ПРОТЕЙ

Мультисервисный коммутатор доступа PROTEI mCore.SSW5

Подсистема AlarmProcessor: Переменные и SNMP-трапы Руководство пользователя

Содержание

1 Общие сведения

Документ содержит описание переменных и трапов подсистемы Alarm Processor Мультисервисного коммутатора доступа PROTEI mCore.SSW5 (далее — SSW5). Также приведены рекомендации действий обслуживающего персонала при получении Alarm-сообщений, информирующие об изменении состояния оборудования или нарушения его работоспособности.

1.1 Техническая поддержка

Техническая поддержка, а также дополнительное консультирование по вопросам, возникающим в процессе установки и эксплуатации изделия, осуществляются производителем и службой технической поддержки.

1.1.1 Производитель

ООО «НТЦ ПРОТЕЙ»

194044, Санкт-Петербург

Большой Сампсониевский пр., д. 60, лит. А

Бизнес-центр «ТЕЛЕКОМ»

Тел.: (812) 449-47-27

Факс: (812) 449-47-29

WEB: http://www.protei.ru

E-mail: info@protei.ru

1.1.2 Служба технической поддержки

ООО «НТЦ ПРОТЕЙ»

194044, Санкт-Петербург

Большой Сампсониевский пр., д. 60, лит. А

Бизнес-центр «ТЕЛЕКОМ»

Тел.: (812) 449-47-27 доп. 5999 (круглосуточно)

Факс: (812) 449-47-29

WEB: http://www.protei.ru

E-mail: mak.support@protei.ru, support.mak@protei.ru

1.2 Термины и сокращения

Используемые в настоящем документе термины и сокращения приведены в таблице 1.

Таблица 1 — Используемые термины и сокращения

Термин/сокращение	Расшифровка
CA	(Component-Address) адрес компоненты
CDR	(Call Detail Record), подробная запись о вызове
CgPA	(Calling Party Address), адрес вызывающего абонента
CgPN	(Calling Party Number), номер вызывающего абонента
MCU	(Microcontroller Unit) Микроконтроллер
MIB	(Management Information Base) Административная база данных
MKD	Мультисервисный коммутатор доступа
PBX	(Private Branch Exchanges) Учрежденческо-производственная автоматическая телефонная станция
SNMP	(Simple Network Management Protocol) простой протокол сетевого управления
МКД	Мультисервисный коммутатор доступа
ПО	Программное обеспечение

1.3 История изменений

История изменений настоящего документа фиксируется в таблице 2.

Таблица 2 — История изменений

Дата	Версия документа	Изменения
09.06.2015	1.0.0	Первая версия документа
15.03.2021	1.1.0	Добавлена таблица терминов и сокращений. Добавлена переменные состояния регистрации/авторизации абонентских устройств на МКД (Таблица 10). Общее форматирование документа.

2 Назначение и основные свойства SSW5

SSW5 — это программный коммутатор, относящийся к оборудованию операторского класса и используемый в качестве управляющего узла городских, сельских или производственных телефонных сетей.

Основная задача программных коммутаторов (Softswitch) — это управление вызовами — поиск и предоставление вызывающей стороне информации о точках соединения, используя которую оборудование вызывающего и вызываемого абонентов будет способно установить соединение.

SSW5 поддерживает базовые услуги и широкий набор дополнительных услуг (переадресация, постановка на ожидание и т.д.), включая контроль доступа абонентов к местной/ междугородной/международной телефонной связи.

2.1 Основные сведения о подсистеме Alarm Processor

Подсистема Alarm Processor — это подсистема мониторинга состояния аппаратных и логических ресурсов SSW5.

В подсистеме Alarm Processor реализовано два способа предоставления информации:

- по запросу оператора (синхронный способ);
- посылка Alarm-сообщения при возникновении события (асинхронный способ).

Подсистема Alarm Processor взаимодействует с SNMP-менеджером, установленным на компьютере оператора, с использованием протокола SNMP. SNMP-менеджер предоставляет оператору удобные средства взаимодействия с подсистемой Alarm Processor и наглядные способы отображения состояния контролируемого оборудования. В качестве SNMP-менеджера может быть использована любая существующая программа, поддерживающая протокол SNMP.

Для SNMP-протокола любое оборудование представляет собой набор переменных, через которые SNMP-менеджер получает информацию о состоянии оборудования, и изменяет характер поведения оборудования через изменения значения этих переменных.

В SSW5 набор SNMP-переменных является отражением компонентной архитектуры программного обеспечения. Все программные компоненты в SSW5 имеют иерархическую зависимость между собой, образуя в совокупности дерево компонентов. Каждый компонент имеет уникальное имя (адрес), представляющее собой путь от корня дерева к компоненту, состоящее из массива слов, разделенных точками.

Каждый компонент в подсистеме Alarm Processor может зарегистрировать свой набор переменных. Переменные, зарегистрированные компонентом, делятся на два типа: обычная переменная и траповая переменная.

При изменении значения обычной переменной со стороны SSW5, никакие события не активизируются. Просмотр обычных переменных происходит по запросу оператора, т.е. синхронным способом.

Траповая переменная способна активизировать событие (трап) при изменении своего значения. Траповая переменная посылается по SNMP-протоколу на компьютер оператора асинхронно, т.е. по факту появления данного события. Трап имеет свойство — приоритет, отображающий важность события. Другой свойство трапа — идентификатор. Идентификатор трапа — это уникальное число в пределах SSW5. По идентификатору трапа SNMP-менеджер выполняет поиск трапа в mib-файле, если поиск завершился успешно, считывается информация о трапе, далее SNMP-менеджер выполняет действия согласно данной информации (например, определяет каким цветом вывести сообщение

о трапе на экран, или какие действия предпринять на компьютере оператора: подача звукового сигнала, вывод окна с информационным сообщением и т.д.).

Для того, чтобы различать обычные и траповые переменные, в составе свойств переменных имеется свойство - «индикатор трапа». Индикатор трапа — это флаг, который установлен в единицу, если переменная является траповой.

Адреса переменных образуются из адреса компонента, которому принадлежит переменная и имени самой переменной. Например, адрес переменной, содержащей компонентный адрес MCU «MKD.Mod_MCU.MCU;CA».

