



Creating Group Reports

The following topics tell you how to create group reports:

- [Group Report Overview, page 3-1](#)
- [Context, page 3-2](#)
- [Group Reports, page 3-3](#)
- [Intradevice Report Examples, page 3-4](#)
- [Interdevice Reports, page 3-6](#)

Group Report Overview

A group is a construct used in Prime Performance Manager for filtering objects. The filter can be a static list of objects or it can be an algorithm that acts on a list of attributes contained in the current context. Group syntax is shown below:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<ns2:GroupList xmlns:ns2="http://cisco.com/ppm/poller">
<Group Id="1469001" editable="true" enabled="true" name="andyByName">
<Processing>
    Algorithm = If(Contains(DeviceName(), "ppm-cls"), true, false);
    Objects = ProcessorList( [ "Node=ems2941j",
        "Node=ems7606e",
        "Node=ems7606c",
        "Node=ems7609d" ] );
    Type = "Interface";
    Usage = ProcessorList( [ "AGG_INTERFACE",
        "GROUP_INTERFACE" ] );
</Processing>
</Group>
</ns2:GroupList>
```

The group report syntax includes the following elements:

- **Id**—The Id is assigned by Prime Performance Manager at group creation. It is essentially sgmid and uniquely identifies the group.
- **Editable**—(true or false) When false the group cannot be altered. This is intended for system level groups provided in the installation package for use by high level Prime Performance Manager functions. If set to true, a user with the appropriate permissions can edit the group to alter the filtering behavior.

- **Enabled**— (true or false) If set to false, the group does not return data to callers. If set to true, the group processes the input data and returns the appropriate values.
- **Name**—is assigned by the group creator and is the name displayed in the GUI.
- **Algorithm**—Defines how the group uses variables in the current context to determine if the current row should be included in the group processing. The algorithm uses the Prime Performance Manager reporting syntax; the same operators and macros used to define reports are also available for use in the algorithm. The final output of the algorithm should be a value of true or false.
- **Objects**—Objects are a static list of fully qualified domain names (FQDNs_ that map to objects in Prime Performance Manager. FQDNs can be specified the same way they are specified for the REST API interface, for example:

```
Node=ems7609e
Node=ems7609e,ifDescr=CEM1/0/7
Node=ems7609e
Node=ems7604h,CPUSlot=1,CPUNum=1,CPUDescr=CPU of Routing Processor
```

Objects contained in this list will cause the group to pass a positive response so the current row is included in the group processing .

The Objects and the Algorithm constructs have an OR relationship where if the current matches the algorithm OR the objects the group, a positive response is generated.

- **Type**—Adds a type tag to the data that passes the group. For example, a group that filters data at the network level might specify a type of Network, while a group that filters data to a particular region might specify a type of Regional. Differentiating between the two is useful when reviewing report data.
- **Usage** - This construct indicates what DBSummaryProcessor or AggregatedDBSummary should use this group when processing statistics. This is not a required parameter if the group is intended to be used solely by threshold processing.
- **DBSummaryProcessor** - this processor is used to aggregate data within the context of a single device. For instance this could aggregate all the traffic for the gigabitethernet interfaces on a single device.
- **AggregatedDBSummary** - this processor is used to aggregate data across multiple devices. For instance this could aggregate all the traffic for the gigabitethernet interfaces in a region for multiple devices.

Context

The context is defined by the location calling a group. A group can be called to run from one of the following:

- **AggregatedDBSummary**—When a group is used in an AggregatedDBSummary context, the list of attributes available to the algorithm consist of the attributes defined in the DBSummaryProcessor to which the AggregatedDBSummary is listening.
- **DBSummaryProcessor**—When a group is used in the context of a DBSummaryProcessor, the list of attributes consist of the attributes defined in the ProcessPollResult to which the DBSummaryProcessor is listening.
- **ThresholdProcessor**—When a group is used in the context of a ThresholdProcessor, the list of attributes available to the algorithm consist of the attributes defined in the DBSummaryProcessor or AggregatedDBSummary that is the root of the threshold.

Group Reports

A group report aggregates statistics from multiple objects into a single row identified by the group name and group type. Group report types are InterDevice and IntraDevice.

Intradevice Reports

Intradevice reports process only objects within a single device. These can be identified by the DBSummaryProcessor definition. The DBSummaryProcessor has the following columns defined:

