# OSS Fault Analytics and Statistics

Customization Guide

Version 1.2.0

Edition 1.0 - December 2016

Hewlett Packard
Enterprise

# Notices

### Legal notice

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

### Trademarks

# Contents

# List of tables

# List of figures

# Preface

## About this guide

This guide describes how to integrate and customize the HPE OSS Fault Analytics and Statistics product.

Product name: HPE OSS Fault Analytics and Statistics
Product version: 1.2.0

## Audience

This customization guide is for anyone who is responsible for integrating or customizing a HPE OSS Fault Analytics and Statistics solution.
The readers are assumed to understand Linux shell concepts, database concepts and xml, json syntax.

## Software versions

**The terms Unix and Linux are used as a generic reference to the operating system, unless otherwise specified. The software versions referred to in this document are as follows:**

**Table 1: Software versions**

| Product version | Supported operating systems |
|---|---|
| HPE OSS Analytics Foundation version 1.1.4 | Red Hat Enterprise Linux Server release 6.8 |
| HPE OSS Fault Analytics and Statistics version 1.2 | Red Hat Enterprise Linux Server release 6.8 |
| HPE Vertica version 7.2.3 | Red Hat Enterprise Linux Server release 6.8 |
| HPE UMB Server version 1.1 | Red Hat Enterprise Linux Server release 6.8 |
| HPE Unified OSS Console 2.3 | Red Hat Enterprise Linux Server release 6.8 |
| HPE TeMIP 6.2 | Red Hat Enterprise Linux Server release 6.8 |

## Typographical Conventions

`Courier` Font:

- Source code and examples of file contents.
- Commands that you enter on the screen.
- Pathnames
- Keyboard key names

*Italic* Text:

- Filenames, programs and parameters.
- The names of other documents referenced in this manual.

Bold Text:

- To introduce new terms and to emphasize important words.

# Associated Documents

The following documents contain useful reference information:

*HPE OSS Analytics Foundation Release Notes*
*HPE OSS Analytics Foundation Integration Guide*
*HPE OSS Fault Analytics and Statistics Release Notes*
*HPE OSS Fault Analytics and Statistics Install and Admin Guide*
*HPE OSS Fault Analytics and Statistics User Guide*

# Support

Please visit our HPE Software Support Online Web site at https://softwaresupport.hpe.com for contact information, and details about HPE Software products, services, and support.

The Software support area of the web site includes the following:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

# Chapter 1 Product overview

## 1.1 Introduction

Please refer to the *HPE OSS Fault Analytics and Statistics Install and Admin Guide* for an introduction to OSS Fault Analytics and Statistics.

## 1.2 Architecture

Please refer to the *HPE OSS Fault Analytics and Statistics Install and Admin Guide* for a description of the architecture of OSS Fault Analytics and Statistics.

# Chapter 2
# Datamart

Before explaining how to customize the HPE OSS Fault Analytics & Statistics solution, we must introduce with some concepts. This document starts with **Datamart**.

A Datamart is a database containing historical and aggregated data stored in a multidimensional schema, generally implemented as a ‘star’ schema, where **facts** are ‘surrounded’ by / linked to **dimensions** tables. Thus, this database is optimized for reporting and analysis; the facts can be analyzed from different axis, the dimensions.

## 2.1 FAS datamart

Two types of facts are stored in the FAS datamart: the alarms and the raw events.

In the following you will find the description of the Facts and Dimensions related to alarms and Facts and Dimensions related to raw events.

### 2.1.1 Alarms related tables

The fact table related to the alarms is the *FCT_FAULT* table.
The alarms coming from TeMIP via UMB are stored over time by FAS within this fact table. FCT_FAULT table will be continually updated overtime, adding new row for each new alarm, or updating an existing row for each updated alarm.

There are 11 dimension tables associated with this fact table. They represent the different types of analysis that can be done on the alarms. Obviously, the dimension tables are also continuously updated overtime depending on the alarm received.

See the diagram below showing the ‘star’: with FCT_FAULT table in the middle, ‘surrounded’ by / linked to the dimension tables.

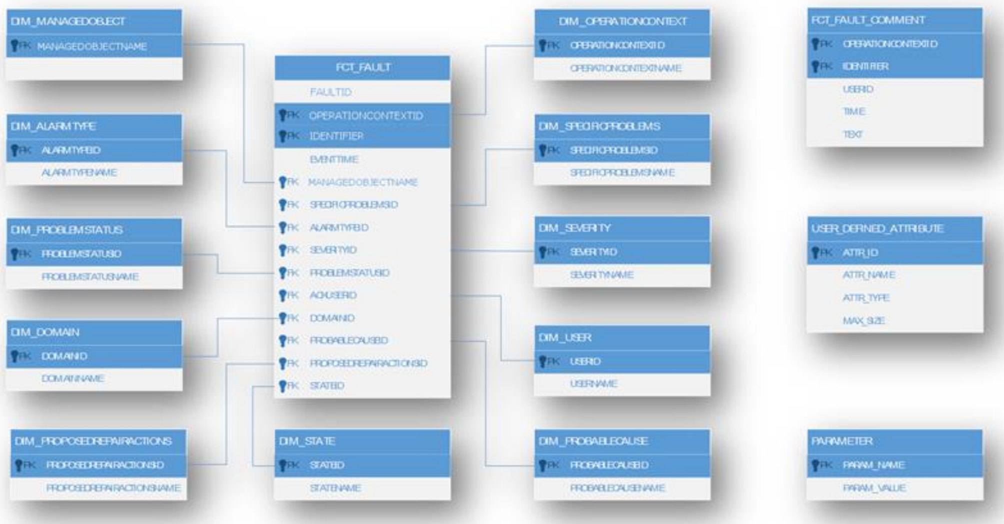


**Figure 1: FAS Datamart: alarms related tables**

### 2.1.1.1 Dimension tables

Here are described the FAS datamart dimensions tables related to alarms:

DIM_ALARMTYPE table

| Column Name | Column Type |
|---|---|
| ALARMTYPEID | INTEGER |
| ALARMTYPENAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_DOMAIN table

| Column Name | Column Type |
|---|---|
| DOMAINID | INTEGER |
| DOMAINNAME | VARCHAR(700) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_MANAGEDOBJECT table

| Column Name | Column Type |
|---|---|
| MANAGEDOBJECTNAME | VARCHAR(700) |
| CLASSNAME | VARCHAR(700) |
| GLOBALMONAME | VARCHAR(700) |
| GLOBALCLASSNAME | VARCHAR(700) |
| TNSNAME | VARCHAR(700) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_OPERATIONCONTEXT table

| Column Name | Column Type |
|---|---|
| OPERATIONCONTEXTID | INTEGER |
| OPERATIONCONTEXTNAME | VARCHAR(700) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_PROBABLECAUSE table

| Column Name | Column Type |
|---|---|
| PROBABLECAUSEID | INTEGER |
| PROBABLECAUSENAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_PROBLEMSTATUS table

| Column Name | Column Type |
|---|---|
| PROBLEMSTATUSID | INTEGER |
| PROBLEMSTATUSNAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_PROPOSEDREPAIRACTIONS table

| Column Name | Column Type |
|---|---|

| | |
|---|---|
| PROPOSEDREPAIRACTIONID | INTEGER |
| PROPOSEDREPAIRACTIONS | VARCHAR(65000) |
| PROPOSEDREPAIRACTIONSNAME | VARCHAR(500) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_SEVERITY table

| Column Name | Column Type |
|---|---|
| SEVERITYID | INTEGER |
| SEVERITYNAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_SPECIFICPROBLEMS table

| Column Name | Column Type |
|---|---|
| SPECIFICPROBLEMSID | INTEGER |
| SPECIFICPROBLEMS | VARCHAR(65000) |
| SPECIFICPROBLEMSNAME | VARCHAR(500) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_STATE table

| Column Name | Column Type |
|---|---|
| STATEID | INTEGER |
| STATENAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_USER table

| Column Name | Column Type |
|---|---|
| USERID | INTEGER |
| USERNAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

### 2.1.1.2 FCT_FAULT table

Here is the description of the FCT_FAULT table:

| Column Name | Column Type | Description |
|---|---|---|
| ALARMTYPEID | INTEGER | Alarm Type Identifier (Foreign Key) |
| DOMAINID | INTEGER | Domain Identifier (Foreign Key) |
| OPERATIONCONTEXTID | INTEGER | Operation Context Identifier (Foreign Key) |
| IDENTIFIER | INTEGER | Uniquely identifies each Alarm Object belonging to a particular Operation Context |
| MANAGEDOBJECTNAME | VARCHAR | Managed Object Name (Foreign Key) |
| PROBABLECAUSEID | INTEGER | Probable Cause Identifier  (Foreign Key) |
| SPECIFICPROBLEMSID | INTEGER | Specific Problems Identifier (Foreign Key) |
| PROPOSEDREPAIRACTIONSID | INTEGER | Proposed Repair Actions Identifier (Foreign Key) |
| SEVERITYID | INTEGER | Severity Identifier (Foreign Key) |
| STATEID | INTEGER | State Identifier (Foreign Key) |

| Column Name | Column Type | Description |
|---|---|---|
| PROBLEMSTATUSID | INTEGER | Problem Status Identifier (Foreign Key) |
| ACKUSERID | INTEGER | Identifier of the user who acknowledged the alarm (Foreign Key) |
| CLOSEUSERID | INTEGER | Identifier of the user who closed the alarm (Foreign Key) |
| HANDLEUSERID | INTEGER | Identifier of the user who handled the alarm (Foreign Key) |
| RELEASEUSERID | INTEGER | Identifier of the user who released the alarm (Foreign Key) |
| TERMUSERID | INTEGER | Identifier of the user who terminated the alarm (Foreign Key) |
| ORIGINALEVENTTIME | TIMESTAMP | Time at which the original Alarm Object was generated |
| EVENTTIME | TIMESTAMP | If there is no associated similar alarm, it is the time at which the alarm occurred.<br>If there are associated similar alarms, it is the time of the most recent similar alarm. |
| CREATIONTIMESTAMP | TIMESTAMP | Time of the creation of the alarm in TeMIP |
| ALARMORIGIN | INTEGER | Defines the origin of this Alarm Object as:<br>an incoming alarm – *IncomingAlarm*<br>or the result of a directive - *Create*, *Demote* or *Promote* Directive. |
| ESCALATEDALARMFLAG | BOOLEAN | Indicates if this alarm has been escalated or not |
| CLEARFLAG | BOOLEAN | Indicates if this alarm has been cleared or not |
| OUTAGEFLAG | BOOLEAN | Specifies if the alarm is from out of service equipment |
| ACKTIMESTAMP | TIMESTAMP | Time of acknowledgement of the alarm |
| CLEARTIMESTAMP | TIMESTAMP | Time of clearance of the alarm |
| CLOSETIMESTAMP | TIMESTAMP | Time of closure of the alarm |
| HANDLETIMESTAMP | TIMESTAMP | Time at which the alarm was handled |
| RELEASETIMESTAMP | TIMESTAMP | The date of release of the Alarm |
| TERMTIMESTAMP | TIMESTAMP | The date of termination of the Alarm |
| NOTIFICATIONIDENTIFIER | INTEGER | Provides unique identification for all alarms concerning a particular Managed Object. The identifier can be used for correlation purposes |
| ADDITIONALTEXT | VARCHAR(65000) | A free form text description of the problem reported |
| ALARMCLASS | INTEGER | Indicates if it is an original alarm (value = 0) or a similar alarm (value = 1) |
| ALARMOBJECTOPERATORNOTE | VARCHAR(128) | A free form text that allows the operator to write comments concerning the alarm |
| BACKEDUPSTATUS | BOOLEAN | A parameter that specifies whether the Managed Object is backed up by another entity or not |
| BACKUPOBJECT | VARCHAR(200) | Specifies the entity that backs up the Managed Object concerned by the alarm |
| ORIGINALSEVERITY | VARCHAR(13) | Defines the original severity of an Alarm Object |
| PARENTALARMOBJECT | INTEGER | The name of the parent Alarm Object of this Alarm Object |
| RULENAME | VARCHAR(100) | Specifies the Alarm FM name for the rule that caused the alarm to fire |
| SECURITYALARMDETECTOR | VARCHAR(300) | Defines the Object Detector of a security alarm |
| TRENDINDICATION | VARCHAR(10) | Indicates whether a problem is becoming more or less severe |

