter
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/bin/bash /work/zemon/converter_prod/start.sh
```

```
ExecStop=/bin/bash /work/zemon/converter_prod/stop.sh
```

```
WorkingDirectory=/work/zemon/converter_prod/build
```

```
User=root
```

Программа должны быть также запущена вручную скриптом `install.sh` с параметром `run`

АЛГОРИТМ РАБОТЫ ПРОГРАММЫ

Программа в основном режиме работы производит ожидание подключения клиента «Пирамида Сети» на предмет получения получасовых данных потребления электроэнергии каждой точкой учета. В случае отсутствия соединения с БД обработка данных, полученных от ПК «Пирамида Сети» не проводится. Программа принимает все данные, затем ожидает следующего подключения от П ПК «Пирамида Сети» для добавления новых значений. Время ожидания задается в конфигурационном файле. По истечении этого

времени программа «ПИРАМИДА» отправляет все присланные в последних подключениях данные в архивную базу данных и базу данных модуля прогнозирования. В случае если в архивной базе данных уже имеются данные за некоторый период, то интеграционный модуль (в зависимости от настроек конфигурационного файла) может сохранить повторно присланные данные с, указанными в конфигурационном файле индексами, региона и сервера. Это производится в случае, когда в очередной посылке от ПК «Пирамиды Сети» будут присутствовать точки.

КОНФИГУРИРОВАНИЕ ИНТЕГРАЦИОННОГО МОДУЛЯ

Структура конфигурационного файла

Интеграционный модуль конфигурируется с помощью файла config.json, который расположен в директории build корневой директории программы. В таблице 2 приведено описание каждого параметра конфигурационного файла.

Таблица 2 - Поля конфигурационного файла config.json

SERVER				
	ip		строка	Прослушиваемый ip, если у системы есть несколько интерфейсов с различными ip, то чтобы принимать соединения со всех интерфейсов, следует указать в данном поле "0.0.0.0"
	port		целое число	Прослушиваемый порт
	tries		целое число	Количество попыток занять порт, если прослушиваемый порт занят, то программа будет пытаться занять его еще раз
	tries_delay		целое число	Интервал между попытками занять порт
ARCHIVE				
	regionindex		целое число	Индекс региона, с которым в архивную базу данных будут добавляться измерения, информации о которых нет в базе

	regionindex _duplicate		целое число	Индекс региона, с которым в архивную базу данных будут добавляться измерения, информация о которых уже есть в базе.
	serverindex		целое число	Индекс сервера, с которым в архивную базу данных будут добавляться данные
	host		строка	ip сервера архивной БД
	port		целое число	Порт сервера архивной БД
	dbname		строка	Имя архивной БД
	user		строка	Имя пользователя для подключения к архивной БД
	password		строка	Пароль пользователя для подключения к архивной бд
	schema		строка	Схема архивной БД, в которой находятся все таблицы
TECHNOLOGY				
	host		строка	ip сервера технологической БД
	port		целое число	Порт сервера технологической БД
	dbname		строка	Имя технологической БД
	user		строка	Имя пользователя для подключения к технологической БД
	password		строка	Пароль пользователя для подключения к технологической БД
	schema		строка	Схема технологической БД, в которой находятся все таблицы
FORECAST				
	host		строка	ip сервера БД модуля прогнозирования
	port		целое число	Порт сервера БД модуля прогнозирования
	dbname		строка	Имя БД модуля прогнозирования
	user		строка	Имя пользователя для подключения к БД модуля прогнозирования
	password		строка	Пароль пользователя для подключения к БД модуля прогнозирования
	schema		строка	Схема БД, в которой находятся все таблицы

				модуля прогнозирования
ASSETS				
	filename		строка	Имя файла, в который будут сохраняться встреченные точки учета (в формате csv: первый столбец - серийный номер, второй - свободный индекс для сохранения в бд) Если данный файл не указать (""), то данный файл сохраняться не будет.
	assets		массив	Структура данных, описывающая каждую точку учета, для которой нужно сохранить измерения в архивную бд (если не указать точку учета, то измерения по ней не сохранятся)
		id	целое число	Индекс, который участвует в формировании индекса dataindex (формируется из id параметра и id точки учета следующим способом: dataindex =<id точки учета> * 1000 + <id параметра>)
		name	строка	Серийный номер точки учета
	parameters		массив	Массив номиналов физических величин, по которым сохраняются измерения
		id	целое число	id параметра, который участвует в формировании индекса dataindex: dataindex =<id точки учета> * 1000 + <id параметра>
		forecast_id	целое число	Псевдоним id параметра, который соответствует id параметра в БД модуля прогнозирования
		uid	строка	Уникальный идентификатор параметра
		name	строка	Наименование параметра
DUMP_ARCHIVE				
	filename		строка	Временный файл буфера данных, в котором копятся данные до сохранения в архивную базу данных. Рекомендуется сохранять данных файл на RAM-дисках.