```
<Var name="Group" type="Group" />
<Var name="GroupType" type="GroupType" />
```

For intradevice aggregations, aggregation data is available within the scope of a single DBSummaryProcessor section. The DBSummaryProcessor section is executed within the scope of a single device. The data is available to be written to the database and can be written in a timely manner like other report data. Intradevice report data is stored on the unit processing the associated device.

Interdevice Reports

InterDevice reports process objects across the scope of multiple devices and are identified by the AggregatedDBSummary definition. Because interdevice reports can span devices, all report processing and data storage occurs on the gateway. After the unit DBSummaryProcessors finalize their data, the data is forwarded to the gateway. The gateway AggregatedDBSummary applies any groups defined for it to the data.

The data for interdevice aggregation is not available within a single DBSummaryProcessor section, so the data is processed by an AggregatedDBSummary section. This section listens to DBSummaryProcessor output. Because data is aggregated for multiple devices, AggregatedDBSummary does not know when the complete data set is available. Devices contributing to an AggregatedDBSummary can be polled at different times. Therefore, Prime Performance Manager waits for a specified time before assuming all devices are polled. This can delay data generation. You can adjust the amount of time to wait before finalizing and writing results using the AGG_MAX_DELAY system property. The default is:

```
aggMaxDelay = Long.getLong("AGG_MAX_DELAY", 360000L);
```

Setting this lower can cause aggregations to be incomplete because some devices might not be polled within the specified interval, so they would be excluded from the aggregation.

For an AggregatedDBSummary report, a Count variable is added to the generated data row. This variable counts the number of objects contributing to the aggregated data. Knowing the object count is useful when comparing the aggregated data across multiple reporting intervals. For example, a group might aggregate all the FastEthernet interfaces in a network. The first reporting interval might have 15 devices with 30 total FastEthernet interfaces. The count would be 30 for the first interval. During the second interval a new device might be discovered that has two more FastEthernet interfaces. Subsequent intervals would have a count of 32. If two devices become unreachable for a period of time greater than the AGG_MAX_DELAY and each of these devices has two Fast Ethernet interfaces, the intervals for the period when the devices are unreachable would have a count of 28.

**Note**

The Count variable is not available for DBSummaryProcessor aggregated reports because the counts can be determined by inspecting other related reports.

Intradevice Report Examples

The following example extracted from the interface.xml file shows some of the features and best practices. Here is the normal ProcessDBSummary definition for a non-aggregated report:

```
<ProcessDBSummary baseTableName="INTERFACE" dbnum="0">
  <Var name="ifDescr" type="String" key="true">ifFormattedName</Var>
  <Var type="Integer">ifIndex</Var>
  <Var type="Integer">ifType</Var>
  <Var type="Double">txSpeed</Var>
  <Var type="Long" operation="Sum">txBytes</Var>
  <Var type="Double" operation="Avg">txUtil</Var>
  ...
  ... This portion removed in the interest of space
  ...
</ProcessDBSummary>
```

Here is an intradevice aggregation ProcessDBSummary definition:

```
<ProcessDBSummary name="GROUP_INTERFACE" baseTableName="GROUP_INTERFACE">
  <Var name="Group" type="Group" />
  <Var name="GroupType" type="GroupType" />
  <Var type="Long" operation="Sum">txBytes</Var>
  <Var type="Double" operation="Avg">txUtil</Var>
  ...
  ... This portion removed in the interest of space
  ...
</ProcessDBSummary>
```

In the above example, the following were removed from the interface definition:

```
<Var name="ifDescr" type="String" key="true">ifFormattedName</Var>
<Var type="Integer">ifIndex</Var>
<Var type="Integer">ifType</Var>
<Var type="Double">txSpeed</Var>
```

These variables are not relevant for aggregations because multiple ifIndex, ifDescr, ifType, and txSpeed values could be involved. In the above example, the following columns are defined:

```
<Var name="Group" type="Group" />
<Var name="GroupType" type="GroupType" />
```

These columns define the report as an intradevice aggregation. When the DBSummaryProcessor sees these fields it knows to process it as an aggregation and also to send the group name and group type as key fields .

The CSV definition, shown below, is the next XML definition related to the intradevice aggregation.