| Column Name | Column Type | Description |
|---|---|---|
| USERIDENTIFIER | INTEGER | A free form identifier defining (and defined by) the user that created the Alarm Object |
| USERTEXT | VARCHAR(128) | A free form text string defined by the user that created the Alarm Object |
| AGENTALARMIDENTIFIER | VARCHAR(65000) | Agent Alarm Identifier is used with the Agent Entity attribute to uniquely identify a 3GPP alarm in the scope of a TeMIP Platform |
| AGENTENTITY | VARCHAR(65000) | Agent Entity is used with the Agent Alarm Identifier attribute to uniquely identify a 3GPP alarm in the scope of a TeMIP Platform |
| ACKFLAG | BOOLEAN | Indicates if this alarm has been acknowledged or not |
| CLOSEFLAG | BOOLEAN | Indicates if this alarm has been closed or not |
| HANDLEFLAG | BOOLEAN | Indicates if this alarm has been handled or not |
| TERMFLAG | BOOLEAN | Indicates if this alarm has been terminated or not |
| PROBLEMOCCURRENCES | INTEGER | The total number of alarm occurrences received for the same problem |
| CLEAROCCURRENCES | INTEGER | Number of cleared similar alarms |
| CRITICALOCCURRENCES | INTEGER | Number of critical similar alarms |
| INDETERMINATEOCCURRENCES | INTEGER | Number of indeterminate similar alarms |
| MAJOROCCURRENCES | INTEGER | Number of major similar alarms |
| MINOROCCURRENCES | INTEGER | Number of minor similar alarms |
| WARNINGOCCURRENCES | INTEGER | Number of minor warning alarms |
| SATOTAL | INTEGER | Total number of similar alarms for this Alarm Object |
| ACKDURATION | INTEGER | Time needed to acknowledge the alarm |
| CLEARDURATION | INTEGER | Time needed to clear the alarm |
| CLOSEDURATION | INTEGER | Time needed to close the alarm |
| HANDLEDURATION | INTEGER | Time needed to handle the alarm |
| TERMDURATION | INTEGER | Time needed to terminate the alarm |
| COLLECTIONDURATION | INTEGER | Time needed to collect the alarm |
| INFAULTDURATION | INTEGER | Difference between the close (or clear) timestamp and the original event time |
| INMGMTDURATION | INTEGER | Difference between the close (or clear) timestamp and the creation timestamp |
| HANDLEDBY | VARCHAR(65000) | Specifies the list of Trouble Tickets that currently handle this alarm |
| CLOSEDBY | VARCHAR(65000) | Specifies the list of Trouble Tickets for which a CLOSE directive has been received for this alarm |
| PARENTS | VARCHAR(65000) | Specifies the Parent Alarm identifier string of the Group to which the Alarm belongs. Can have multiple Parent Alarms identifier separated by “,”. |
| CHILDREN | VARCHAR(65000) | Specifies the Children Alarm identifier string of the Group to which this Alarm belongs. Can have multiple Children Alarms Identifier separated by “””. |
| LASTMODIFICATIONTIMESTAMP | TIMESTAMP | Latest time at which the Alarm Object was modified |
| UPDATE_TIMESTAMP | TIMESTAMP | Time at which this record has been persisted |

The FCT_FAULT primary key is: *OPERATIONCONTEXTID, IDENTIFIER*.

### 2.1.1.3 Summarization tables

As part of the Fault Analytics and Statistics product the '**Summarization**' processing regularly fills other facts tables containing summarized data based on the FCT_FAULT data.
The aggregations are done on different dimensions and on different time granularities (hourly, daily, weekly, monthly).

FAS comes with 11 default summarizations for alarms divided on 3 categories:

| Category of summarizations | Time aggregation | Usage of summarized data |
|---|---|---|
| Management | hourly, daily, weekly, monthly | Analysis of the **network management activity**<br>Multi dimensional analysis of alarm indicators per *Operation Contexts/Users* |
| Network | hourly, daily, weekly, monthly | Analysis of the **global activity of the network in term of fault**<br>Multi dimensional analysis of alarm indicators per *Operation Contexts/Domains/Managed Objects* |
| Equipment | daily, weekly, monthly | Analysis of the **faults in term of alarm type**<br>Multi dimensional analysis of alarm indicators per *Managed Objects/Alarmtype/ SpecificProblem/ProblemStatus/State/Severity* |

11 default summarization tables are thus populated by FAS:

```
SUMM_HOURLY_MANAGEMENT,
SUMM_DAILY_MANAGEMENT,
SUMM_WEEKLY_MANAGEMENT,
SUMM_MONTHLY_MANAGEMENT,

SUMM_HOURLY_NETWORK,
SUMM_DAILY_NETWORK,
SUMM_WEEKLY_NETWORK,
SUMM_MONTHLY_NETWORK,

SUMM_DAILY_EQUIPMENT,
SUMM_WEEKLY_EQUIPMENT,
SUMM_MONTHLY_EQUIPMENT.
```

Those summarized tables are also facts tables and thus are also associated to dimension tables (same ones than for `FCT_FAULT`).
The benefit of those tables is that they contains pre-calculated aggregated fact values which can be requested directly for statistical analysis.
For optimization purpose, FAS reports rely on those tables.

Here are the descriptions of three of those summarized tables: `SUMM_HOURLY_MANAGEMENT`, `SUMM_HOURLY_NETWORK` and `SUMM_DAILY_EQUIPMENT`.

Note that the descriptions of the other tables for the other time granularities in the same categories are equivalent to the ones below.

Here is the description of the **SUMM_HOURLY_MANAGEMENT** table:

| Column Name | Column Type | Description |
|---|---|---|
| TIME | TIMESTAMP | Timestamp of the Hour for which the aggregation occurred |
| OPERATIONCONTEXTID | INTEGER | Operation Context Identifier (Foreign Key) for the aggregated alarms during this hour |

| Column Name | Column Type | Description |
|---|---|---|
| PROBLEMSTATUSID | INTEGER | Problem Status Identifier (Foreign Key) for the aggregated alarms during this hour |
| ACKUSERID | INTEGER | Identifier (Foreign Key) of the user who acknowledged the alarms aggregated during this hour |
| CLOSEUSERID | INTEGER | Identifier(Foreign Key) of the user who closed the alarms aggregated during this hour |
| HANDLEUSERID | INTEGER | Identifier (Foreign Key) of the user who handled the alarms aggregated during this hour |
| RELEASEUSERID | INTEGER | Identifier (Foreign Key) of the user who released the alarms aggregated during this hour |
| TERMUSERID | INTEGER | Identifier (Foreign Key) of the user who terminated the alarms aggregated during this hour |
| SUMMARIZED_COUNT | INTEGER | Number of FCT_FAULT rows aggregated for this hour |
| ALARM_OBJECTS_COUNTER_SUM | INTEGER | Number of alarm objects during this hour |
| SIMILAR_ALARM_COUNTER_SUM | INTEGER | Number of similar alarms during this hour |
| ALARM_COUNTER_SUM | INTEGER | Number of alarms (AO and similar) during this hour |
| CRITICAL_OCCURENCES_SUM | INTEGER | Number of critical occurrences during this hour |
| MAJOR_OCCURENCES_SUM | INTEGER | Number of major occurrences during this hour |
| MINOR_OCCURENCES_SUM | INTEGER | Number of minor occurrences during this hour |
| WARNING_OCCURENCES_SUM | INTEGER | Number of warning occurrences during this hour |
| INDETERMINATE_OCCURENCES_SUM | INTEGER | Number of indeterminate occurrences during this hour |
| CLEAR_OCCURENCES_SUM | INTEGER | Number of clear occurrences during this hour |
| CLEAR_COUNTER_SUM | INTEGER | Number of clearance during this hour |
| ESCALATED_COUNTER_SUM | INTEGER | Number of escalated alarms during this hour |
| HANDLED_COUNTER_SUM | INTEGER | Number of handled alarms during this hour |
| CLOSED_COUNTER_SUM | INTEGER | Number of closed alarms during this hour |
| ACK_COUNTER_SUM | INTEGER | Number of acknowledged alarms during this hour |
| TERMINATED_FLAG_SUM | INTEGER | Number of terminated alarms during this hour |
| ACK_BY_COUNT | INTEGER | Number of user who acknowledged alarms during this hour |
| CLOSED_BY_COUNT | INTEGER | Number of user who closed alarms during this hour |
| TERMINATED_BY_COUNT | INTEGER | Number of user who terminated alarms during this hour |
| RELEASED_BY_COUNT | INTEGER | Number of user who released alarms during this hour |
| HANDLED_BY_COUNT | INTEGER | Number of user who handled alarms during this hour |
| CLEAR_DURATION_SUM | INTEGER | Total clear duration for alarms of this hour |
| HANDLE_DURATION_SUM | INTEGER | Total handle duration for alarms of this hour |
| CLOSE_DURATION_SUM | INTEGER | Total close duration for alarms of this hour |
| ACK_DURATION_SUM | INTEGER | Total acknowledgment duration for alarms of this hour |
| TERM_DURATION_SUM | INTEGER | Total termination duration for alarms of this hour |
| COLLECTION_DURATION_SUM | INTEGER | Total collection duration for alarms of this hour |
| IN_FAULT_DURATION_SUM | INTEGER | Total in fault duration for alarms of this hour |
| IN_MANAGEMENT_DURATION_SUM | INTEGER | Total in management duration for alarms of this hour |
| COLLECTION_COUNTER_SUM | INTEGER | Total counter of not null collection duration for alarms of this hour |
| IN_FAULT_COUNTER_SUM | INTEGER | Total counter of in fault duration greater than 0 for alarms of this hour |
| IN_MANAGEMENT_COUNTER_SUM | INTEGER | Total counter of in management duration greater than 0 for alarms of this hour |
| CLEAR_DURATION_AVG | FLOAT | Average of clear duration for alarms of this hour |
| HANDLE_DURATION_AVG | FLOAT | Average of handle duration for alarms of this hour |
| CLOSE_DURATION_AVG | FLOAT | Average of close duration for alarms of this hour |

| Column Name | Column Type | Description |
|---|---|---|
| ACK_DURATION_AVG | FLOAT | Average of acknowledgment duration for alarms of this hour |
| TERM_DURATION_AVG | FLOAT | Average of termination duration for alarms of this hour |
| COLLECTION_DURATION_AVG | FLOAT | Average of collection duration for alarms of this hour |
| IN_FAULT_DURATION_AVG | FLOAT | Average of in fault duration for alarms of this hour |
| IN_MANAGEMENT_DURATION_AVG | FLOAT | Average of in management duration for alarms of this hour |
| UPDATE_TIMESTAMP | TIMESTAMP | Time at which this record has been persisted |