Адрес переменной для передачи по протоколу SNMP имеет несколько иную форму — это набор чисел, разделенных точками, например, «1.2.3.100.1». Поэтому перед отправкой переменной SNMP-менеджеру, ее адрес преобразуется из компонентного адреса в SNMP-адрес. Правила преобразования из компонентного адреса в SNMP-адрес находятся в файле конфигурации /usr/protei/Protei-MKD/MKD/config/Alarm/ap.cfg (/home/protei/Protei-MKD/MKD/config/Alarm/ap.cfg), содержащий, кроме этого, и другие параметры настройки подсистемы Alarm Processor.

2.2 Использование протокола SNMP в подсистеме Alarm Processor

Простой протокол сетевого управления (SNMP) – это протокол управления компонентами сети. Протокол SNMP позволяет выполнять мониторинг текущего состояния отдельных компонентов сети, а также позволяет выполнять изменение параметров компонентов сети, изменяя таким образом характер поведения данных сетевых компонентов.

Основная идея протокола SNMP — это то, что мониторинг состояния сетевого компонента и управление им, производятся через набор переменных, хранимых в самом устройстве, - в Административной Базе Данных (MIB). Например, для того чтобы проконтролировать состояние сетевого компонента, необходимо получить доступ к его MIB, и проанализировать значения интересующих переменных. Таким образом снимается зависимость протокола SNMP от конкретной реализации оборудования, делая его универсальным средством

На текущий момент протокол SNMP является фактически стандартом при реализации процесса мониторинга состояния сетевого оборудования и управления его параметрами. Использование протокола SNMP подсистемой Alarm Processor в SSW5 обеспечило совместимость со множеством существующих программ, предназначенных для управления сетевыми компонентами (SNMP-менеджеры). Одной из наиболее известных таких программ является программа SNMPc Castle Rock.

В SSW5 для протокола SNMP не поддержана возможность управления через изменение значений переменных, реализован только мониторинг состояния логических и аппаратных ресурсов.

Использование протокола SNMP для мониторинга состояния SSW5, дает следующие возможности:

- Получение в режиме реального времени состояния аппаратных и логических ресурсов;
- Посылка уведомительных сообщений (trap) при изменении состояния аппаратных или логических ресурсов;
- Настройка условий формирования уведомительных сообщений;
- Использование программ SNMP-менеджеров сторонних разработчиков;

Полный SNMP-адрес переменной можно разделить на две части: идентификатор предприятия изготовителя сетевого компонента и идентификатор переменной в пределах

сетевого компонента. Идентификатор предприятия-изготовителя является отражением глобальной иерархической структуры и будет неизменным для всех продуктов производства ООО «НТЦ «Протей». Идентификатор переменной в пределах сетевого компонента является отражением иерархической структуры аппаратных и логических ресурсов конкретного компонента. Таким образом достигается уникальность SNMP-адреса переменной. Т.е., любая SNMP-переменная является частью дерева SNMP-переменных в пределах сетевого компонента, это дерево является частью глобального дерева.

Идентификацией предприятия в составе глобального дерева занимается специальная международная организация, определяющая SNMP-адрес корневого узла дерева переменных для данного предприятия. Для ООО «НТЦ Протей» определен SNMP-адрес корневого узла дерева переменных — «1.3.6.1.4.1.20873».

3 Настройка подсистемы Alarm Processor

Для настройки подсистемы Alarm Processor существуют два файла:

- ap.cfg — файл конфигурации подсистемы Alarm Processor;
- ap_dictonary — словарь подсистемы Alarm Processor.

Файлы ap.cfg и ap_dictonary находятся в разделах /usr/protei/Protei-MKD/MKD/config/Alarm/ или /home/protei/Protei-MKD/MKD/config/Alarm/ в зависимости от дистрибутива.

Файл ap.cfg содержит параметры подсистемы Alarm Processor, параметры SNMP-соединения и правила преобразования компонентных адресов в SNMP-адреса.

В файле ap_dictonary находятся соответствия между значениями переменных и идентификаторами трапов. Идентификаторы трапов используются SNMP-менеджером для соответствующей их обработки.

Пример файла конфигурации ap.cfg приведен в «Приложении».

3.1 Файл конфигурации ap.cfg

В таблице 3 приведено описание секций, из которых состоит файл ap.cfg.

Таблица 3 — Состав секций файла ap.cfg

Имя секции	Описание
General	Основные параметры.
AtePath2ObjName	Правила преобразования компонентного адреса переменной в SNMP-адрес.
SNMP	Сетевые параметры протокола SNMP.
StandartMib	Объекты стандартного MIB-а.
SNMPTrap	Правила посылки трапов.
Filter	Правила фильтрации Alarm-сообщений.
SpecificTrapCA_Object	Соответствие идентификатора трапа адресу компонента.
SpecificTrapCT_Object	Соответствие идентификатора трапа типу компонента (CT).
SpecificTrapCA_Var	Соответствие идентификатора трапа компонентному адресу переменной.
Logs	Параметры ведения журналов подсистемой Alarm Processor.

В таблице 4 описаны параметры секции **[General]**.

Параметры секции **[General]** не требует какого-либо редактирования.

Таблица 4 — Параметры секции **[General]**

Имя параметра	Описание
ApplicationAddress	Адрес приложения. Значение по умолчанию — «MKD.201» Параметр изменять не рекомендуется.
MaxConnectionCount	Максимальное кол-во одновременных подключений к AP_Agent. Значение по умолчанию — 100.
ManagerThread	Флаг установки способа запуска подсистемы Alarm Processor. Возможные значения: 0 — в главном потоке приложения; 1 — в отдельном потоке. Значение по умолчанию — 0.
CyclicWalkTree	Флаг установки циклического обхода дерева компонентов. Значение по умолчанию — 1. Параметр изменять не рекомендуется.

Значение параметра ManagerThread зависит от загрузки процессора. Если загрузка процессора достаточно большая, то подсистему Alarm Processor рекомендуется запускать в отдельном потоке (ManagerThread=1).

Секция **[AtePath2ObjName]** содержит правила преобразования компонентных адресов переменных в SNMP-адреса.

Правило преобразования состоит из двух частей, разделенные символом «;»: правила преобразования адреса компонента в SNMP-адрес и соответствие имени переменной SNMP-идентификатору.

Пример правила преобразования:

```
{MKD(600).Mod_MCU(3).MCU(1,1);CA(4096);};
```

Правило преобразования, приведенное в примере выше — «{MKD(600).Mod_MCU(3).MCU(1,1);CA(4096);};», определяет преобразование из компонентного адреса «MKD.Mod_MCU.MCU.CA» в SNMP-адрес «600.3.1.1.4096».