```
<CSV name="GROUPINTERFACE" location="gateway" listen="GROUP_INTERFACE" aggregate="true">
  <Column name="GroupType">GroupType</Column>
  <Column name="SendTotalPkts">txTotPkts</Column>
  <Column name="SendTotalPktRate">txTotPkts / IntervalDuration()</Column>
  ...
  ... This portion removed in the interest of space
  ...
</CSV>
```

In the example, the CSV processor listens to the defined GROUP_INTERFACE processor. Because `aggregate=true`, the CSV processor generates a CSV file based on the aggregated results. No reference to the group name is provided because in aggregated CSV report files the group name is a key field and therefore it is automatically included as part of the data.

The WebReport definitions are the next xml definitions in intradevice aggregations. In the following example, items of interest include:

- **category**—In this example, IntraDevice is broken into its own level in the tree because of the large number of reports. For smaller report trees, placing intradevice and interdevice group reports at the same tree level is acceptable.
- **reportId**—Be careful when using this parameter because it can cause problems with report activation. To control group reports separately from normal reports and to identify interdevice reports from intradevice reports, each report type should have a unique reportId. In the Prime Performance Manager interface three reportids are specified in the Poll construct in the interface.xml file to specify the appropriate entry for the individual webreports:

```
<Poll name="Interface" reportId="INTERFACE, INTERFACEINTRA, INTERFACEINTER">
```

- **context**—Allows the report to be placed under the Grouped Reports branch of the navigation tree. “Group” is the indicator that determines the report position in the Grouped Reports tree.
- **baseTable="GROUP_INTERFACE"**—Similar to other reports, the DB table the webreport is based on must be specified.
- **Link**—Specifies the hyperlinks to display in the webreport for the summary table, tableview, and the legend.

The following shows an example intradevice report:

```
<Link name="node" context="Node">fqdnid</Link>
<Key name="group" context="Group">Group</Key>
<Key name="groupType" context="groupType">GroupType</Key>

<WebReport name="wrnIfBroadcastPercentGroup"
category="level1Transport, level2Interface, level3InterfaceIntraDevice"
reportId="INTERFACEINTRA"
context="Network, Group"
textProps="interface"
sortWeight="16">

<Criteria>IF_MIB</Criteria>

<GraphView>
<GraphSummary title="ifBroadcastPercentageGroup" />
<Graph title="ifAvgTotalBroadcastPercentage">
<Util name="broadcastPercentage">(txBcastPktPercent + rxBcastPktPercent) / 2</Util>
</Graph>
<Graph title="ifAvgSendBroadcastPercentage">
<Util name="sendBroadcastPercentage">txBcastPktPercent</Util>
</Graph>
<Graph title="ifAvgRecvBroadcastPercentage">
<Util name="recvBroadcastPercentage">rxBcastPktPercent</Util>
</Graph>

<LeafGraph title="ifAvgSendRecvBroadcastPercentageGroup">
<Util name="ifAvgTotalBroadcastPercentage">(txBcastPktPercent + rxBcastPktPercent) /
2</Util>
<Util name="ifAvgSendBroadcastPercentage">txBcastPktPercent</Util>
<Util name="ifAvgRecvBroadcastPercentage">rxBcastPktPercent</Util>
</LeafGraph>
</GraphView>
```

```

<TableViewbaseTable="GROUP_INTERFACE">
<IdLabel/>
<Label colSpan="1" name="" />
<Label colSpan="2" name="sendPackets"/>
<Label colSpan="3" name="sendPercentage"/>
<Label colSpan="2" name="recvPackets"/>
<Label colSpan="3" name="recvPercentage"/>
<HeaderRow/>
<Link name="node" context="Node">fqdnid</Link>
<Key name="group" context="Group">Group</Key>
<Key name="groupType" context="groupType">GroupType</Key>

<Time/>

<Column name="allcast">txTotPkts</Column>
<Column name="allcastSec">txTotPkts / IntervalDuration()</Column>
<Util name="unicast">txUcastPktPercent</Util>
<Util name="multicast">txMcastPktPercent</Util>
<Util default="true" name="broadcast">txBcastPktPercent</Util>
<Column name="allcast">rxTotPkts</Column>
<Column name="allcastSec">rxTotPkts / IntervalDuration()</Column>
<Util name="unicast">rxUcastPktPercent</Util>
<Util name="multicast">rxMcastPktPercent</Util>
<Util name="broadcast">rxBcastPktPercent</Util>

</TableView>
</WebReport>

?

```

Interdevice Reports

The following example from the interface.xml file shows interdevice features and best practices. The following section is the ProcessDBSummary definition for a non-aggregated report that is the source of data for an AggregatedDBSummary section.

```

<ProcessDBSummarybaseTableName="INTERFACE" dbnum="0">

<Var name="ifDescr" type="String" key="true">ifFormattedName</Var>
<Var type="Integer">ifIndex</Var>
<Var type="Integer">ifType</Var>
<Var type="Double">txSpeed</Var>
<Var type="Long" operation="Sum">txBytes</Var>
<Var type="Double" operation="Avg">txUtil</Var>
...
... This portion removed in the interest of space
...
</ProcessDBSummary>

```

The following example is the AggregatedDBSummary section for an interdevice aggregation definition. The AggregatedDBSummary listens for data from the INTERFACE ProcessDBSummary as input.