	delay		целое число	Задержка перед отправкой всех данных буфера в архивную БД (выражена в

				миллисекундах)
	mode		строка	Режим добавления данных в архивную БД ("REPLACE"- получасовка будут заменяться, "STRAIGHT" - получасовки дописываются с новым индексом региона и сервера)
DUMP_FORECAST				
	filename		строка	Временный файл буфера данных, в котором копятся данные до сохранения в базу данных модуля прогнозирования. Рекомендуется сохранять данных файл на RAM-дисках.
	delay		целое число	задержка перед отправкой всех запросов в БД модуля прогнозирования (выражена в миллисекундах)

Добавление точек учета в конфигурационный файл

Все новые точки учета будут перечислены в файле, имя которого было задано в секции ASSETS конфигурационного файла (см. таблицу 2). Если потребуется добавить новую точку учета, то нужно просто добавить объект в массив assets секции ASSETS конфигурационного файла и перезапустить сервис. Каждая точка учета добавляется в конфигурационный файл в виде строки: { "id": 435, "name": "1504992" }

Значение поля dataindex формируется из id параметра и id точки учета - $\langle \text{id точки учета} \rangle * 1000 + \langle \text{id параметра} \rangle$.

Id точки учета не может иметь минимальное значение 1.

ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММЫ

Для проверки работоспособности программы вначале необходимо проверить статус запуска сервиса командой: `$ systemctl status converter.service`

Внимание! Сервис должен быть активным.

Прослушивание порта подключения Пирамиды

Чтобы удостовериться в том, что интеграционный модуль прослушивает порт требуется ввести команду: \$ ss -tunlp

Если в выводе данной команды будет присутствовать заданный в текущей конфигурации порт, то это свидетельствует о том, что программа запущена и требуемый порт прослушивается.

Служебная База данных

Для логирования всех событий ниже приведена служебная БД (см. таблицу 3).

Таблица 3 - Перечень отношений технологической Базы данных

Отношение	Атрибут	Тип данных (postgres)	NULL	Скрипт Генерации
tb_events	id	integer	NOT NULL	<pre>DROP TABLE IF EXISTS public.tb_events CASCADE; CREATE TABLE public.tb_events(ID integer NOT NULL, Time timestamp without time zone NOT NULL, Status smallint , Details varchar(512));</pre>
	time	timestamp without time zone	NOT NULL	
	status	smallint		
	details	varchar (512)		

В данном отношении приводится описание случившихся событий (time - время, когда произошло событие, status - код события, ставящий в соответствие описание из таблицы tb_description).

tb_description	id	smallint	NOT NULL	<pre>DROP TABLE IF EXISTS public.tb_description CASCADE; CREATE TABLE public.tb_description(ID smallint NOT NULL, Description varchar(256) NOT NULL);</pre>
	description	varchar (256)	NOT NULL	

Отношение дает описание каждому виду события

tb_ip	sessionid	integer	NOT NULL	<pre>DROP TABLE IF EXISTS public.TB_IP CASCADE; CREATE TABLE public.TB_IP(SessionID integer NOT NULL, IP character(15) NOT NULL, Port integer NOT NULL);</pre>
	time	timestamp without time zone	NOT NULL	
	ip	character (15)	NOT NULL	
	port	integer	NOT NULL	

Отношение содержит данные по подключениям, id сессии, время открытия сессии, ip и порт подключившегося

tb_connectopentime	sessionid	integer	NOT NULL	<pre> DROP TABLE IF EXISTS public.TB_ConnectOpenTime CASCADE; CREATE TABLE public.TB_ConnectOpenTime(SessionID integer NOT NULL, Milliseconds integer NOT NULL DEFAULT 0, Time timestamp without time zone NOT NULL); </pre>
	milliseconds	integer	NOT NULL DEFAULT 0	
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени подключения к модулю

tb_connectclosetime	sessionid	integer	NOT NULL	<pre>DROP TABLE IF EXISTS public.TB_ConnectCloseTime CASCADE; CREATE TABLE public.TB_ConnectCloseTime(SessionID integer NOT NULL, Time timestamp without time zone NOT NULL);</pre>
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени отключения от комплекса

tb_startparse	sessionid	integer	NOT NULL	<pre>DROP TABLE IF EXISTS public.tb_startparse CASCADE; CREATE TABLE public.tb_startparse(SessionID integer NOT NULL, time timestamp without time zone NOT NULL);</pre>
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени начала обработки присланных Пирамидой данных

tb_endparse	sessionid	integer	NOT NULL	<pre>DROP TABLE IF EXISTS public.tb_endparse CASCADE; CREATE TABLE public.tb_endparse(SessionID integer NOT NULL, time timestamp without time zone NOT NULL);</pre>
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени окончания обработки присланных Пирамидой данных