Необходимо отметить, что при передаче по сети, к этому SNMP-адресу всегда будет добавляться SNMP-адрес корневого узла дерева переменных, о котором говорилось в пункте 2.2.

Секция **[SNMP]** определяет сетевые параметры протокола SNMP:

- ListenIP — IP-адрес SSW5;
- ListenPort — порт, используемый протоколом SNMP (по умолчанию — 161);
- OwnEnterprise — ветвь в глобальном SNMP-дереве для продуктов ООО «НТЦ Протей» (этот параметр имеет постоянное значение — 1.3.6.1.4.1.20873).

В секции **[StandardMib]** определяется список стандартных переменных и их значений.

Формат записи, описывающей стандартную переменную:

```
{<SNMP-адрес переменной>;<тип ответа>;<ответ>;};
```

Секция **[SNMPTrap]** определяет параметры взаимодействия с SNMP-менеджерами. Подсистема Alarm Processor может взаимодействовать с несколькими SNMP-менеджерами одновременно. Для каждого SNMP-менеджера в секции **[SNMPTrap]** можно определить свои параметры.

Формат записи секции **[SNMPTrap]:**

```
{<IP-адрес SNMP-менеджера>;<порт SNMP-менеджера>;<фильтр компонентного адреса>;<фильтр типа компонента>;<фильтр компонентного адреса переменной>;};
```

Секция **[Filter]** определяет фильтры по адресу компонентов-источников трапов, по типу компонентов-источников трапов, по компонентному адресу переменной. Эти фильтры «отсеивают» трапы на входе подсистемы Alarm Processor, т.е. между логикой, которая является источником трапов и подсистемой Alarm Processor.

В таблице 5 ниже приведено описание параметров секции **[Filter]**.

Таблица 5 — Параметры секции **[Filter]**

Имя параметра	Описание
CA_Object	Фильтр по адресу компонента-источника трапа.
CT_Object	Фильтр по типу компонента-источника трапа.
CA_Var	Фильтр по адресу переменной.
TrapIndicator	Фильтр по индикатору трапа.
DynamicIndicator	Фильтр по индикатору динамического объекта.

Пример секции **[Filter]**, в котором разрешены трапы от всех компонентов, имеющие любой тип, адрес переменной может быть любым:

```
CA_Object=".*";
CT_Object=".*";
CA_Var=".*";
```

3.2 Алгоритм формирования идентификатора трапа

В формировании идентификатора трапа участвует информация, содержащаяся в файлах ap.cfg и ap_dictonary.

При возникновении какого-либо события подсистема Alarm Processor получает адрес компонента источника события и переменную, связанную с событием. Далее, используя информацию файлов ap.cfg и ap_dictonary, Подсистема Alarm Processor вычисляет значение идентификатора трапа. Файл ap.cfg предоставляет информацию для вычисления базового значения идентификатора трапа, ap_dictonary — смещения. В условной форме формулу вычисления идентификатора трапа можно записать:

trap_id = ap.cfg * 1000 + ap_dictonary

Эту запись можно интерпретировать так: значение, найденное в файле ap.cfg умножается на 1000, и к этому результату прибавляется значение, найденное в файле ap_dictonary.

Поиск в файле ap.cfg выполняется путем последовательного просмотра секций: **[SpecificTrapCA_Object]** — поиск по адресу компонента источника события в секции **[SpecificTrapCT_Object]** — поиск по типу компонента источника события в секции, **[SpecificTrapCA_Var]** — поиск по адресу переменной в секции. Поиск прекращается

на первом попавшемся совпадении. Это означает, что если, например, в секции **[SpecificTrapCA_Object]** будет найден адрес компонента источника события, то поиск в файле ap.cfg прекращается, далее будет выполняться поиск имени переменой в файле ap_dictonary. И наоборот, если результат поиска в секции **[SpecificTrapCA_Object]** отрицательный, то поиск продолжается в секции **[SpecificTrapCT_Object]** по типу компонента источника события, и далее в секции **[SpecificTrapCA_Var]**, если в **[SpecificTrapCT_Object]** нет искомого типа компонента.

Рассмотрим работу алгоритма на конкретном примере. Допустим в файле ap.cfg представлено следующее содержимое секций **[SpecificTrapCT_Object]** и **[SpecificTrapCA_Var]**:

```
[SpecificTrapCA_Var]                # Rules for creating SNMP trap id
{"Calls"; 101;};

[SpecificTrapCT_Object]                    # Rules for creating SNMP trap id
{"MKD.Sys"; 1;};
{"MKD.Mod_MCU.MCU"; 2;};
```

Файл ap_dictonary содержит следующие данные:

```
OSTATE =
{
 1; SP_Trap = 1;
 0; SP_Trap = 2;
 -1; SP_Trap = 3;
};
Alarm.Version =
{
      2; SP_Trap = 4;
      1; SP_Trap = 5;
};
```

Предположим, что в подсистему Alarm Processor от логики поступило событие об изменении оперативного состояния (переменная OSTATE) компоненты с адресом «MKD.Sys». Подсистема Alarm Processor начинает поиск в файле ap.cfg, в секции **[SpecificTrapCA_Object]**. Данная секция в ap.cfg отсутствует. Далее продолжается поиск в секции **[SpecificTrapCT_Object]**. В данной секции обнаруживается запись («{"MKD.Sys"; 1;};» с искомым типом компонента. Эта запись содержит соответствующий идентификатор - «1». На этом поиск в файле ap.cfg прекращается. Найденный идентификатор «1» умножается на «1000» в результате получаем базовое значение идентификатора - «1000». Далее подсистема Alarm Processor выполняет поиск переменной с именем «OSTATE» в файле ap_dictonary. Поиск завершается с положительным результатом — блок «OSTATE». Блок «OSTATE» в примере выше содержит три записи, каждая из которых состоит из значения переменной и соответствующего идентификатора. Результатом поиска будет идентификатор, соответствующий значению переменной «OSTATE». Допустим переменная «OSTATE» имеет значение «-1», ему в примере соответствует идентификатор «3».

В завершение складываем базовое значение идентификатора («1000») с результатом поиска в файле ap_dictonary - «3», получаем «1003». Это и будет итоговое значение идентификатора трапа.