Here is the description of the SUMM_HOURLY_NETWORK table:

| Column Name | Column Type | Description |
|---|---|---|
| TIME | TIMESTAMP | Timestamp of the Hour for which the aggregation occurred |
| DOMAINID | INTEGER | Domain Identifier (Foreign Key) for the alarms aggregated during this hour |
| OUTAGEFLAG | BOOLEAN | Specifies if the alarms aggregated on this hour are from out of service equipment or not |
| OPERATIONCONTEXTID | INTEGER | Operation Context Identifier (Foreign Key) for the alarms aggregated during this hour |
| MANAGEDOBJECTNAME | VARCHAR | Managed Object Name (Foreign Key) for the alarms aggregated during this hour |
| SUMMARIZED_COUNT | INTEGER | Number of FCT_FAULT rows aggregated for this hour |
| ALARM_OBJECTS_COUNTER_SUM | INTEGER | Number of alarm objects during this hour |
| SIMILAR_ALARM_COUNTER_SUM | INTEGER | Number of similar alarms during this hour |
| ALARM_COUNTER_SUM | INTEGER | Number of alarms (AO and similar) during this hour |
| CRITICAL_OCCURENCES_SUM | INTEGER | Number of critical occurrences during this hour |
| MAJOR_OCCURENCES_SUM | INTEGER | Number of major occurrences during this hour |
| MINOR_OCCURENCES_SUM | INTEGER | Number of minor occurrences during this hour |
| WARNING_OCCURENCES_SUM | INTEGER | Number of warning occurrences during this hour |
| INDETERMINATE_OCCURENCES_SUM | INTEGER | Number of indeterminate occurrences during this hour |
| CLEAR_OCCURENCES_SUM | INTEGER | Number of clear occurrences during this hour |
| CLEAR_COUNTER_SUM | INTEGER | Number of clearance during this hour |
| ESCALATED_COUNTER_SUM | INTEGER | Number of escalated alarms during this hour |
| HANDLED_COUNTER_SUM | INTEGER | Number of handled alarms during this hour |
| CLOSED_COUNTER_SUM | INTEGER | Number of closed alarms during this hour |
| ACK_COUNTER_SUM | INTEGER | Number of acknowledged alarms during this hour |
| TERMINATED_FLAG_SUM | INTEGER | Number of terminated alarms during this hour |
| ACK_BY_COUNT | INTEGER | Number of user who acknowledged alarms during this hour |
| CLOSED_BY_COUNT | INTEGER | Number of user who closed alarms during this hour |
| TERMINATED_BY_COUNT | INTEGER | Number of user who terminated alarms during this hour |
| RELEASED_BY_COUNT | INTEGER | Number of user who released alarms during this hour |
| HANDLED_BY_COUNT | INTEGER | Number of user who handled alarms during this hour |
| CLEAR_DURATION_SUM | INTEGER | Total clear duration for alarms of this hour |
| HANDLE_DURATION_SUM | INTEGER | Total handle duration for alarms of this hour |
| CLOSE_DURATION_SUM | INTEGER | Total close duration for alarms of this hour |
| ACK_DURATION_SUM | INTEGER | Total acknowledgment duration for alarms of this hour |

| Column Name | Column Type | Description |
|---|---|---|
| TERM_DURATION_SUM | INTEGER | Total termination duration for alarms of this hour |
| COLLECTION_DURATION_SUM | INTEGER | Total collection duration for alarms of this hour |
| IN_FAULT_DURATION_SUM | INTEGER | Total in fault duration for alarms of this hour |
| IN_MANAGEMENT_DURATION_SUM | INTEGER | Total in management duration for alarms of this hour |
| COLLECTION_COUNTER_SUM | INTEGER | Total counter of collection duration greater than 0 for alarms of this hour |
| IN_FAULT_COUNTER_SUM | INTEGER | Total counter of in fault duration greater than 0 for alarms of this hour |
| IN_MANAGEMENT_COUNTER_SUM | INTEGER | Total counter of in management duration greater than 0 for alarms of this hour |
| CLEAR_DURATION_AVG | FLOAT | Average of clear duration for alarms of this hour |
| HANDLE_DURATION_AVG | FLOAT | Average of handle duration for alarms of this hour |
| CLOSE_DURATION_AVG | FLOAT | Average of close duration for alarms of this hour |
| ACK_DURATION_AVG | FLOAT | Average of acknowledgment duration for alarms of this hour |
| TERM_DURATION_AVG | FLOAT | Average of termination duration for alarms of this hour |
| COLLECTION_DURATION_AVG | FLOAT | Average of collection duration for alarms of this hour |
| IN_FAULT_DURATION_AVG | FLOAT | Average of in fault duration for alarms of this hour |
| IN_MANAGEMENT_DURATION_AVG | FLOAT | Average of in management duration for alarms of this hour |
| UPDATE_TIMESTAMP | TIMESTAMP | Time at which this record has been persisted |

Here is the description of the SUMM_DAILY_EQUIPMENT table:

| Column Name | Column Type | Description |
|---|---|---|
| TIME | TIMESTAMP | Timestamp of the Day for which the aggregation occurred |
| ALARMTYPEID | INTEGER | Alarm Type Identifier (Foreign Key) for the alarms aggregated for this day |
| SEVERITYID | INTEGER | Severity Identifier (Foreign Key) for the alarms aggregated for this day |
| OUTAGEFLAG | BOOLEAN | Specifies if the alarms aggregated on this day are from out of service equipment or not |
| STATEID | INTEGER | State Identifier (Foreign Key) for the alarms aggregated for this day |
| MANAGEDOBJECTNAME | VARCHAR | Managed Object Name (Foreign Key) for the alarms aggregated for this day |
| PROBABLECAUSEID | INTEGER | Probable Cause Identifier  (Foreign Key) for the alarms aggregated for this day |
| PROBLEMSTATUSID | INTEGER | Problem Status Identifier (Foreign Key) for the alarms aggregated for this day |
| SPECIFICPROBLEMSID | INTEGER | Specific Problems Identifier (Foreign Key) for the alarms aggregated for this day |
| SUMMARIZED_COUNT | INTEGER | Number of FCT_FAULT rows aggregated for this day |
| ALARM_OBJECTS_COUNTER_SUM | INTEGER | Number of alarm objects during this day |
| SIMILAR_ALARM_COUNTER_SUM | INTEGER | Number of similar alarms during this day |
| ALARM_COUNTER_SUM | INTEGER | Number of alarms (AO and similar) during this day |
| CRITICAL_OCCURENCES_SUM | INTEGER | Number of critical occurrences during this day |
| MAJOR_OCCURENCES_SUM | INTEGER | Number of major occurrences during this day |
| MINOR_OCCURENCES_SUM | INTEGER | Number of minor occurrences during this day |
| WARNING_OCCURENCES_SUM | INTEGER | Number of warning occurrences during this day |
| INDETERMINATE_OCCURENCES_SUM | INTEGER | Number of indeterminate occurrences during this day |

| Column Name | Column Type | Description |
|---|---|---|
| CLEAR_OCCURENCES_SUM | INTEGER | Number of clear occurrences during this day |
| CLEAR_COUNTER_SUM | INTEGER | Number of clearance during this day |
| ESCALATED_COUNTER_SUM | INTEGER | Number of escalated alarms during this day |
| HANDLED_COUNTER_SUM | INTEGER | Number of handled alarms during this day |
| CLOSED_COUNTER_SUM | INTEGER | Number of closed alarms during this day |
| ACK_COUNTER_SUM | INTEGER | Number of acknowledged alarms during this day |
| TERMINATED_FLAG_SUM | INTEGER | Number of terminated alarms during this day |
| ACK_BY_COUNT | INTEGER | Number of user who acknowledged alarms during this day |
| CLOSED_BY_COUNT | INTEGER | Number of user who closed alarms during this day |
| TERMINATED_BY_COUNT | INTEGER | Number of user who terminated alarms during this day |
| RELEASED_BY_COUNT | INTEGER | Number of user who released alarms during this day |
| HANDLED_BY_COUNT | INTEGER | Number of user who handled alarms during this day |
| CLEAR_DURATION_SUM | INTEGER | Total clear duration for alarms of this day |
| HANDLE_DURATION_SUM | INTEGER | Total handle duration for alarms of this day |
| CLOSE_DURATION_SUM | INTEGER | Total close duration for alarms of this day |
| ACK_DURATION_SUM | INTEGER | Total acknowledgment duration for alarms of this day |
| TERM_DURATION_SUM | INTEGER | Total termination duration for alarms of this day |
| COLLECTION_DURATION_SUM | INTEGER | Total collection duration for alarms of this day |
| IN_FAULT_DURATION_SUM | INTEGER | Total in fault duration for alarms of this day |
| IN_MANAGEMENT_DURATION_SUM | INTEGER | Total in management duration for alarms of this day |
| COLLECTION_COUNTER_SUM | INTEGER | Total counter of collection duration greater than 0 for alarms of this day |
| IN_FAULT_COUNTER_SUM | INTEGER | Total counter of in fault duration greater than 0 for alarms of this day |
| IN_MANAGEMENT_COUNTER_SUM | INTEGER | Total counter of in management duration greater than 0 for alarms of this day |
| CLEAR_DURATION_AVG | FLOAT | Average of clear duration for alarms of this day |
| HANDLE_DURATION_AVG | FLOAT | Average of handle duration for alarms of this day |
| CLOSE_DURATION_AVG | FLOAT | Average of close duration for alarms of this day |
| ACK_DURATION_AVG | FLOAT | Average of acknowledgment duration for alarms of this day |
| TERM_DURATION_AVG | FLOAT | Average of termination duration for alarms of this day |
| COLLECTION_DURATION_AVG | FLOAT | Average of collection duration for alarms of this day |
| IN_FAULT_DURATION_AVG | FLOAT | Average of in fault duration for alarms of this day |
| IN_MANAGEMENT_DURATION_AVG | FLOAT | Average of in management duration for alarms of this day |
| UPDATE_TIMESTAMP | TIMESTAMP | Time at which this record has been persisted |

### 2.1.1.4 FCT_FAULT_COMMENT table

The table FCT_FAULT_COMMENT is another fact table. It contains the comments associated to alarms. The comment is uniquely identified by the operation context of the alarm, the identifier of the alarm, the user who set the comment and the time at which the comment was set.

Here is the content of the FCT_FAULT_COMMENT table:

| Column Name | Column Type |
|---|---|
| OPERATIONCONTEXTID | INTEGER |
| IDENTIFIER | INTEGER |

| Column Name | Column Type |
|---|---|
| USERID | VARCHAR(100) |
| TIME | TIMESTAMP |
| TEXT | VARCHAR(1024) |
| UPDATE_TIMESTAMP | TIMESTAMP |

## 2.1.2 Raw events related tables

The fact table related to the raw events is the *FCT_RAW_EVENT* table.
The raw events coming from TeMIP AM via UMB are stored over time by FAS within this fact table. FCT_RAW_EVENT table will be continually updated overtime, adding new row for each new raw event.

There are 8 dimension tables associated with this fact table. They represent the different types of analysis that can be done on the raw events. Obviously, the dimension tables are also continuously updated overtime depending on the raw events received.

See the diagram below showing the 'star': with FCT_RAW_EVENT table in the middle, 'surrounded' by / linked to the dimension tables.

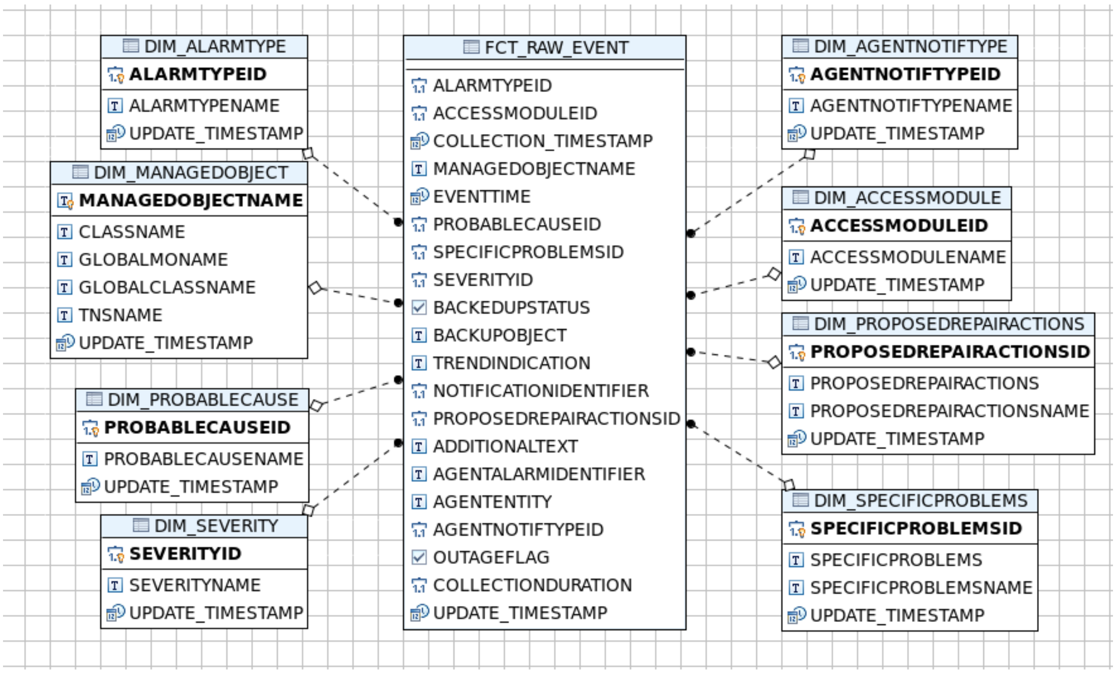


**Figure 2: FAS Datamart: raw events related tables**

### 2.1.2.1 Dimension tables

Here are described the FAS datamart dimensions tables related to raw events:

DIM_ALARMTYPE table

It is the same dimension table than the one referred by FCT_FAULT (see description upper)