tb_dbconnecttime	sessionid	integer	NOT NULL	<pre>DROP TABLE IF EXISTS public.TB_DBConnectTime CASCADE; CREATE TABLE public.TB_DBConnectTime(SessionID integer NOT NULL, Time timestamp without time zone NOT NULL);</pre>
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени подключения к архивной БД

tb_dbinserttime	sessionid	integer	NOT NULL	<pre>DROP TABLE IF EXISTS public.TB_DBInsertTime CASCADE; CREATE TABLE public.TB_DBInsertTime(SessionID integer NOT NULL, RcrdCnt integer NOT NULL, Time timestamp without time zone NOT NULL);</pre>
	rcrdcnt	integer	NOT NULL	
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени окончания вставки значений в архивную БД и началу работы хранимой процедуры

tb_dbdiscnttime	sessionid	integer	NOT NULL	DROP TABLE IF EXISTS public.TB_DBDiscntTime CASCADE; CREATE TABLE public.TB_DBDiscntTime(SessionID integer NOT NULL, Time timestamp without time zone NOT NULL);
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени отключения от архивной БД

tb_srv_avbl	time	timestamp without time zone	NOT NULL	DROP TABLE IF EXISTS public.TB_Srv_Avbl CASCADE; CREATE TABLE public.TB_Srv_Avbl(Time timestamp without time zone NOT NULL, Status smallint);
	status	smallint		

Отношение содержит данные о статусе работы программы

tb_db2connecttime	sessionid	integer	NOT NULL	DROP TABLE IF EXISTS public.TB_DB2ConnectTime CASCADE; CREATE TABLE public.TB_DB2ConnectTime(SessionID integer NOT NULL, Time timestamp without time zone NOT NULL);
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени подключения к БД модуля прогнозирования

tb_db2inserttime	sessionid	integer	NOT NULL	DROP TABLE IF EXISTS public.TB_DB2InsertTime CASCADE; CREATE TABLE public.TB_DB2InsertTime(SessionID integer NOT NULL, RcrdCnt integer NOT NULL, Time timestamp without time zone NOT NULL);
	rcrdcnt	integer	NOT NULL	
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени окончания вставки значений в БД модуля прогнозирования и началу работы хранимой процедуры

tb_db2discnttime	sessionid	integer	NOT NULL	DROP TABLE IF EXISTS public.TB_DB2DiscntTime CASCADE; CREATE TABLE public.TB_DB2DiscntTime(SessionID integer NOT NULL, Time timestamp without time zone NOT NULL);
	time	timestamp without time zone	NOT NULL	

Отношение содержит данные по времени отключения от БД модуля прогнозирования

tb_datasize	sessionid	integer	NOT NULL	DROP TABLE IF EXISTS public.tb_datasize CASCADE; CREATE TABLE public.tb_datasize(SessionID integer NOT NULL, size integer NOT NULL);
	size	integer	NOT NULL	

Отношение содержит данные по размеру присланных Пирамидой посылок (берется только тела ответа)

tb_elemscount	sessionid	integer	NOT NULL	DROP TABLE IF EXISTS public.tb_elemscount CASCADE; CREATE TABLE public.tb_elemscount(SessionID integer NOT NULL, count integer NOT NULL);
	count	integer	NOT NULL	

Отношение содержит данные по количеству присланных Пирамидой полчасовок

Архивная База данных

Выходные данные заносятся в таблицу DsvValueTavle. В поле DataIndex заносится значение, описывающее параметр, сохраненный в таблице. Значение этого поля формируется из значений полей конфигурационной таблицы. Значение параметра сохраняется в поле Value. При этом можно использовать таблицу DayCatalog.

В таблице «DayCatalog» каждая запись в архив отмечается временной меткой, которая представляет собой значение времени в стандарте UTC с точкой отсчета «00.00.00 01.01.1970». Таким образом, временная метка архивной записи – это количество секунд, прошедших от указанной точки отсчета, с учетом поясного, зимнего, летнего времени.

При создании записи каждой временной метке сопоставляется индекс, значение которого вычисляется по следующей формуле:

(текущий год – 2000)х400 + номер дня в году.

Например, значение «1945» означает «10 декабря 2004 года».

Для хранения временных меток и соответствующих им индексов база данных «SystelArchive» включает таблицу «DayCatalog», описание полей которой приведено в таблице 4.

время записи в архив в секундах в формате UTC с точкой отсчета «00.00.00 01.01.1970» с учетом часового пояса

Таблица 4 – Описание полей таблицы «DayCatalog»

<i>Имя поля</i>	<i>Назначения поля</i>
«DayIndex»	индекс даты
«DayDate»	дата записи в архив
«DayBegin»	время записи в архив в секундах в формате UTC с точкой отсчета «00.00.00 01.01.1970» с учетом часового пояса

В таблице DsvValueTable начало каждой даты есть dayindex и время UTC, начиная с 1970 года в секундах (время это местное, то есть, 0 означает 3 часа ночи 1 января 1970 года).