Примечание — Отрицательному результату поиска в файле ap_dictonary соответствует значение «0», т.е. итоговое значение идентификатора трапа будет совпадать с базовым значением, если взять пример выше - «trap_id = 1000 + 0».

4 Использование SNMP-менеджеров

Протокол SNMP, в силу своей универсальности, позволяет использовать для мониторинга сетевого компонента и его управления любой известный SNMP-менеджер. SNMP-менеджер обычно имеет графический интерфейс пользователя, предоставляющий оператору удобные средства контроля текущего состояния сетевого компонента.

Любой SNMP-менеджер способен взаимодействовать с несколькими сетевыми компонентами одновременно. Для того чтобы SNMP-менеджер начал контролировать какой-либо сетевой компонент (например, SSW5), данный сетевой компонент должен быть зарегистрирован в SNMP-менеджере. Дополнительно для данного компонента должен быть создан mib-файл, описывающий переменные и трапы. mib-файл формируется производителем сетевого компонента (в нашем случае это SSW5), и включается в состав программного обеспечения поставляемого изделия. Все SNMP-менеджеры способны отображать содержимое mib-файла: список переменных и трапов и их свойства.

Контролируемый сетевой компонент в большинстве случаев отображается SNMP-менеджером в виде значка (иконки) с подписью. Возможны и другие способы отображения.

SNMP-менеджер при взаимодействии с SSW5 выполняет следующие действия:

- чтение значений переменных по запросу оператора (просмотр переменных);
- прием трапов.

При просмотре значений переменных SNMP-менеджер обычно отображает их в виде иерархического списка (дерево) или в виде таблицы. Дополнительно в SNMP-менеджере может быть реализована возможность представления значений переменной в виде графика. Данное свойство предоставляет достаточно наглядный способ контроля быстро изменяющихся переменных (например, температуры).

Трапы по своим свойствам отличаются от простых переменных. Во-первых, трапы — это асинхронные события. Во-вторых, трапы имеют приоритет (или, другими словами, уровень важности). В-третьих, для трапа можно определить правила его обработки и фильтрации.

В таблице 6 ниже приведен список приоритетов трапов.

Таблица 6 — Приоритеты трапов

Приоритет	Числовое значение
Critical (критичный)	1
Severe (строгий)	2
Major (важный)	3
Minor (низкий)	4
Warning (предупреждение)	5
Normal (нормальный)	6
Info (информационный)	7

SNMP-менеджер отображает информацию, связанную с трапами несколькими способами одновременно. Например, путем изменения цвета иконки сетевого компонента источника трапа, или в виде таблицы, содержащей информацию о пришедших трапах. В зависимости от приоритета трапа, строка в таблице может быть выделена своим цветом.

Как правило, таблица с трапами имеет набор фильтров, позволяющие ограничить выводимую информацию по принадлежности к какому-либо сетевому компоненту, по приоритету трапа или по другим свойствам.

Каждый трап в наборе свойств, определенных в mib-файле, имеет правила его обработки. SNMP-менеджер может изменять правила обработки трапа, определенные в mib-файле, или добавлять свои. Обычно изменения хранятся в отдельном файле. Наиболее частыми видами обработки трапа являются: вызов какой-либо программы, отправка email-сообщения, подача звукового сигнала, вывод всплывающего окна с соответствующим сообщением, занесение трапа в базу данных.

5 Описание переменных и трапов SSW5

5.1 Переменные

Пример. Фрагмент файла с переменными МКД.

```
# System variables of MKD
{ MKD(600).Sys(2,1,1); CA(4096); };
{ MKD(600).Sys(2,1,1); OSTATE(4097); };
{ MKD(600).Sys(2,1,1); Calls(5000); };

{ MKD(600).Sys(2).Period(3,1,1); CA(4096); };
{ MKD(600).Sys(2).Period(3,1,1); NoCgPN(6000);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Count(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Answer(2);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Busy(3);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).NoAns(4);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Rel(5);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Max(6);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Min(7);};
{ MKD(600).Sys(2).Period(3,1,1); Ratio(6002).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).In(1).Count(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).In(1).Fail(2);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Input(2).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Step1(3).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Step3(4).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Step4(5).Err(1);};

{ MKD(600).Sys(2).RTCP(2,1,1); CA(4096); };
{ MKD(600).Sys(2).RTCP(2,1,1); USER(6000);};
{ MKD(600).Sys(2).RTCP(2,1,1); DIRECT(6001);};
{ MKD(600).Sys(2).RTCP(2,1,1); ADDR(6002);};
{ MKD(600).Sys(2).RTCP(2,1,1); FRLOST(6003);};

{ MKD(600).User(7,1,1);CA(4096); };
{ MKD(600).User(7,1,1);RegStat(99); };

# Variables of MCU
{ MKD(600).Mod_MCU(3).MCU(1,1);CA(4096); };
{ MKD(600).Mod_MCU(3).MCU(1,1);OSTATE(4097); };
{ MKD(600).Mod_MCU(3).MCU(1,1);IP_addr(6000); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Port(6001); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Version(6002); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Users(6003).Resrvd(1); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Users(6003).Used(2); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Alarm(7000).Version(1); };

# variables of RD_NAS
{ MKD(600).Mod_RD(4).RD(1,1);CA(4096); };
{ MKD(600).Mod_RD(4).RD(1,1);OSTATE(4097); };

# variables of CDR
{ MKD(600).Mod_CDR(5,1,1);CA(4096); };
{ MKD(600).Mod_CDR(5,1,1);PBX_ID(6000); };
```

В таблицах 7 —13 приведено описание Alarm-переменных SSW5.

Таблица 7 — Alarm-переменные PROTEI.MKD.Sys-Table

Переменная	Тип	Описание
sys-CA (600.2.1.1.4096)	строка	Компонентный адрес модуля CPS.
sys-OSTATE (600.2.1.1.4097)	строка	Траповая переменная. Оперативное состояние модуля CPS: active — нормальная работа модуля CPS. failed — критический сбой в работе модуля CPS. unknown — оперативное состояние модуля CPS неизвестно. (см. трапы 1001, 1002, 1003 в подразделе «Трапы»)
sys-Calls (600.2.1.1.5000)	число	Информационная переменная, для оценки нагрузки на модуль CPS. Показывает количество вызовов, обрабатываемых в данный момент времени модулем CPS. Период обновления - ~500 мс.