DIM_MANAGEDOBJECT table

It is the same dimension table than the one referred by FCT_FAULT (see description upper)

DIM_PROBABLECAUSE table

It is the same dimension table than the one referred by FCT_FAULT (see description upper)

DIM_PROPOSEDREPAIRACTIONS table

It is the same dimension table than the one referred by FCT_FAULT (see description upper)

DIM_SEVERITY table

It is the same dimension table than the one referred by FCT_FAULT (see description upper)

DIM_SPECIFICPROBLEMS table

It is the same dimension table than the one referred by FCT_FAULT (see description upper)

DIM_AGENTNOTIFYTYPE table

| Column Name | Column Type |
|---|---|
| AGENTNOTIFTYPEID | INTEGER |
| AGENTNOTIFTYPENAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

DIM_ACCESSMODULE table

| Column Name | Column Type |
|---|---|
| ACCESSMODULEID | INTEGER |
| ACCESSMODULENAME | VARCHAR(100) |
| UPDATE_TIMESTAMP | TIMESTAMP |

### 2.1.2.2 FCT_RAW_EVENT table

Here is the description of the FCT_RAW_EVENT table:

| Column Name | Column Type | Description |
|---|---|---|
| ALARMTYPEID | INTEGER | Alarm Type Identifier (Foreign Key) |
| ACCESSMODULEID | INTEGER | Access Module Identifier (Foreign Key) |
| COLLECTION_TIMESTAMP | TIMESTAMP | Time at which the event was collected |
| MANAGEDOBJECTNAME | VARCHAR | Managed Object Name (Foreign Key) |
| EVENTTIME | TIMESTAMP | Time at which the event occurred |
| PROBABLECAUSEID | INTEGER | Probable Cause Identifier  (Foreign Key) |
| SPECIFICPROBLEMSID | INTEGER | Specific Problems Identifier (Foreign Key) |
| SEVERITYID | INTEGER | Severity Identifier (Foreign Key) |
| BACKEDUPSTATUS | BOOLEAN | A parameter that specifies whether the Managed Object is backed up by another entity or not |
| BACKUPOBJECT | VARCHAR(200) | Specifies the entity that backs up the Managed Object concerned by the raw event |
| TRENDINDICATION | VARCHAR(10) | Severity change of the event. Values can be: (*LessSevere*, *NoChange*, *MoreSevere*) |
| NOTIFICATIONIDENTIFIER | INTEGER | Provides unique identification for all raw events |
| PROPOSEDREPAIRACTIONSID | INTEGER | Proposed Repair Actions Identifier (Foreign Key) |
| ADDITIONALTEXT | VARCHAR(65000) | A free form text description of the problem reported |

| Column Name | Column Type | Description |
|---|---|---|
| AGENTALARMIDENTIFIER | VARCHAR(65000) | Agent Alarm Identifier is used with the Agent Entity attribute to uniquely identify a 3GPP alarm in the scope of a TeMIP Platform |
| AGENTENTITY | VARCHAR(65000) | Agent Entity is used with the Agent Alarm Identifier attribute to uniquely identify a 3GPP alarm in the scope of a TeMIP Platform |
| AGENTNOTIFTYPEID | INTEGER | Agent Notif Type Identifier (Foreign Key) |
| OUTAGEFLAG | BOOLEAN | Specifies if the raw event is from out of service equipment |
| COLLECTIONDURATION | INTEGER | Time needed to collect the raw event |
| UPDATE_TIMESTAMP | TIMESTAMP | Time at which this record has been persisted |

## 2.1.2.3 Summarization tables

Like for alarms, FAS provides summarization processing for raw events: aggregated raw events data are pushed into summarization tables.
Summarization tables allow to perform efficient multi dimensional analysis about raw events per *AccessModule/AlarmType/Severity* for different granularities hourly, daily, weekly, monthly.

Here are the summarization tables about raw events:

```
SUMM_HOURLY_RAW_EVENT_EQUIPMENT
SUMM_DAILY_RAW_EVENT_EQUIPMENT
SUMM_WEEKLY_RAW_EVENT_EQUIPMENT
SUMM_MONTHLY_RAW_EVENT_EQUIPMENT
```

Here is the description of the summarized table: `SUMM_HOURLY_RAW_EVENT_EQUIPMENT`
Note that the description of the other tables for the other time granularities are equivalent to the one below.

`SUMM_HOURLY_RAW_EVENT_EQUIPMENT`

| Column Name | Column Type | Description |
|---|---|---|
| TIME | TIMESTAMP | Timestamp of the Hour for which the aggregation occurred |
| ALARMTYPEID | INTEGER | Alarm Type Identifier (Foreign Key) |
| ACCESSMODULEID | INTEGER | Access Module Identifier (Foreign Key) |
| SEVERITYID | INTEGER | Severity Identifier (Foreign Key) |
| OUTAGEFLAG | BOOLEAN | Specifies if the raw event is from out of service equipment |
| AGENTNOTIFTYPEID | INTEGER | Agent Notif Type Identifier (Foreign Key) |
| MANAGEDOBJECTNAME | VARCHAR | Managed Object Name (Foreign Key) |
| PROBABLECAUSEID | INTEGER | Probable Cause Identifier (Foreign Key) |
| SPECIFICPROBLEMSID | INTEGER | Specific Problems Identifier (Foreign Key) |
| SUMMARIZED_COUNT | INTEGER | Number of raw events rows aggregated for this hour |
| RAW_EVENT_COUNTER_SUM | INTEGER | Number of raw events during this hour |
| CRITICAL_OCCURENCES_SUM | INTEGER | Number of critical occurrences during this hour |
| MAJOR_OCCURENCES_SUM | INTEGER | Number of major occurrences during this hour |
| MINOR_OCCURENCES_SUM | INTEGER | Number of minor occurrences during this hour |
| WARNING_OCCURENCES_SUM | INTEGER | Number of warning occurrences during this hour |
| INDETERMINATE_OCCURENCES_SUM | INTEGER | Number of indeterminate occurrences during this hour |
| CLEAR_OCCURENCES_SUM | INTEGER | Number of clear occurrences during this hour |

| Column Name | Column Type | Description |
|---|---|---|
| COLLECTION_DURATION_SUM | INTEGER | Total collection duration for raw events of this hour |
| COLLECTION_DURATION_AVG | FLOAT | Average of collection duration for raw events of this hour |
| UPDATE_TIMESTAMP | TIMESTAMP | Time at which this record has been persisted |

# Chapter 3
# Metadata

## 3.1 Metadata concept

The first part of this document dealt with the FAS Datamart.
This chapter deals with the **FAS metadata**.
Metadata is an .xml file that represents an abstraction of the datamart.

Thus, metadata describes:
- the definitions of Dimensions
- the definitions of Facts
- the relations between Facts & Dimensions
- the display names, descriptions, units for Unified OSS Console
- the datasource, tables and columns where to retrieve data from the Datamart
- how to retrieve fact values, and/or aggregate values (depending on the granularity requested)
- the roles allowed for accessing an object (package, dimension, fact) and under which conditions

## 3.2 FAS metadata description

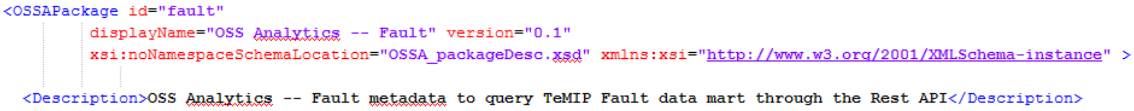
After installing the FAS kit you will find the FAS metadata at this location:

*/opt/ossa/repo-fas/FAS/metadata/ossa_fault_metadata.xml*

This FAS metadata contains identifiers, and display names of Facts and Dimensions which are directly used by the Unified OSS Console FAS reports for determining data to display.

Here is the description of the FAS metadata.

Metadata header:

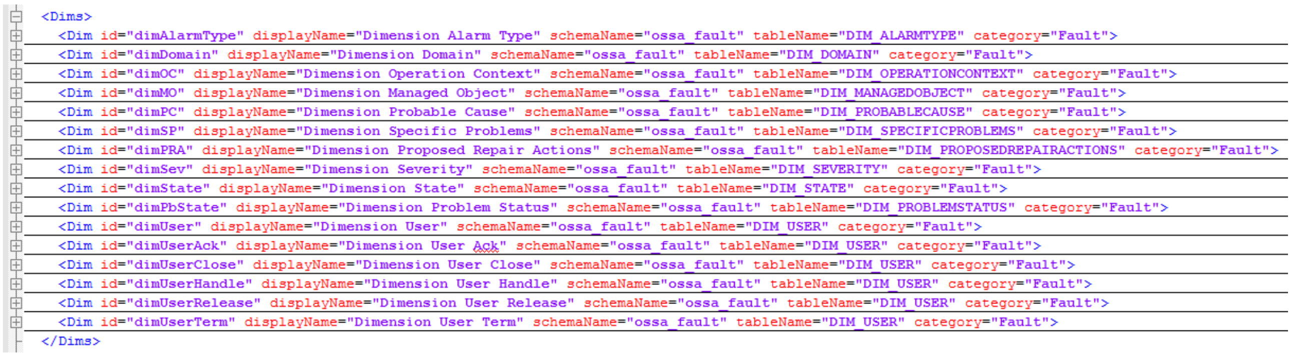


```
<OSSAPackage id="fault"
        displayName="OSS Analytics -- Fault" version="0.1"
        xsi:noNamespaceSchemaLocation="OSSA_packageDesc.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >

  <Description>OSS Analytics -- Fault metadata to query TeMIP Fault data mart through the Rest API</Description>
```

This section contains:

| | |
|---|---|
| - the name of the FAS metadata package: | fault |
| - its display name: | OSS Analytics – Fault |
| - its description | |

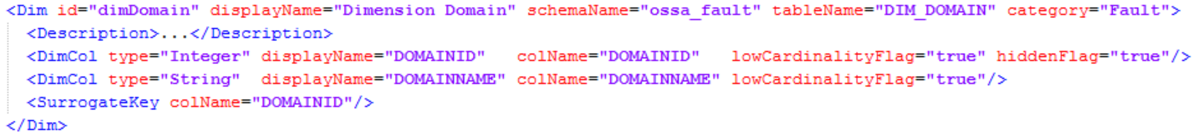
Natural dimensions declarations:



```
<Dims>
  <Dim id="dimAlarmType" displayName="Dimension Alarm Type" schemaName="ossa_fault" tableName="DIM_ALARMTYPE" category="Fault">
  <Dim id="dimDomain" displayName="Dimension Domain" schemaName="ossa_fault" tableName="DIM_DOMAIN" category="Fault">
  <Dim id="dimOC" displayName="Dimension Operation Context" schemaName="ossa_fault" tableName="DIM_OPERATIONCONTEXT" category="Fault">
  <Dim id="dimMO" displayName="Dimension Managed Object" schemaName="ossa_fault" tableName="DIM_MANAGEDOBJECT" category="Fault">
  <Dim id="dimPC" displayName="Dimension Probable Cause" schemaName="ossa_fault" tableName="DIM_PROBABLECAUSE" category="Fault">
  <Dim id="dimSP" displayName="Dimension Specific Problems" schemaName="ossa_fault" tableName="DIM_SPECIFICPROBLEMS" category="Fault">
  <Dim id="dimPRA" displayName="Dimension Proposed Repair Actions" schemaName="ossa_fault" tableName="DIM_PROPOSEDREPAIRACTIONS" category="Fault">
  <Dim id="dimSev" displayName="Dimension Severity" schemaName="ossa_fault" tableName="DIM_SEVERITY" category="Fault">
  <Dim id="dimState" displayName="Dimension State" schemaName="ossa_fault" tableName="DIM_STATE" category="Fault">
  <Dim id="dimPbState" displayName="Dimension Problem Status" schemaName="ossa_fault" tableName="DIM_PROBLEMSTATUS" category="Fault">
  <Dim id="dimUser" displayName="Dimension User" schemaName="ossa_fault" tableName="DIM_USER" category="Fault">
  <Dim id="dimUserAck" displayName="Dimension User Ack" schemaName="ossa_fault" tableName="DIM_USER" category="Fault">
  <Dim id="dimUserClose" displayName="Dimension User Close" schemaName="ossa_fault" tableName="DIM_USER" category="Fault">
  <Dim id="dimUserHandle" displayName="Dimension User Handle" schemaName="ossa_fault" tableName="DIM_USER" category="Fault">
  <Dim id="dimUserRelease" displayName="Dimension User Release" schemaName="ossa_fault" tableName="DIM_USER" category="Fault">
  <Dim id="dimUserTerm" displayName="Dimension User Term" schemaName="ossa_fault" tableName="DIM_USER" category="Fault">
</Dims>
```

This section presents all the FAS natural dimensions, with their display name (names that will be displayed in the Unified OSS Console) and with their corresponding datamart schema names and table names.