Всю информацию по получасовкам модуль сохраняет в таблицу DsvValueTable архивной БД.

Таблица 5 - Описание полей таблицы «DsvValueTable»

Имя поля	Тип значения
dayindex	smallint
seconds	integer
regionindex	integer
dataindex	integer
value	double precision
status	integer
serverindex	integer

Значение поля dataindex формируется из id параметра и id точки учета - $\langle \text{id точки учета} \rangle * 1000 + \langle \text{id параметра} \rangle$ Другими словами, алгоритм формирования значения поля DataIndex таблицы _DsvValueTable следующий: первые три младших разряда десятичного представления содержат порядковый номер хранимого параметра:

120 - параметр, интегрируемый в направлении перетока / Энергия А+ за 30 минут;

121 - параметр, интегрируемый в направлении перетока / Энергия А- за 30 минут;

122 - параметр, интегрируемый в направлении перетока / Энергия Р+ за 30 минут;

123 - параметр, интегрируемый в направлении перетока / Энергия Р- за 30 минут.

Следующие старшие разряды поля DataIndex хранят порядковый номер прибора учета.

Порядковые номера приборов учета и их индексы, применяемые в ПО «Пирамида Сети», в модуле будут содержаться в таблице соответствия в конфигурационном файле.

Индексы в ПО «Пирамида Сети» текстовые, они содержат расшифровку с именем прибора учета и его серийным номером.

База данных Прогнозирования

Таблица	Атрибут	Тип данных (postgres)	NULLABLE	Описание
Точки учёта				
point	id	integer	NOT NULL	Уникальный идентификатор
	caption	character varying(80)	NOT NULL	Полное название точки учета в интерфейсе пользователя
	created_at	timestamp(0) without time zone	NOT NULL	Время добавления точки учета
	serial_number	text		Серийный номер точки учета
	asdu_name	text		Имя точки учета
	text	text		Название точки учета присланное пирамидой
	measurement_type_ids	integer		Внешнее поле связи с таблицей аномалий
	measurement_type_id	integer		Внешнее поле связи с таблицей point_extended
	substation	character varying(80)		Название подстанции
	transformer	character varying(80)		Название трансформатора
Аномалии				
anomaly	id	integer	NOT NULL	Идентификатор
	point_id	integer		Идентификатор точки учёта, для которой определена аномалия
	lost_value	integer		Количество пропущенных значений.
	overload_count	integer		Количество выбросов
	measurement_type_id	integer	NOT NULL	Физическая величина, по которой зафиксирована аномалия.

Данные измерений и прогнозов.

measurement	id	integer		NOT NULL	Уникальный идентификатор
	created_at	timestamp zone	without time		Время получения посылки от пирамиды
	updated_at	timestamp zone	without time		Время обновления измерения
	point_id	integer			Внешний ключ к таблице точек учёта, указывающий на точку, с которой было произведено измерение
	measurement_type_id	integer			Внешний ключ к таблице типов измерений, указывающий на тип полученного измерения
	value	precision		NOT NULL	Полученное значение измерения
	hour	integer			В системах собирающих данные с приборов учета из разных часовых поясов, обозначает часовой пояс
	timetag	timestamp(0) zone	without time	NOT NULL	Время измерения точкой учета

Классификатор физических величин

measurement_type	id	integer		NOT NULL	Уникальный идентификатор
	caption	character varying(80)			Название типа измерения
	unit	character varying(80)			Единицы исчисления
	guid	text			Уникальный идентификатор
	created_at	timestamp(0) zone	without time	NOT NULL	Время создания типа измерения

Прогнозная модель				
model	id	integer	NOT NULL	Уникальный идентификатор
	point_id	integer		Идентификатор точки учёта, для которой построена модель.
	last_start	timestamp(0) without time zone		Последний запуск обучения
	learning_dimension	integer		Обучающая выборка
	mean_day_error	numeric(5,2)		Средняя ошибка за сутки
	mean_week_error	numeric(5,2)		Средняя ошибка за неделю
	start_period	character varying		Период запуска
	deterioration_limit	numeric(5,2)		Предел ухудшения
	progress	numeric(5,2)		Прогресс
	seconds_remaining	integer		Шкала прогресса
Дополнительные параметры точек учета				
point_extended	point_id	integer	NOT NULL	Уникальный идентификатор внешний ключ
	accuracy	numeric(5,2)		Точность
	is_blocked	boolean	NOT NULL	Признак блокировки
	is_abnormal	boolean	NOT NULL	Признак наличия аномалии.