Таблица 8 — Переменные счетчиков Sys-Period

Переменная	Тип	Описание
Sys-Period-CA (600.2.3.1.1.4096)	строка	CA = MKD.Sys.Period.* , где * - период, за который отправляется статистика, в секундах. Сейчас статистика отправляется за 5 секунд, 1 минуту, 10 минут и 1 час. Т.е. CA = MKD.Sys.Period.5, MKD.Sys.Period.60, MKD.Sys.Period.600, MKD.Sys.Period.3600. Примечание — За 5-секундные интервалы статистика выдается только по всему МКД и по всем направлениям, без детализации по классам ошибок и без суммарной нагрузки в секундах за период времени. Статистика в МКД ведется по каждому PBX (PBX.*) и всему MKD(ALL) в целом. Для каждого объекта, по которому ведется статистика, она подразделяется по направлениям: To_ALL - на все направления; To_INSIDERS - на инсайдеров; DIRECTION.* - на именованные направления (* - имя направления); DirectionsList - определяет список направлений для объекта по которому ведется статистика. В объекте ALL - перечислены списки направлений для всех PBX системы.

Переменная	Тип	Описание
		(траповая переменная, трап 4101).
Sys-Period-NoCgPN (600.2.3.1.1.6000)	число	Информационная переменная. Пишется за период для МКД и каждого PBX, не детализируется по направлениям. Показывает количество поступивших в систему вызовов без CgPN.
Sys-Period-Call-Out-Count (600.2.3.1.1.6001.1.1)	число	Информационная переменная. Показывает количество попыток вызовов за период времени.
MKD-Sys-Period-Call-Out-Answer (600.2.3.1.1.6001.1.2)	число	Информационная переменная. Показывает количество успешных вызовов за период времени (вызовов с ответом вызываемого абонента).
MKD-Sys-Period-Call-Out-Busy (600.2.3.1.1.6001.1.3)	число	Информационная переменная. Показывает количество вызовов на занятых абонентов за период времени (отбой до ответа с cause=17).
MKD-Sys-Period-Call-Out-NoAns (600.2.3.1.1.6001.1.4)	число	Информационная переменная. Показывает количество вызовов без ответа вызываемого абонента за период времени (вызовы, отбитые стороной А, Б или системой до ответа вызываемого абонента, но после получения Alerting со стороны Б).
MKD-Sys-Period-Call-Out-Rel (600.2.3.1.1.6001.1.5)	число	Информационная переменная. Показывает общее кол-во неуспешных исходящих вызовов.
MKD-Sys-Period-Call-Out-Max (600.2.3.1.1.6001.1.6)	число	Информационная переменная. Показывает пиковое количество одновременных вызовов за период времени.
MKD-Sys-Period-Call-Out-Min (600.2.3.1.1.6001.1.7)	число	Информационная переменная. Показывает минимальное количество одновременных вызовов за период времени.
MKD-Sys-Period-Ratio-Err (600.2.3.1.1.6002.1)	число	Информационная переменная. Показывает отношение неуспешных вызовов ко всему количеству вызовов.
MKD-Sys-Period-Call-In-Count (600.2.3.1.1.6003.1.1)	число	Информационная переменная. Показывает количество попыток входящих на МКД вызовов за период времени.
MKD-Sys-Period-Call-In-Fail (600.2.3.1.1.6003.1.2)	число	Информационная переменная. Показывает общее количество неуспешных входящих вызовов.
MKD-Sys-Period-Call-Input-Err (600.2.3.1.1.6003.2.1)	число	Информационная переменная. Показывает количество раз набора номера вне плана набора.
MKD-Sys-Period-Call-Step1-Err (600.2.3.1.1.6003.3.1)	число	Информационная переменная. Показывает количество раз ошибочного выполнения Step1 (правило маршрутизации вызова).

Переменная	Тип	Описание
MKD-Sys-Period-Call-Step3-Err (600.2.3.1.1.6003.4.1)	число	Информационная переменная. Показывает количество раз ошибочного выполнения Step3 (внутреннее правило маршрутизации вызова на один из шлюзов МКД).
MKD-Sys-Period-Call-Step4-Err (600.2.3.1.1.6003.5.1)	число	Информационная переменная. Показывает количество раз ошибочного выполнения Step4 (внутреннее правило преобразования номеров в необходимый для соединения формат).

Таблица 9 — Переменные, получаемые при превышении порога потерь RTP пакетов у абонента на прием, за отчетный период. Sys-RTCP

Переменная	Тип	Описание
MKD-Sys-RTCP-CA (600.2.2.1.1.4096)	строка	Адрес компоненты MKD_RTCP. Траповая переменная, трап посылается при превышении порога потерь RTP пакетов у абонента на прием. (Трап 5101).
MKD-Sys-RTCP-USER (600.2.2.1.1.6000)	число	Набор чисел. Внешний номер абонента, который перестал принимать пакеты.
MKD-Sys-RTCP-DIRECT (600.2.2.1.1.6001)	строка	Адрес направления, где находится абонент.
MKD-Sys-RTCP-ADDR (600.2.2.1.1.6002)	строка	Адрес абонента.
MKD-Sys-RTCP-FRLOST (600.2.2.1.1.6003)	число	Набор чисел. Доля потерь за отчетный период. Доля потерь – это отношение потерянных пакетов к ожидаемым, умноженное на 256.

Таблица 10 — Переменные состояния регистрации/авторизации абонентских устройств на МКД

Переменная	Тип	Описание переменной
MKD-User-CA (600.7.1.1.4096)	строка	Адрес компоненты, который формируется подсистемой Alarm Processor для каждого абонентского устройства при регистрации/авторизации на МКД. CA = MKD.User.N, где N - число, присваиваемое каждому абонентскому устройству.
MKD-User-RegStat (600.7.1.1.990)	число	Переменная статуса регистрации/авторизации абонентских устройств на МКД. Значения: 0 — Не зарегистрирован; 1 — Зарегистрирован; 2 — Регистрация не требуется.