Focusing on one natural dimension, you can see its attributes declarations. For example, the Domain dimension is declared like:



```
<Dim id="dimDomain" displayName="Dimension Domain" schemaName="ossa_fault" tableName="DIM_DOMAIN" category="Fault">
  <Description>...</Description>
  <DimCol type="Integer" displayName="DOMAINID"   colName="DOMAINID"   lowCardinalityFlag="true" hiddenFlag="true"/>
  <DimCol type="String"  displayName="DOMAINNAME" colName="DOMAINNAME" lowCardinalityFlag="true"/>
  <SurrogateKey colName="DOMAINID"/>
</Dim>
```

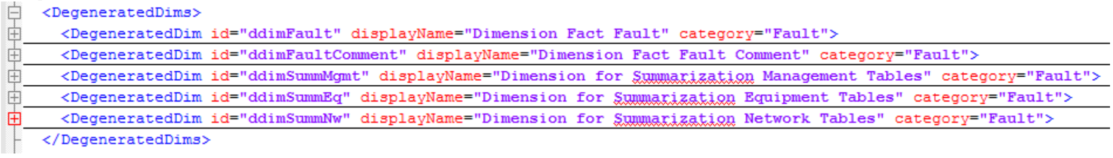
This sub-section declares the association between dimension attributes and their corresponding columns in the datamart. The *SurrogateKey* of the dimension table is also declared, and will be referenced later in the metadata within the Fact section (*<FactPatterns>*) in order to associate a fact with a dimension.
The *lowCardinalityFlag* allows the OSS Console to behave differently (in some case of display) depending on the number of different values for the dimension attribute:

- *lowCardinalityFlag* = true : very few number of different values will be stored in datamart for this dimension attribute
- *lowCardinalityFlag* = false : high number of different values will be stored in datamart for this dimension attribute.

The *category* allows to organize the dimensions presentation within the OSS Console.

The *hiddenFlag* allows the OSS Console to determine whether to expose or not the dimension attribute.

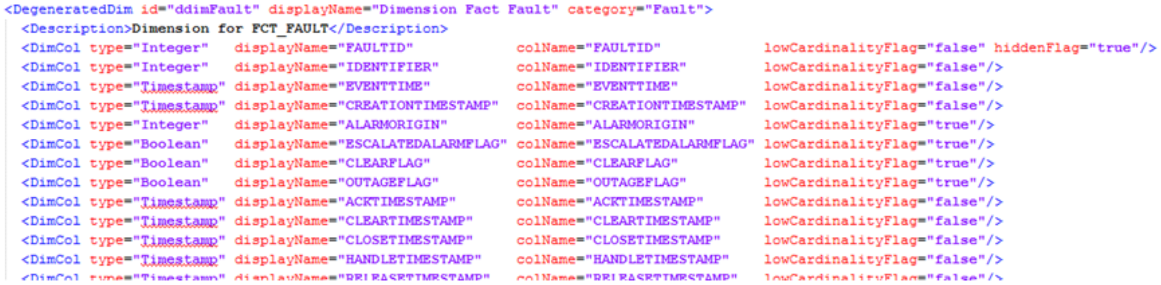
## Degenerated dimensions declarations:



```
<DegeneratedDims>
  <DegeneratedDim id="ddimFault" displayName="Dimension Fact Fault" category="Fault">
  <DegeneratedDim id="ddimFaultComment" displayName="Dimension Fact Fault Comment" category="Fault">
  <DegeneratedDim id="ddimSummMgmt" displayName="Dimension for Summarization Management Tables" category="Fault">
  <DegeneratedDim id="ddimSummEq" displayName="Dimension for Summarization Equipment Tables" category="Fault">
  <DegeneratedDim id="ddimSummNw" displayName="Dimension for Summarization Network Tables" category="Fault">
</DegeneratedDims>
```

This section presents all the FAS degenerated dimensions.
Degenerated dimensions are group of degenerated dimension attributes which act as standard dimensions from user perspective (from OSS Console) but which are stored in FACTS tables within datamart (whereas natural dimensions are stored in specific dimension tables). That is why their declarations are a bit different.

Focusing on one degenerated dimension, you can see its attributes declarations. For example, here is an extract of the *ddimFault* degenerated dimension:

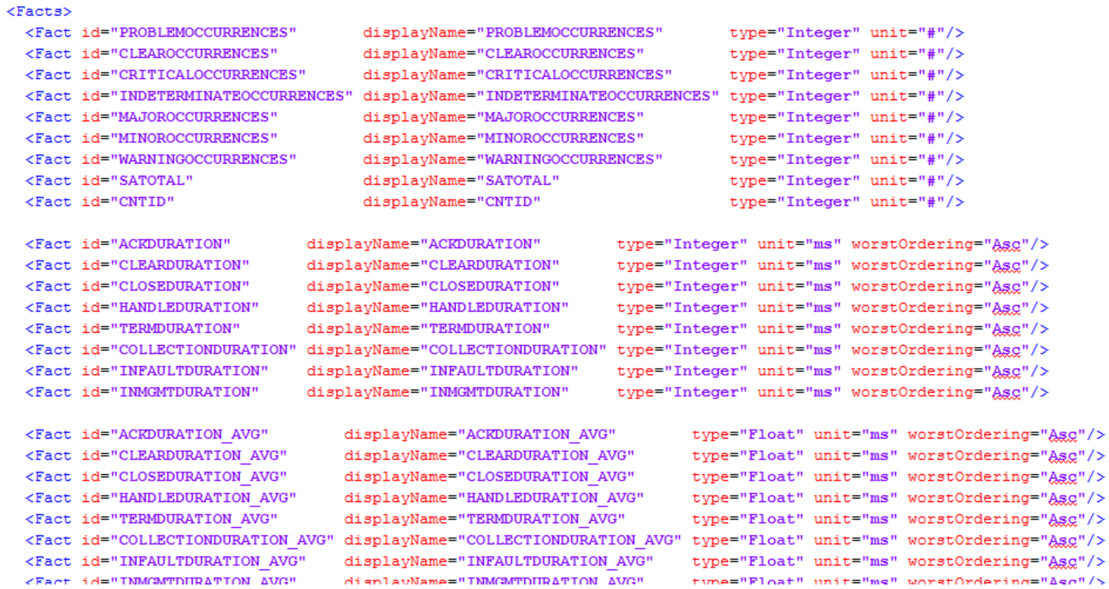


```
<DegeneratedDim id="ddimFault" displayName="Dimension Fact Fault" category="Fault">
  <Description>Dimension for FCT_FAULT</Description>
  <DimCol type="Integer"   displayName="FAULTID"             colName="FAULTID"             lowCardinalityFlag="false" hiddenFlag="true"/>
  <DimCol type="Integer"   displayName="IDENTIFIER"          colName="IDENTIFIER"          lowCardinalityFlag="false"/>
  <DimCol type="Timestamp" displayName="EVENTTIME"           colName="EVENTTIME"           lowCardinalityFlag="false"/>
  <DimCol type="Timestamp" displayName="CREATIONTIMESTAMP"   colName="CREATIONTIMESTAMP"   lowCardinalityFlag="false"/>
  <DimCol type="Integer"   displayName="ALARMORIGIN"         colName="ALARMORIGIN"         lowCardinalityFlag="true"/>
  <DimCol type="Boolean"   displayName="ESCALATEDALARMFLAG"  colName="ESCALATEDALARMFLAG"  lowCardinalityFlag="true"/>
  <DimCol type="Boolean"   displayName="CLEARFLAG"           colName="CLEARFLAG"           lowCardinalityFlag="true"/>
  <DimCol type="Boolean"   displayName="OUTAGEFLAG"          colName="OUTAGEFLAG"          lowCardinalityFlag="true"/>
  <DimCol type="Timestamp" displayName="ACKTIMESTAMP"        colName="ACKTIMESTAMP"        lowCardinalityFlag="false"/>
  <DimCol type="Timestamp" displayName="CLEARTIMESTAMP"      colName="CLEARTIMESTAMP"      lowCardinalityFlag="false"/>
  <DimCol type="Timestamp" displayName="CLOSETIMESTAMP"      colName="CLOSETIMESTAMP"      lowCardinalityFlag="false"/>
  <DimCol type="Timestamp" displayName="HANDLETIMESTAMP"     colName="HANDLETIMESTAMP"     lowCardinalityFlag="false"/>
  <DimCol type="Timestamp" displayName="RELEASETIMESTAMP"    colName="RELEASETIMESTAMP"    lowCardinalityFlag="false"/>
```

Those degenerated dimensions are referenced later in the metadata file, at the place where FACTS are declared (*colName* declared here corresponds to the column of the FACT table referencing the degenerated dimension).

## Facts declarations:

Here is an extract of the FAS Facts declarations:

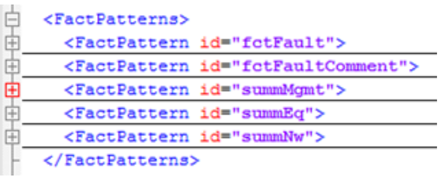


```
<Facts>
  <Fact id="PROBLEMOCCURRENCES"         displayName="PROBLEMOCCURRENCES"         type="Integer" unit="#"/>
  <Fact id="CLEAROCCURRENCES"           displayName="CLEAROCCURRENCES"           type="Integer" unit="#"/>
  <Fact id="CRITICALOCCURRENCES"        displayName="CRITICALOCCURRENCES"        type="Integer" unit="#"/>
  <Fact id="INDETERMINATEOCCURRENCES" displayName="INDETERMINATEOCCURRENCES" type="Integer" unit="#"/>
  <Fact id="MAJOROCCURRENCES"           displayName="MAJOROCCURRENCES"           type="Integer" unit="#"/>
  <Fact id="MINOROCCURRENCES"           displayName="MINOROCCURRENCES"           type="Integer" unit="#"/>
  <Fact id="WARNINGOCCURRENCES"         displayName="WARNINGOCCURRENCES"         type="Integer" unit="#"/>
  <Fact id="SATOTAL"                    displayName="SATOTAL"                    type="Integer" unit="#"/>
  <Fact id="CNTID"                      displayName="CNTID"                      type="Integer" unit="#"/>

  <Fact id="ACKDURATION"        displayName="ACKDURATION"        type="Integer" unit="ms" worstOrdering="Asc"/>
  <Fact id="CLEARDURATION"      displayName="CLEARDURATION"      type="Integer" unit="ms" worstOrdering="Asc"/>
  <Fact id="CLOSEDURATION"      displayName="CLOSEDURATION"      type="Integer" unit="ms" worstOrdering="Asc"/>
  <Fact id="HANDLEDURATION"     displayName="HANDLEDURATION"     type="Integer" unit="ms" worstOrdering="Asc"/>
  <Fact id="TERMDURATION"       displayName="TERMDURATION"       type="Integer" unit="ms" worstOrdering="Asc"/>
  <Fact id="COLLECTIONDURATION" displayName="COLLECTIONDURATION" type="Integer" unit="ms" worstOrdering="Asc"/>
  <Fact id="INFAULTDURATION"    displayName="INFAULTDURATION"    type="Integer" unit="ms" worstOrdering="Asc"/>
  <Fact id="INMGMTDURATION"     displayName="INMGMTDURATION"     type="Integer" unit="ms" worstOrdering="Asc"/>

  <Fact id="ACKDURATION_AVG"        displayName="ACKDURATION_AVG"        type="Float" unit="ms" worstOrdering="Asc"/>
  <Fact id="CLEARDURATION_AVG"      displayName="CLEARDURATION_AVG"      type="Float" unit="ms" worstOrdering="Asc"/>
  <Fact id="CLOSEDURATION_AVG"      displayName="CLOSEDURATION_AVG"      type="Float" unit="ms" worstOrdering="Asc"/>
  <Fact id="HANDLEDURATION_AVG"     displayName="HANDLEDURATION_AVG"     type="Float" unit="ms" worstOrdering="Asc"/>
  <Fact id="TERMDURATION_AVG"       displayName="TERMDURATION_AVG"       type="Float" unit="ms" worstOrdering="Asc"/>
  <Fact id="COLLECTIONDURATION_AVG" displayName="COLLECTIONDURATION_AVG" type="Float" unit="ms" worstOrdering="Asc"/>
  <Fact id="INFAULTDURATION_AVG"    displayName="INFAULTDURATION_AVG"    type="Float" unit="ms" worstOrdering="Asc"/>
  <Fact id="INMGMTDURATION_AVG"     displayName="INMGMTDURATION_AVG"     type="Float" unit="ms" worstOrdering="Asc"/>
```

This section only lists all the Facts which are stored in the FAS datamart.
They are referenced and described later in the metadata, within the *FactCalculation* section.