Таблица 11 — PROTEI.MKD.MCU-Table

Переменная	Тип	Описание
mcu-CA (600.3.1.1.4096)	строка	Компонентный адрес модуля MCU, подключенного к данному CPS. Значение – MKD.Mod_MCU.MCU.*, где * – номер MCU.
MKD-Mod_MCU-MCU-OSTATE (600.3.1.1.4097)	число	Переменная показывает совместимость версии МКД и MCU. Значения: 1 — Норма; 2 — Версии несовместимы.
mcu-Connection	строка	Траповая переменная. Показывает состояние соединения модуля MCU с модулем CPS. Значения: Active — Соединение активно; failed — Соединение не установлено; Unknown — Данное значение возможно в начале загрузки ПО, когда переменная еще не инициализирована. Дождитесь окончания загрузки ПО. (см. трапы 2001, 2002, 2003 в подразделе «Трапы»).
mcu-IP_addr (600.3.1.1.6000)	строка	Информационная переменная. IP-адрес модуля MCU.
mcu-Port (600.3.1.1.6001)	число	Информационная переменная. Показывает порт, используемый для соединения с MCU.
mcu-Users-Resrvd (600.3.1.1.6003.1)	число	Информационная переменная. Показывает количество зарезервированных ресурсов (портов для RTP-каналов и т.д.). Для неактивных соединений с MCU переменная будет иметь значение 0.
mcu-Users-Used (600.3.1.1.6003.2)	число	Информационная переменная. Показывает количество используемых ресурсов (портов для RTP-каналов и т.д.). Для неактивных соединений с MCU переменная будет иметь значение 0.
mcu-Alarm-Version (600.3.1.1.7000.1)	строка	Траповая переменная. Показывает совместимость версий модулей CPS и MCU. Значения: compatible — Версии совместимы not_compatible — Версии не совместимы. Если версия MCU устаревшая/несовместима с модулем CPS, то данный модуль MCU не будет использоваться CPS. Проверка выполняется при значении параметра StrictVersionCheck = 1 (параметр содержится в

Переменная	Тип	Описание
		конфигурационном файле /usr/protei/MKD/MCU/config/config.cfg). Обновите версию MCU. (см. трапы 2004, 2005 в подразделе «Трапы»).
mcu-Version (600.3.1.1.6002)	строка	Информационная переменная. Показывает версию MCU. Для неактивных соединений с модулем MCU переменная имеет значение - «0.0.0.0».

Таблица 12 — Переменные RD_NAS

Переменная	Тип	Описание переменной
RD_NAS-CA (600.4.1.1.4096)	строка	Компонентный адрес модуля интерфейса с RADIUS.
RD_NAS-OSTATE (600.4.1.1.4097)	строка	Траповая переменная Оперативное состояние соединения SSW5 с RADIUS. Возможные значения: Active — Соединение активно Failed — Соединение не установлено. Вызовы требующие авторизации на RADIUS-сервере перестанут проходить. Необходимо проверить, что случилось с RADIUS сервером или RD_NAS. Unknown — Данное значение возможно в начале загрузки ПО, когда переменная еще не инициализирована. Необходимо дождаться окончания загрузки ПО. (см. трапы 3001, 3002, 3003 в подразделе «Трапы»).

Таблица 13 — Переменные CDR

Переменная	Тип	Описание переменной
MKD-Mod_CDR-CA (600.5.1.1.4097)	строка	Адрес компоненты MKD-Mod_CDR. Траповая переменная, выбрасывается при некорректной отработке скрипта преобразования номеров CgPN, CdPN перед записью в CDR. (см. трап 6101).
MKD-Mod_CDR-PBX_ID (600.5.1.1.6000)	число	Идентификатор PBX'a, для которого вызывался скрипт.

5.2 Трапы

В таблице 14 ниже приведено описание SNMP-трапов SSW5.

Таблица 14 — SNMP-трапы SSW5

Номер и название трапа	Текстовое сообщение (английский)	Описание
1001 trapSystem-OSTATE-Active	MKD is active	Приоритет — Нормальный (normal) Действие при получении — запись в журнал Трап посылается при загрузке ПО SSW5. Причины перезапуска МКД смотреть по журналам info.log, trace.log, warning.log.
1002 trapSystem-OSTATE-Failed	MKD is failed	Приоритет — Критичный (critical) Действие при получении — запись в журнал, выдача на экран аварийного сообщения, подача звукового сигнала Трап посылается при критическом сбое в ПО SSW5. Необходимо просмотреть причины сбоя по журналам info.log, trace.log, warning.log и обратиться в службу технической поддержки, приложив все имеющиеся журналы ПРОТЕЙ-МКД
1003 trapSystem-OSTATE-Unknown	Operative state of MKD is unknown	Приоритет — Информационный (info) Действие при получении — запись в журнал Трап может прийти во время загрузки ПО SSW5. Дождаться полной загрузки ПО ПРОТЕЙ-МКД и прихода трапа №1001 или №1002
1101 trapSystem-Calls-CurrentValue	Current calls number processing by MKD - * («*» - текущее количество вызовов, обслуживаемых CPS).	Приоритет — Информационный (info) Действие при получении — запись в журнал Трап содержит количество вызовов, обслуживаемых SSW5 в текущий момент.
2001 trapMCU-Connection-Active	Connection between MKD and module MCU with address MKD.Mod_MCU.MCU. * is active («*» - номер модуля MCU).	Приоритет — Информационный (info) Действие при получении — запись в журнал Трап посылается при нормальном установлении соединения с модулем MCU.

Номер и название трапа	Текстовое сообщение (английский)	Описание
2002 trapMCU-Connection-Failed	Connection between MKD and module MCU with address MKD.Mod_MCU.MCU.* is failed («*» - номер модуля MCU).	Приоритет — Важный (major) Действие при получении — запись в журнал, выдача на экран аварийного сообщения, подача звукового сигнала Трап посылается при потере соединения с модулем MCU. Проверить совместимость версий МКД (модуля CPS) и модуля MCU с помощью переменной mcu-AlarmVersion и доступность модуля CPS с модуля MCU (например, командой ping).
2003 trapMCU-Connection-Unknown	State of connection between MKD and module MCU with address MKD.Mod_MCU.MCU.* is unknown («*» - номер модуля MCU).	Приоритет — Информационный (info) Действие при получении — запись в журнал Происходит загрузка ПО оборудования и переменная еще не инициализирована. Дождитесь окончания загрузки ПО.
2004 trapMCU-AlarmVersion-NotCompatible	Version of module MCU with address MKD.Mod_MCU.MCU.* is not compatible with version of MKD («*» - номер модуля MCU).	Приоритет — Важный (major) Действие при получении — запись в журнал, выдача на экран аварийного сообщения, подача звукового сигнала Проверка версии MCU завершилась неуспешно (несовместимая версия модуля MCU). Замените версию модуля MCU на более новую, выполнить рестарт ПО.
2005 trapMCU-AlarmVersion-Compatible	Version of module MCU with address MKD.Mod_MCU.MCU.* is compatible with version of MKD («*» - номер модуля MCU).	Приоритет — Нормальный (normal) Действие при получении — запись в журнал Проверка версии MCU завершилась успешно.
3002 trapRD-Connection-Failed	Connection between MKD and RD with address $2 is failed	Приоритет — Важный (major) Действие при получении — запись в журнал, выдача на экран аварийного сообщения, подача звукового сигнала Трап посылается при потере соединения с RADIUS. Проверить что случилось с RADIUS сервером или RD_NAS. Вызовы требующие авторизации на

Номер и название трапа	Текстовое сообщение (английский)	Описание
		RADIUS-сервере перестанут проходить.
3001 trapRD-Connection-Active	Connection between MKD and RD with address $2 is active	Приоритет — Нормальный (normal) Действие при получении — запись в журнал Трап посылается при нормальном установлении соединения с RADIUS.
3003 trapRD-Connection-Unknown	State of connection between MKD and RD with address $2 is unknown	Приоритет — Информационный (info) Действие при получении — запись в журнал Происходит загрузка ПО оборудования и переменная еще не инициализирована. Дождитесь окончания загрузки ПО.
4101 trapPer-Statistic	Statistic from mkd address $1	Приоритет — Нормальный (normal) Действие при получении — запись в журнал Трапы по статистическим параметрам МКД - Call.Out.Count, Call.Out.Answer, Call.Out.Busy, Call.Out.NoAns, Call.Out.Rel, Call.Out.Max, Call.Out.Min,Ratio.Err, NoCgPN, Call.In.Count, Call.In.Fail,Call.Input.Err,Call.Step1.Err,Call.Step3.Err, Call.Step4.Err.
5101 trapRTCP	Tcp: very much package was lost. FracLost - $2, User - $3, Dir - $4, Addr - $	Приоритет — Важный (major) Действие при получении — запись в журнал Трап посылается при превышении порога потерь RTP-пакетов у абонента на прием.
6101 trapCDR-Error-Script	Script for cdr' was finished with error, PBX - $2	Приоритет — Важный (major) Действие при получении — запись в журнал, выдача на экран аварийного сообщения, подача звукового сигнала Трап посылается при некорректной отработке скрипта преобразования номеров CgPN, CdPN перед записью в CDR.

Пример. Фрагмент mib-файла с трапами МКД.

```
trapSystem-OSTATE-Failed TRAP-TYPE
 ENTERPRISE mkd
 VARIABLES { sys-CA, sys-OSTATE }
 --&ACTIONS { log, critical, alarm, sound }
 --&CLEARS { 1001, 1003 }
 --&MESG "mkd is failed"
 DESCRIPTION "mkd failed"
 ::= 1002

trapSystem-OSTATE-Active TRAP-TYPE
 ENTERPRISE mkd
 VARIABLES { sys-CA, sys-OSTATE }
 --&ACTIONS { log, normal }
```

```
        --&CLEARS { 1002, 1003 }
        --&MESG "mkd is active"
        DESCRIPTION "mkd active"
        ::= 1001

       trapSystem-OSTATE-Unknown TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { sys-CA, sys-OSTATE }
        --&ACTIONS { log, info }
        --&CLEARS { 1001, 1002 }
        --&MESG "Operative state of mkd is unknown"
        DESCRIPTION "Operative state of mkd is unknown"
        ::= 1003

       trapSystem-Calls-CurrentValue TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { sys-CA, sys-Calls }
        --&ACTIONS { log, info }
        --&MESG "Current calls number processing by mkd - $2"
        DESCRIPTION "mkd calls number"
        ::= 1101


       trapMCU-Connection-Failed TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { mcu-CA, mcu-Connection }
        --&ACTIONS { log, major, alarm, sound }
        --&CLEARS { 2001, 2003 }
        --&MESG "Connection between mkd and module mcu with address $1 is failed"
        DESCRIPTION "mcu failed"
        ::= 2002

       trapMCU-Connection-Active TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { mcu-CA, mcu-Connection }
        --&ACTIONS { log, normal }
        --&CLEARS { 2002, 2003 }
        --&MESG "Connection between mkd and module mcu with address $1 is active"
        DESCRIPTION "mcu active"
        ::= 2001

       trapMCU-Connection-Unknown TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { mcu-CA, mcu-Connection }
        --&ACTIONS { log, info }
        --&CLEARS { 2001, 2002 }
        --&MESG "State of connection between mkd and module mcu with address $1 is
unknown"
        DESCRIPTION "Operative state of mcu is unknown"
        ::= 2003

       trapMCU-AlarmVersion-Compatible TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { mcu-CA, mcu-AlarmVersion }
        --&ACTIONS { log, normal }
        --&CLEARS { 2004 }
```

```
        --&MESG "Version of module mcu with address $1 is compatible with version of mkd"
        DESCRIPTION "Version of mcu is compatible with version of mkd"
        ::= 2005

       trapMCU-AlarmVersion-NotCompatible TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { mcu-CA, mcu-AlarmVersion }
        --&ACTIONS { log, major, alarm, sound }
        --&CLEARS { 2005 }
        --&MESG "Version of module mcu with address $1 is not compatible with version of
mkd"
        DESCRIPTION "Version of mcu is not compatible with version of mkd"
        ::= 2004


       trapRD-Connection-Failed TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { rd-CA, rd-Connection }
        --&ACTIONS { log, major, alarm, sound }
        --&CLEARS { 3001, 3003 }
        --&MESG "Connection between mkd and rd with address $1 is failed"
        DESCRIPTION "rd failed"
        ::= 3002

       trapRD-Connection-Active TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { rd-CA, rd-Connection }
        --&ACTIONS { log, normal }
        --&CLEARS { 3002, 3003 }
        --&MESG "Connection between mkd and rd with address $1 is active"
        DESCRIPTION "rd active"
        ::= 3001

       trapRD-Connection-Unknown TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { rd-CA, rd-Connection }
        --&ACTIONS { log, info }
        --&CLEARS { 3001, 3002 }
        --&MESG "State of connection between mkd and rd with address $1 is unknown"
        DESCRIPTION "Operative state of rd is unknown"
        ::= 3003


       trapPer-Statistic TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { per-CA, per-NoCgPN, per-OutCoun, per-OutAnsw, per-OutBusy, per-
OutNoAns, per-OutRel, per-OutMax, per-OutMin, per-Ratio, per-InCount, per-InFail, per-
InputErr, per-st1, per-st3, per-st4}
        --&ACTIONS { log, normal }
        --&MESG "Statistic from mkd address $1"
        DESCRIPTION "Receive statistic from mkd"
        ::= 4101


       trapRTCP-veryMuchLost TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { rtcp-CA, rtcp-FRLOST, rtcp-USER, rtcp-DIRECT, rtcp-ADDR }
```

```
        --&ACTIONS { log, major}
        --&MESG "rtcp: very much package was lost. FracLost - $2, User - $3, Dir - $4, Addr -
$5"
        DESCRIPTION "rtcp: very much package was lost"
        ::= 5101



       trapCDR-Error-Script TRAP-TYPE
        ENTERPRISE mkd
        VARIABLES { cdr-CA, cdr-PBX-ID }
        --&ACTIONS { log, major, alarm, sound }
        --&MESG "Script for cdr's was finished with error, PBX - $2"
        DESCRIPTION "cdr-script failed"
        ::= 6101
```