## Facts Patterns declarations:

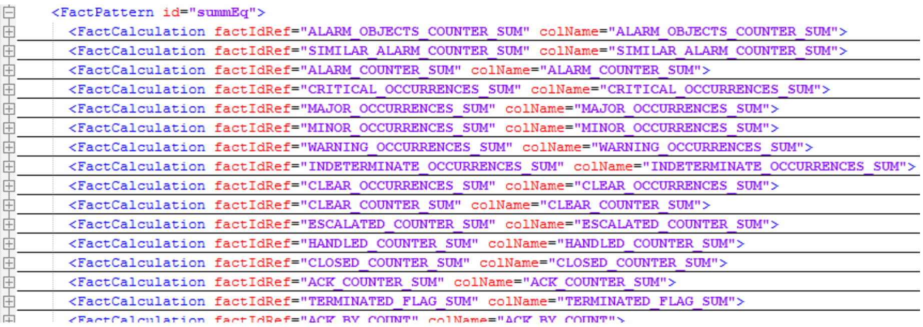
Here are the FAS Fact Patterns:



```
<FactPatterns>
  <FactPattern id="fctFault">
  <FactPattern id="fctFaultComment">
  <FactPattern id="summMgmt">
  <FactPattern id="summEq">
  <FactPattern id="summNw">
</FactPatterns>
```

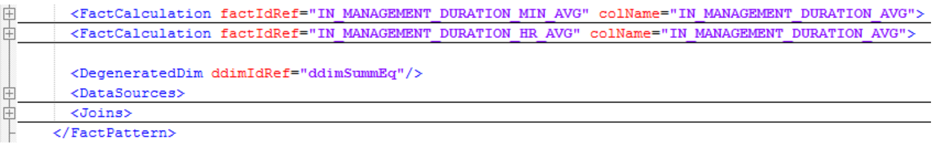
A *FactPattern* represents a set of Facts which can be found in different Datamart tables having the same table definition, but in which the fact values are time-aggregated differently; for example: hourly, daily, weekly, monthly aggregation. You can see above that FAS metadata exposes the facts concerning the raw alarms (id=”*fctFault*”), the facts containing the alarm comments (id=”*fctFaultComment*”), and the 3 facts concerning the summarized data, for the 3 kinds of summarizations : Management (id=”*summMgmt*”), Equipment (id=”*summEq*”), Network (id=”*summNw*”).

Here is an extract of the fact pattern *summEq* describing the facts issued from summarized tables dealing with Equipment analysis:



```
<FactPattern id="summEq">
  <FactCalculation factIdRef="ALARM_OBJECTS_COUNTER_SUM" colName="ALARM_OBJECTS_COUNTER_SUM">
  <FactCalculation factIdRef="SIMILAR_ALARM_COUNTER_SUM" colName="SIMILAR_ALARM_COUNTER_SUM">
  <FactCalculation factIdRef="ALARM_COUNTER_SUM" colName="ALARM_COUNTER_SUM">
  <FactCalculation factIdRef="CRITICAL_OCCURRENCES_SUM" colName="CRITICAL_OCCURRENCES_SUM">
  <FactCalculation factIdRef="MAJOR_OCCURRENCES_SUM" colName="MAJOR_OCCURRENCES_SUM">
  <FactCalculation factIdRef="MINOR_OCCURRENCES_SUM" colName="MINOR_OCCURRENCES_SUM">
  <FactCalculation factIdRef="WARNING_OCCURRENCES_SUM" colName="WARNING_OCCURRENCES_SUM">
  <FactCalculation factIdRef="INDETERMINATE_OCCURRENCES_SUM" colName="INDETERMINATE_OCCURRENCES_SUM">
  <FactCalculation factIdRef="CLEAR_OCCURRENCES_SUM" colName="CLEAR_OCCURRENCES_SUM">
  <FactCalculation factIdRef="CLEAR_COUNTER_SUM" colName="CLEAR_COUNTER_SUM">
  <FactCalculation factIdRef="ESCALATED_COUNTER_SUM" colName="ESCALATED_COUNTER_SUM">
  <FactCalculation factIdRef="HANDLED_COUNTER_SUM" colName="HANDLED_COUNTER_SUM">
  <FactCalculation factIdRef="CLOSED_COUNTER_SUM" colName="CLOSED_COUNTER_SUM">
  <FactCalculation factIdRef="ACK_COUNTER_SUM" colName="ACK_COUNTER_SUM">
  <FactCalculation factIdRef="TERMINATED_FLAG_SUM" colName="TERMINATED_FLAG_SUM">
  <FactCalculation factIdRef="ACK_BY_COUNT" colName="ACK_BY_COUNT">
```

. . .



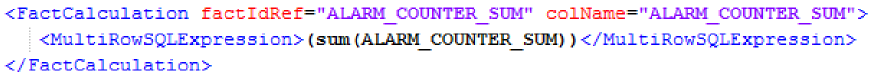
```
  <FactCalculation factIdRef="IN_MANAGEMENT_DURATION_MIN_AVG" colName="IN_MANAGEMENT_DURATION_AVG">
  <FactCalculation factIdRef="IN_MANAGEMENT_DURATION_HR_AVG" colName="IN_MANAGEMENT_DURATION_AVG">

  <DegeneratedDim ddimIdRef="ddimSummEq"/>
  <DataSources>
  <Joins>
</FactPattern>
```

Within a FactPattern, are declared the:

- *<FactCalculation> :* describing how to retrieve and calculate fact values (for facts referencing already defined<Fact> entries)
- *<DegeneratedDim>* : degenerated dimensions associated to the fact. This attribute *DegeneratedDim ddimIdRef* just references <*DegeneratedDim*> already defined entries from the first sections of the metadata
- *<DataSources>* : the location of the facts within the FAS datamart
- *<Joins>* : the associated natural dimensions of the fact

## Fact Pattern FactCalculation declarations:

Here is a *FactCalculation* declared in the *FactPattern summEq*:

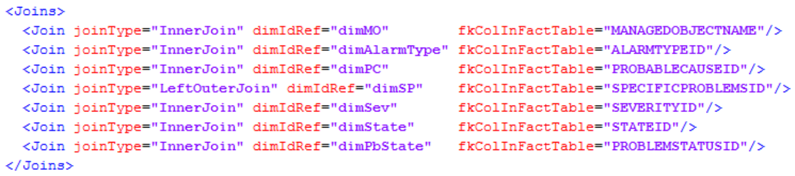


```
<FactCalculation factIdRef="ALARM_COUNTER_SUM" colName="ALARM_COUNTER_SUM">
   <MultiRowSQLExpression>(sum(ALARM_COUNTER_SUM))</MultiRowSQLExpression>
</FactCalculation>
```

- *factIdRef* : this *FactCalculation* refers to the *Fact Id ALARM_COUNTER_SUM* (previously declared in <*Fact*>)
- *colName* : the fact retrieval is done from the column named *ALARM_COUNTER_SUM*
- *MultiRowSQLExpression* : defines the SQL expression to be used when time aggregation or dimension aggregation is requested. In this case, when requesting the counter of alarms for a granularity higher than the time granularity defined for this table, the *sum(ALARM_COUNTER_SUM)* is applied on the related fact rows.

## Fact Pattern Joins declarations:

Here are the *Joins* declared in the *FactPattern summEq*:

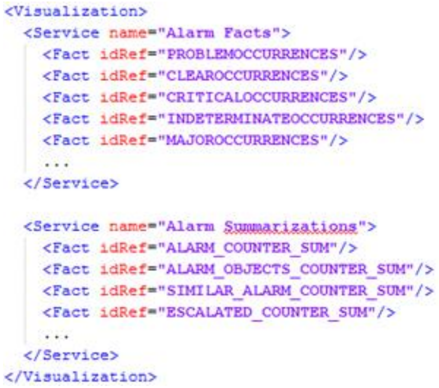


```
<Joins>
  <Join joinType="InnerJoin" dimIdRef="dimMO"        fkColInFactTable="MANAGEDOBJECTNAME"/>
  <Join joinType="InnerJoin" dimIdRef="dimAlarmType" fkColInFactTable="ALARMTYPEID"/>
  <Join joinType="InnerJoin" dimIdRef="dimPC"        fkColInFactTable="PROBABLECAUSEID"/>
  <Join joinType="LeftOuterJoin" dimIdRef="dimSP"    fkColInFactTable="SPECIFICPROBLEMSID"/>
  <Join joinType="InnerJoin" dimIdRef="dimSev"       fkColInFactTable="SEVERITYID"/>
  <Join joinType="InnerJoin" dimIdRef="dimState"     fkColInFactTable="STATEID"/>
  <Join joinType="InnerJoin" dimIdRef="dimPbState"   fkColInFactTable="PROBLEMSTATUSID"/>
</Joins>
```

They declare all the associations from the FactPattern to its natural dimensions in term of type of join (*joinType*), natural dimension referenced (*dimIdRef* previously declared in <*Dims*>), and foreign key (*fkColInFactTable*) used for the association.

## Visualization declaration:

Here is the *Visualization* declaration:



```
<Visualization>
  <Service name="Alarm Facts">
    <Fact idRef="PROBLEMOCCURRENCES"/>
    <Fact idRef="CLEAROCCURRENCES"/>
    <Fact idRef="CRITICALOCCURRENCES"/>
    <Fact idRef="INDETERMINATEOCCURRENCES"/>
    <Fact idRef="MAJOROCCURRENCES"/>
    ...
  </Service>

  <Service name="Alarm Summarizations">
    <Fact idRef="ALARM_COUNTER_SUM"/>
    <Fact idRef="ALARM_OBJECTS_COUNTER_SUM"/>
    <Fact idRef="SIMILAR_ALARM_COUNTER_SUM"/>
    <Fact idRef="ESCALATED_COUNTER_SUM"/>
    ...
  </Service>
</Visualization>
```

This Visualization section is used by the OSS Console in order to organize the facts visualization per service and possible subfolders.

With this FAS *Visualization*, in OSS Console the facts concerning the alarms will be put in a folder named “*Alarm facts*” and the aggregated facts (coming from summarization) will be put in a folder named “*Alarm Summarizations*”.

In the same way, facts concerning raw events and aggregated data have their own section also.

# Chapter 4

# FAS default reports

Please refer to the *HPE Fault Analytics and Statistics user guide* for a presentation of the FAS default reports.

# Chapter 5
# FAS customization

## 5.1 Mapping rules from TeMIP alarms to FAS datamart

### 5.1.1 Specific problems configuration

Fault Analytics and Statistics lets the integrator choose whether he needs to support TeMIP alarm Specific Problem or not.
The choice is driven by the configuration of FAS as introduced in the “*Configuring FAS alarm export application*” section of the *HPE Fault Analytics and Statistics – Installation and Administration guide*.

The command for setting the FAS parameter values is:

```
${OSSA_HOME}/bin/ossa-repo.sh setParam OSSA_FAULT AlarmExportConfig <XX>
${OSSA_HOME}/bin/ossa-repo.sh reload
```

where `<XX>` is a json description of the configuration.
Within this json description you can define which alarm Specific Problems will be exported to Vertica or not. The setting is done per TeMIP Global Class.