6 Приложение

6.1 Пример реального файла конфигурации ap.cfg.

```
[General]
ApplicationAddress=MKD.60;
MaxConnectionCount=10;
ManagerThread=1;
CyclicTreeWalk=1;

[Dynamic]
# Format {caVar;strValue;};

[AtePath2ObjName]
# Format {ctObject;caVar;};

# System variables of MKD
{ MKD(600).Sys(2,1,1); CA(4096); };
{ MKD(600).Sys(2,1,1); OSTATE(4097); };
{ MKD(600).Sys(2,1,1); Calls(5000); };

{ MKD(600).Sys(2).Period(3,1,1); CA(4096); };
{ MKD(600).Sys(2).Period(3,1,1); NoCgPN(6000);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Count(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Answer(2);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Busy(3);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).NoAns(4);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Rel(5);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Max(6);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6001).Out(1).Min(7);};
{ MKD(600).Sys(2).Period(3,1,1); Ratio(6002).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).In(1).Count(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).In(1).Fail(2);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Input(2).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Step1(3).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Step3(4).Err(1);};
{ MKD(600).Sys(2).Period(3,1,1); Call(6003).Step4(5).Err(1);};

{ MKD(600).Sys(2).RTCP(2,1,1); CA(4096); };
{ MKD(600).Sys(2).RTCP(2,1,1); USER(6000);};
{ MKD(600).Sys(2).RTCP(2,1,1); DIRECT(6001);};
{ MKD(600).Sys(2).RTCP(2,1,1); ADDR(6002);};
{ MKD(600).Sys(2).RTCP(2,1,1); FRLOST(6003);};


{ MKD(600).User(7,1,1);CA(4096); };
{ MKD(600).User(7,1,1);RegStat(99); };

# Variables of MCU
{ MKD(600).Mod_MCU(3).MCU(1,1);CA(4096); };
{ MKD(600).Mod_MCU(3).MCU(1,1);OSTATE(4097); };
{ MKD(600).Mod_MCU(3).MCU(1,1);IP_addr(6000); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Port(6001); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Version(6002); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Users(6003).Resrvd(1); };
{ MKD(600).Mod_MCU(3).MCU(1,1);Users(6003).Used(2); };
```

```
        { MKD(600).Mod_MCU(3).MCU(1,1);Alarm(7000).Version(1); };

        # variables of RD_NAS
        { MKD(600).Mod_RD(4).RD(1,1);CA(4096); };
        { MKD(600).Mod_RD(4).RD(1,1);OSTATE(4097); };

        # variables of CDR
        { MKD(600).Mod_CDR(5,1,1);CA(4096); };
        { MKD(600).Mod_CDR(5,1,1);PBX_ID(6000); };



        [SNMP]
        ListenIP = 0.0.0.0;
        ListenPort = 1161;
        OwnEnterprise = 1.3.6.1.4.1.20873.600;

        [StandardMib]
        # Format {request;answer_type;answer;};

        # SysDescr
        {1.3.6.1.2.1.1.1.0;STRING;"MKD";};
        # SysObjectID
        {1.3.6.1.2.1.1.2.0;OBJECT_ID;1.3.6.1.4.1.20873;};

        [AtePath2Oid]                   # Not used
        # Format {ctObject;caVar;ObjectID;};

        [SNMPTrap]
        # Format {SNMP_ManagerIP;SNMP_ManagerPort;caObjectFilter;ctObjectFilter;caVarFil
ter;};
        #{"192.168.108.111";162;};
        FirstVarOwn = 0;

        [Filter]
        # filters send to AP_Agent
        CA_Object=".*";
        CT_Object=".*";
        CA_Var=".*";
        TrapIndicator=-1;
        DynamicIndicator=-1;

        [Logs]
        TreeTimerPeriod=300000;

        [Test]
        Freq=1700000;

        [SpecificTrapCA_Var]
        #Sys
        {"Calls"; 101;};

        #Period
        {"NoCgPN"; 101;};
        #{"Call.Out.Count"; 102;};
        #{"Call.Out.Answer"; 103;};
```

```
#{"Call.Out.Busy"; 104;};
#{"Call.Out.NoAns"; 105;};
#{"Call.Out.Rel"; 106;};
#{"Call.Out.Max"; 107;};
#{"Call.Out.Min"; 108;};
#{"Ratio.Err"; 109;};
#{"Call.In.Count"; 110;};
#{"Call.In.Fail"; 111;};
#{"Call.Input.Err"; 112;};
#{"Call.Step1.Err"; 113;};
#{"Call.Step3.Err"; 114;};
#{"Call.Step4.Err"; 115;};

#RTCP
{"USER"; 101;};
#{"DIRECT"; 102;};
#{"ADDR; 103;};
#{"FRLOST"; 104;};

#CDR
{"PBX_ID"; 101;};

[SpecificTrapCA_Object]

[SpecificTrapCT_Object]
{"MKD.Sys"; 1;};
{"MKD.Mod_MCU.MCU"; 2;};
{"MKD.Mod_RD.RD"; 3;};
{"MKD.Sys.Period"; 4;};
{"MKD.Sys.RTCP"; 5;};
{"MKD.Mod_CDR"; 6;};
```