See the following example of FAS configuration json description:

```
{
    "consumerGroupConfigs": [
        {
            "consumerGroupProperties": {
                "group.id": "st-01",
                "zookeeper.connect": "myhost.com:2181",
                "auto.commit.enable": "true",
                "auto.offset.reset": "smallest"
            },
            "consumerThreadNumber": 16
        }
    ],
    "temipClassSpecificProblems": [
        {
            "temipClass": "OSI_SYSTEM",
            "supportsSP": true
        },{
            "temipClass": "",
            "supportsSP": false
        }
    ]
}
```

The configuration example above defines that Specific Problems attribute of alarms which TeMIP Global Class starts with *OSI_SYSTEM* will be persisted into *DIM_SPECIFICPROBLEMS* table. Obviously, those alarms stored into *FCT_FAULT* will have their *SPECIFICPROBLEMSID* referencing the corresponding *DIM_SPECIFICPROBLEMS*.
The configuration example above also says that alarms with other TeMIP Global Class will not have their specific problems stored into *DIM_SPECIFICPROBLEMS*.

**NOTE:** This example was focused on the Specific Problems configuration for **alarm** consumption, but note that the same kind of configuration can be applied for the consumption of **raw events**.

# 5.2 FAS solution enrichment

## 5.2.1 User Defined Attributes example

In order to introduce the possible customizations of FAS, we will take the common example of adding TeMIP User Defined Attributes within FAS solution.

### 5.2.1.1 User Defined Attributes datamart enrichment

In addition to the standard TeMIP attributes, FAS can handle TeMIP User Defined Attributes.
In the following, we will take as example one new user defined attribute named MYUDA. Obviously, the same steps must be done for all the user defined attributes required in Datamart.

**NOTE:** This example focus on User Defined Attributes for alarms and thus present different configurations steps related to alarms, but note that the same kind of configurations could be applied for the raw events.

First, you need to create your own .ddl file containing the declaration of the additional columns corresponding to the user defined attributes needed in the FAS datamart schema. Then, you will have to load your .ddl file, as FAS database user.

FCT_FAULT table

Start by adding column definition for FCT_FAULT table.

```
ALTER TABLE FCT_FAULT ADD COLUMN <Column Name> <Column Constraint>;
```

For example:

```
ALTER TABLE FCT_FAULT ADD COLUMN MYUDA VARCHAR(1024);
```

Then, enrich the USER_DEFINED_ATTRIBUTE table.

```
INSERT INTO USER_DEFINED_ATTRIBUTE (ATTR_ID, ATTR_NAME, ATTR_TYPE, ATTR_SCOPE, MAX_SIZE)
VALUES (<Attribute ID>, '<Column Name>', '<Attribute Type>', '<Attribute Scope>', <Max
Size>);
```

For example:

```
INSERT INTO USER_DEFINED_ATTRIBUTE (ATTR_ID, ATTR_NAME, ATTR_TYPE, ATTR_SCOPE, MAX_SIZE)
VALUES (10056, 'MYUDA', 'String', 'alarm', 1024);
```

The `<Attribute ID>` must correspond to the ID of the attribute within TeMIP.
The `<Column Name>` must be the name of the column you just added in FCT_FAULT.
The `<Attribute Type>` is used for classification, you can specify "String", "Int" or "Long".
The `<Attribute Scope>` is used for determining which FAS consumption flow is concerned by this user defined attribute. <Attribute Scope> value can be: 'alarm' , 'raw event' or NULL (if both flows, alarms and raw events, are concerned by this user defined attribute).
The `<Max Size>` is useful only when the type of the column you add is "VARCHAR". It tells FAS to truncate the extra characters when needed. Please make sure that the value corresponds to the column definition from FCT_FAULT table.

Now, you must restart your OSSAF Server:

as *ossa* linux user, in a terminal where you have sourced the ossa environment:

```
$ jbossstop
$ jbossstart
```

FAS is now able to export User Defined Attributes to FAS Datamart table FCT_FAULT.

### Summarizations tables

**Important point**: based on the UOCv2 views you want to design (dealing with summarized data), determine the necessary User Defined Attributes to add within the summarized tables definitions for each of the three summarization categories: Network, Management or Equipment.

So, add the column definitions for each required User Defined Attribute within all required summarized tables by executing:

```
ALTER TABLE SUMM_XXX ADD COLUMN <Column Name> <Column Type>;
```

In the following example, we add the MYUDA User Defined Attribute in the 11 existing summarized tables:

```
ALTER TABLE SUMM_DAILY_EQUIPMENT ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_WEEKLY_EQUIPMENT ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_MONTHLY_EQUIPMENT ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_HOURLY_MANAGEMENT ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_DAILY_MANAGEMENT ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_WEEKLY_MANAGEMENT ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_MONTHLY_MANAGEMENT ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_HOURLY_NETWORK ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_DAILY_NETWORK ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_WEEKLY_NETWORK ADD COLUMN MYUDA VARCHAR(200);
ALTER TABLE SUMM_MONTHLY_NETWORK ADD COLUMN MYUDA VARCHAR(200);
```

and, modify the UNIQUE constraints on the tables you have modified, for example:

```
-- first drop existing unique constraints on summarized table
ALTER TABLE SUMM_DAILY_EQUIPMENT DROP CONSTRAINT UQ_SUMM_DAILY_EQUIPMEN;
ALTER TABLE SUMM_WEEKLY_EQUIPMENT DROP CONSTRAINT UQ_SUMM_WEEKLY_EQUIPM;
ALTER TABLE SUMM_MONTHLY_EQUIPMENT DROP CONSTRAINT UQ_SUMM_MONTHLY_EQ;
ALTER TABLE SUMM_HOURLY_MANAGEMENT DROP CONSTRAINT UQ_SUMM_HOURLY_MANAGEME;
ALTER TABLE SUMM_DAILY_MANAGEMENT DROP CONSTRAINT UQ_SUMM_DAILY_MANAGEME;
ALTER TABLE SUMM_WEEKLY_MANAGEMENT DROP CONSTRAINT UQ_SUMM_WEEKLY_MGMT;
ALTER TABLE SUMM_MONTHLY_MANAGEMENT DROP CONSTRAINT UQ_SUMM_MONTHLY_MGMT;
ALTER TABLE SUMM_HOURLY_NETWORK DROP CONSTRAINT UQ_SUMM_HOURLY_NETWORK;
ALTER TABLE SUMM_DAILY_NETWORK DROP CONSTRAINT UQ_SUMM_DAILY_NETWORK_;
ALTER TABLE SUMM_WEEKLY_NETWORK DROP CONSTRAINT UQ_SUMM_WEEKLY_NETWORK;
ALTER TABLE SUMM_MONTHLY_NETWORK DROP CONSTRAINT UQ_SUMM_MONTHLY_NETWORK;
      -- then create the new unique constraints with new user defined attribute
ALTER TABLE SUMM_DAILY_EQUIPMENT ADD CONSTRAINT UQ_SUMM_DAILY_EQUIPMEN UNIQUE
   (TIME, MANAGEDOBJECTNAME, ALARMTYPEID, PROBABLECAUSEID, SPECIFICPROBLEMSID,
    PROBLEMSTATUSID, SEVERITYID, STATEID, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_WEEKLY_EQUIPMENT ADD CONSTRAINT UQ_SUMM_WEEKLY_EQUIPM UNIQUE
   (TIME, MANAGEDOBJECTNAME, ALARMTYPEID, PROBABLECAUSEID, SPECIFICPROBLEMSID,
    PROBLEMSTATUSID, SEVERITYID, STATEID, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_MONTHLY_EQUIPMENT ADD CONSTRAINT UQ_SUMM_MONTHLY_EQ UNIQUE
```

```
   (TIME, MANAGEDOBJECTNAME, ALARMTYPEID, PROBABLECAUSEID, SPECIFICPROBLEMSID,
    PROBLEMSTATUSID, SEVERITYID, STATEID, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_HOURLY_MANAGEMENT ADD CONSTRAINT UQ_SUMM_HOURLY_MANAGEME UNIQUE
   (TIME, OPERATIONCONTEXTID, ACKUSERID, CLOSEUSERID, HANDLEUSERID, RELEASEUSERID,
    TERMUSERID, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_DAILY_MANAGEMENT ADD CONSTRAINT UQ_SUMM_DAILY_MANAGEME UNIQUE
   (TIME, OPERATIONCONTEXTID, ACKUSERID, CLOSEUSERID, HANDLEUSERID, RELEASEUSERID,
    TERMUSERID, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_WEEKLY_MANAGEMENT ADD CONSTRAINT UQ_SUMM_WEEKLY_MGMT UNIQUE
   (TIME, OPERATIONCONTEXTID, ACKUSERID, CLOSEUSERID, HANDLEUSERID, RELEASEUSERID,
    TERMUSERID, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_MONTHLY_MANAGEMENT ADD CONSTRAINT UQ_SUMM_MONTHLY_MGMT UNIQUE
    (TIME, OPERATIONCONTEXTID, ACKUSERID, CLOSEUSERID, HANDLEUSERID, RELEASEUSERID,
     TERMUSERID, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_HOURLY_NETWORK ADD CONSTRAINT UQ_SUMM_HOURLY_NETWORK UNIQUE
    (TIME, DOMAINID, OPERATIONCONTEXTID, MANAGEDOBJECTNAME, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_DAILY_NETWORK ADD CONSTRAINT UQ_SUMM_DAILY_NETWORK_ UNIQUE
    (TIME, DOMAINID, OPERATIONCONTEXTID, MANAGEDOBJECTNAME, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_WEEKLY_NETWORK ADD CONSTRAINT UQ_SUMM_WEEKLY_NETWORK UNIQUE
    (TIME, DOMAINID, OPERATIONCONTEXTID, MANAGEDOBJECTNAME, OUTAGEFLAG, MYUDA);
ALTER TABLE SUMM_MONTHLY_NETWORK ADD CONSTRAINT UQ_SUMM_MONTHLY_NETWORK UNIQUE
    (TIME, DOMAINID, OPERATIONCONTEXTID, MANAGEDOBJECTNAME, OUTAGEFLAG, MYUDA);
```

Please ensure that your Datamart has been enhanced with new User Defined Attributes within the FCT_FAULT table and within the summarized tables, and that the constraints have been modified accordingly.

### 5.2.1.2 User Defined Attributes summarization batch job enrichment

You must now modify the summarization batch job in order to take into account the User Defined Attributes.

As *ossa* linux user, modify the summarization batch job description (please backup it first):

```
 ${OSSA_HOME}/repo-fas/FAS/batch/FASsummJob.xml
```

For each summarization on which you want to handle User Defined Attribute, add its column name in the property `src_dimensions`:

```
<property name="src_dimensions" value=". . .,MYUDA" />
```

For example:

```
<property name="src_dimensions"
value="OPERATIONCONTEXTID,ACKUSERID,CLOSEUSERID,HANDLEUSERID,
       RELEASEUSERID,TERMUSERID,OUTAGEFLAG,MYUDA" />
```

If all the summarizations needs to have the MYUDA User Defined Attribute, 11 modifications of this kind must be done in the summarization batch job description.

Then, load the new summarization batch job description into the OSSA server:
as *ossa* linux user, in a terminal where you have sourced the ossa environment:

```
ossa-repo.sh loadParam FAS FASsummJob.xml \
       ${OSSA_HOME}/repo-fas/FAS/batch/FASsummJob.xml
ossa-repo.sh reload
```

The new summarization batch job will now be scheduled, and the new User Defined Attributes will be populated in the summarized tables.

### 5.2.1.3 User Defined Attributes FAS metadata enrichment

Start by backuping the original fault metadata:
`${OSSA_HOME}/repo-fas/FAS/metadata/ossa_fault_metadata.xml`

Then, enhance it by adding the necessary User Defined Attributes for the four degenerated dimensions definitions *ddimFault*, *ddimSummMgmt*, *ddimSummEq*, *ddimSummNw*.

For example:

```
…
<DegeneratedDim id="ddimFault" displayName="Dimension Fact Fault" category="Fault">
   <Description>Dimension for FCT_FAULT</Description>
…
   <DimCol type="String" lowCardinalityFlag="false" displayName="My User Defined Attribute"
colName="MYUDA"/>
…

<DegeneratedDim id="ddimSummMgmt" displayName="Dimension for Summarization Management
Tables" category="Fault">
      <Description>Dimension for Summarization Management Tables</Description>
             …
      <DimCol type="String" displayName="MYUDA" colName="MYUDA" lowCardinalityFlag="false"/>
…

<DegeneratedDim id="ddimSummEq" displayName="Dimension for Summarization Equipment Tables"
category="Fault">
      <Description>Dimension for Summarization Equipment Tables</Description>
             …
      <DimCol type="String" displayName="MYUDA" colName="MYUDA" lowCardinalityFlag="false"/>
…

<DegeneratedDim id="ddimSummNw" displayName="Dimension for Summarization Network Tables"
category="Fault">
      <Description>Dimension for Summarization Network Tables</Description>
      …
      <DimCol type="String" displayName="MYUDA" colName="MYUDA" lowCardinalityFlag="false"/>
 …
```

Then load the modified metadata (with the FAS views and workspaces also):

```
ossa-repo.sh loadMetadataViewsWks \
             ${OSSA_HOME}/repo-fas/FAS/metadata/ossa_fault_metadata.xml \
             ${OSSA_HOME}/repo-fas/FAS/ui/views.json \
             ${OSSA_HOME}/repo-fas/FAS/ui/workspaces.json
ossa-repo.sh reload
```

OSSAF Server is now ready to handle requests concerning your User Defined Attributes.

### 5.2.1.4 User Defined Attributes FAS reports

Once the FAS/OSSA Server has been enriched with User Defined Attributes, your OSS Console reports might need to be enhanced or new ones might need to be created.
For this, please refer to *HPE Fault Analytics and Statistics User Guide*, and to *HPE UOC User Guide*.

## 5.2.2 Date Time Dimension Configuration

In order to get the correct reports for FAS seasonal/cyclical variations, the user has to enrich the date time dimensions with holiday, event day, working hours and shift information. FAS provides scripts in the directory “<installation directory>/bin” to accomplish these tasks. Before executing the following commands, the below command must be executed.

```
# source ossa_env.sh
```

### 5.2.2.1 Holiday configuration

The holiday configuration command provides features to add a holiday, list holidays and remove a holiday.

The following command is an example to add holidays. There are two parameters, type and date. The type parameter has two values, 0 for a holiday every year, and 1 for a holiday of one year. The date parameter is the date of the added holiday in format yyyy-mm-dd (for a holiday of one year) or mm-dd (for a holiday every year).

```
# ./holiday_config.sh add <type> <date: yyyy-mm-dd or mm-dd format>
```

The holiday_config command can also list all holidays in a specific time period. There are two parameters, start_date and end_date. The start_date parameter is the date of the first day in the time period in the format yyyy-mm-dd; The end_date parameter is the date of the last day in the time period in the format yyyy-mm-dd.

```
# ./holiday_config.sh list <start_date: yyyy-mm-dd> <end_date: yyyy-mm-dd>
```

In order to remove a holiday, the below command can be executed. It has the similar parameters as the adding command.

```
# ./holiday_config.sh remove <type> <date: yyyy-mm-dd or mm-dd format>
```

### 5.2.2.2 Event configuration

The event configuration command provides features to add an event, list events and remove an event.

The following command is an example to add events. There are two or three parameters, type and date. The type parameter has two values, 0 for one day event, and 1 for several days' event. The event_name parameter is the name or title of the event. For one day event, the date parameter is the date of the added event in the format mm-dd. For several days’ event, the start_date parameter is the date of the first day of the event in the format yyyy-mm-dd; The end_date parameter is the date of the last day of the event in the format yyyy-mm-dd.

```
# ./event_config.sh add 0 <event_name> <date: yyyy-mm-dd>
# ./event_config.sh add 1 <event_name> <start_date> <end_date>
```

The event_config command can also list all events in a specific time period. There are two parameters, start_date and end_date. The start_date parameter is the date of the first day in the time period in the format yyyy-mm-dd; The end_date parameter is the date of the last day in the time period in the format yyyy-mm-dd.

```
# ./event_config.sh list <start_date: yyyy-mm-dd> <end_date: yyyy-mm-dd>
```

In order to remove an event, the below command can be executed. It has a parameter event_name to tell which event to be removed.

```
# ./event_config.sh remove <event_name>
```

### 5.2.2.3 Working hour configuration

The working hour configuration command provides features to set working hours, list working hours and unset working hours.

The following command is an example to set working hours from 9:00 to 17:00. There are two parameters, start_hour and end_hour. The start_hour parameter is the first working hour everyday (inclusively). The end_hour parameter is the last working hour everyday (exclusively).

```
Command format:
      working_hours_config.sh set <start_hour> <end_hour>

For example,
# ./working_hours_config.sh set 9 17
```

The working hour_config command can also list all working hours.

```
# ./working_hours_config.sh list
```

In order to unset working hours, the below command can be executed. It has the similar parameters as the “set” command.

```
# ./working_hours_config.sh unset <start_hour> <end_hour>
```

### 5.2.2.4 Shift configuration

The shift configuration command provides features to add a shift, list all shifts and remove a shift.

The following commands are examples to configure 3 shifts, each of which has 8 hours. There are 3 parameters, shift_name, start_hour, and end_hour. The shift_name is the name of the shift. The start_hour is the first hour of the shift everyday (inclusively). The end_hour is the last hour of the shift everyday (exclusively).

```
Command format:
      shift_config.sh add <shift_name> <start_hour> <end_hour>

For example,
# ./shift_config.sh add shift1 0 8
# ./shift_config.sh add shift2 8 16
# ./shift_config.sh add shift3 16 24
```

The shift_config command can also list all shifts.

```
# ./shift_config.sh list
```

In order to remove a shift with a specific shift name, the below command can be executed. It has a parameter “shift_name” to indicate which shift to be removed.

```
# ./shift_config.sh remove <shift_name>
```

### 5.2.2.5 Operator processing rate configuration

The operator processing rate configuration command provides features to set the operator processing rate.

The following command is an example of setting the operator processing command:

```
Command format:
      operator_processing_rate_config.sh set <operator_processing_rate>
For example,
# ./operator_processing_rate_config.sh set 1000
```

The command without parameters will set operator processing rate to 100 by default and show the help information.

## 5.2.3 Other customizations

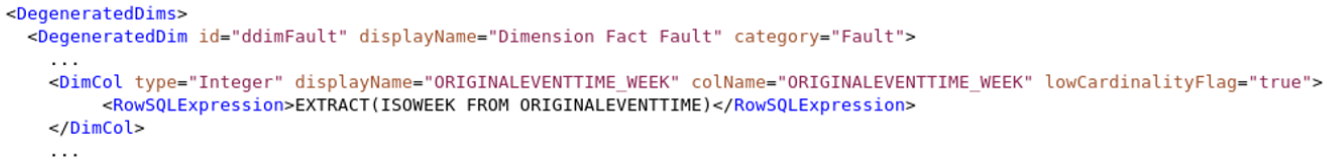
### 5.2.3.1 Other metadata enrichment

In the previous section, what has been highlighted concerning metadata enrichment is about addition of new simple degenerated dimensions.

But obviously, you can also add:
- more complex degenerated dimensions
- fact attributes

This is what is presented below.

Addition of SQL defined degenerated dimension:
In the same way that was described previously, a new degenerated dimension attribute can be added within metadata in the <*DegeneratedDims*> section.
Here we take as example the addition of the a new *DegeneratedDim* based on the original event time of the alarm and that gives the number of the week in the year. Here is the example of this new degenerated dim named:
`ORIGINALEVENTTIME_WEEK`



```
<DegeneratedDims>
  <DegeneratedDim id="ddimFault" displayName="Dimension Fact Fault" category="Fault">
    ...
    <DimCol type="Integer" displayName="ORIGINALEVENTTIME_WEEK" colName="ORIGINALEVENTTIME_WEEK" lowCardinalityFlag="true">
         <RowSQLExpression>EXTRACT(ISOWEEK FROM ORIGINALEVENTTIME)</RowSQLExpression>
    </DimCol>
    ...
```

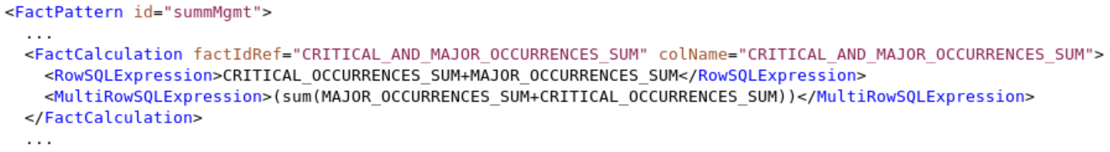
This shows that the retrieval of degenerated dimension value can be done thanks to a SQL clause.

Addition of natural dimensions
Concerning the addition of natural dimension, you just have to declare it in the <*Dims*> section, with the declaration of the corresponding Datamart table, column and declaration of the surrogate key. You can refer to the previous chapter *FAS metadata description / natural dimension declarations* for more details.

Note that your new natural dimension will certainly have to be associated from a fact: thus, the <*Joins*> part of the corresponding <*FactPattern*> will have to be enhanced.
Please refer to the previous chapter *FAS metadata description / Fact Pattern Joins* declarations for more details.

Addition of fact attributes
Concerning the addition of fact attributes, you must declare it in the relevant <*FactCalculation*> section.

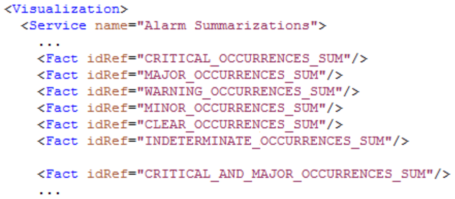
In this example, for the *FactPattern summMgmt* , we declare a new fact attribute *CRITICAL_AND_MAJOR_OCCURRENCES_SUM* which is the sum of existing fact attributes *CRITICAL__OCCURRENCES_SUM* and *MAJOR_OCCURRENCES_SUM*.



```
<FactPattern id="summMgmt">
  ...
  <FactCalculation factIdRef="CRITICAL_AND_MAJOR_OCCURRENCES_SUM" colName="CRITICAL_AND_MAJOR_OCCURRENCES_SUM">
    <RowSQLExpression>CRITICAL_OCCURRENCES_SUM+MAJOR_OCCURRENCES_SUM</RowSQLExpression>
    <MultiRowSQLExpression>(sum(MAJOR_OCCURRENCES_SUM+CRITICAL_OCCURRENCES_SUM))</MultiRowSQLExpression>
  </FactCalculation>
  ...
```

The *<RowSQLExpression>CRITICAL_OCCURRENCES_SUM+MAJOR_OCCURRENCES_SUM</RowSQLExpression>* defines the value of the fact when no time aggregation is requested.
When time aggregation is requested, the fact rows implied within the time granularity will be aggregated with the formula described in:
*<MultiRowSQLExpression>(sum(MAJOR_OCCURRENCES_SUM+CRITICAL_OCCURRENCES_SUM))</MultiRowSQLExpression>*
In this example, we define the fact attribute within the *FactPattern summMgmt*.
So, as already explained in the *FAS metadata description / Fact Pattern DataSources declarations*, the fact attribute will be associated to the summarized tables `SUMM_HOURLY_MANAGEMENT`, `SUMM_DAILY_MANAGEMENT`, `SUMM_WEEKLY_MANAGEMENT`, `SUMM_MONTHLY_MANAGEMENT` related to Management analysis.

Finally, in order to have this new fact attribute viewable in the OSS Console, you must add it in the *Visualization* section, as described below:



```
<Visualization>
  <Service name="Alarm Summarizations">
    ...
    <Fact idRef="CRITICAL_OCCURRENCES_SUM"/>
    <Fact idRef="MAJOR_OCCURRENCES_SUM"/>
    <Fact idRef="WARNING_OCCURRENCES_SUM"/>
    <Fact idRef="MINOR_OCCURRENCES_SUM"/>
    <Fact idRef="CLEAR_OCCURRENCES_SUM"/>
    <Fact idRef="INDETERMINATE_OCCURRENCES_SUM"/>

    <Fact idRef="CRITICAL_AND_MAJOR_OCCURRENCES_SUM"/>
    ...
```

## 5.2.3.2 Reports scheduling

The FAS / OSSA server / UOC solution allows the end user to schedule the generation of FAS reports and sending them by mails.
This can be done through the execution of customized scheduled batch jobs loaded into OSSA server.

Those customized batch jobs must use *ossa.ConsoleReport* batchlet (which allows to request the generation of UOC report) and *ossa.Mail* batchlet (which allows to send reports by mail).

FAS kit contains an example of such batch job, it is located at:

```
/opt/ossa/repo-fas/FAS/batch/report_WS-FAS-NETWORK-MGMT-HEALTH.xml
```

For detailed description of such kind of batch job, you can refer to the sections *OSSA Batchlet Library - ConsoleReport* and *OSSA Batchlet Library – Mailer* from *HPE OSS Analytics Foundation Integration guide*.