```

<AggregatedDBSummarybaseTableName="AGG_INTERFACE"
  location="gateway" maxEntriesPerId="100"
  listen="INTERFACE">

<Var type="Long" operation="Sum">txBytes</Var>
<Var type="Double" operation="Avg">txUtil</Var>
...
... This portion removed in the interest of space

```

```
...
</AggregatedDBSummary>
```

In the above example, the following items were removed from the INTERFACE definition:

```
<Var name="ifDescr" type="String" key="true">ifFormattedName</Var>
<Var type="Integer">ifIndex</Var>
<Var type="Integer">ifType</Var>
<Var type="Double">txSpeed</Var>
```

These variables are not relevant for aggregations because multiple ifIndex, ifDescr, ifType, and txSpeed values could be involved. The AggregatedDBSummary code knows to send group name and group type as key fields. In addition, a count field is included. This field contains the count of objects that contributed to the aggregation. If seven interfaces passed the group filtering, the count variable is set to 7.

The CSV definition is the next interdevice aggregation xml definition.

```
<CSV name="AGGINTERFACE" location="gateway" listen="AGG_INTERFACE" aggregate="true">

<Column name="GroupType">groupType</Column>
<Column name="Count">count</Column>

<Column name="SendTotalPkts">txTotPkts</Column>
<Column name="SendTotalPktRate">txTotPkts / IntervalDuration()</Column>
...
... This portion removed in the interest of space
...
</CSV>
```

In the above example, the CSV processor listens to the defined AGG_INTERFACE processor. The aggregate="true" attribute causes the CSV processor to generate a CSV file based on the aggregated results.

No reference to the group name is provided because in aggregated report CSV files group name is the key field and automatically included in the data.

WebReport definitions are the next intradevice aggregation xml definitions. Items of interest include:

- **category**—In this example, InterDevice is placed in its own tree level because the number of the large number of reports. For smaller report trees, placing intradevice and interdevice group reports at the same navigation tree level is acceptable.
- **reportId**—Use this parameter with care because it can cause report activation problems. To control group reports separately from normal reports and interdevice reports from intradevice reports, each report type should have a unique reportId. In the interface example, three reportids on the Poll construct are added in the interface.xml file to specify the appropriate entry for the individual web reports:

```
<Poll name="Interface" reportId="INTERFACE, INTERFACEINTRA, INTERFACEINTER">
```

- **context**—Allows the report to be placed under the Grouped Reports section of the Prime Performance Manager navigation tree. “Group” is the indicator that sets this.
- **baseTable="AGG_Interface"**—All reports must specify the DB table on which the webreport is based.
- **Link**—Specifies the hyperlinks to display in the webreport for the summary table, tableview, and the legend

```
<Linkname="group" context="Group">Group</Link>
<Key name="groupType" context="groupType">GroupType</Key>
```

- **Count**—Is an additional field to add to the AggregatedDBSummary-based web reports

An example interdevice group report is shown below:

```
<Column name="Count" thresholdable="false">count</Column>

<WebReport name="wrnIfBroadcastPercentAgg"
category="level1Transport,level2Interface,level3InterfaceInterDevice"
reportId="INTERFACEINTER"
context="Network,Group"
textProps="interface"
sortWeight="16">

<Criteria>IF_MIB</Criteria>

<GraphView>
<GraphSummary title="ifBroadcastPercentageAgg" />
<Graph title="ifAvgTotalBroadcastPercentage">
<Util name="broadcastPercentage">(txBcastPktPercent + rxBcastPktPercent) / 2</Util>
</Graph>
<Graph title="ifAvgSendBroadcastPercentage">
<Util name="sendBroadcastPercentage">txBcastPktPercent</Util>
</Graph>
<Graph title="ifAvgRecvBroadcastPercentage">
<Util name="recvBroadcastPercentage">rxBcastPktPercent</Util>
</Graph>

<LeafGraph title="ifAvgSendRecvBroadcastPercentageAgg">
<Util name="ifAvgTotalBroadcastPercentage">(txBcastPktPercent + rxBcastPktPercent) /
2</Util>
<Util name="ifAvgSendBroadcastPercentage">txBcastPktPercent</Util>
<Util name="ifAvgRecvBroadcastPercentage">rxBcastPktPercent</Util>
</LeafGraph>
</GraphView>

<TableViewbaseTable="AGG_Interface">
<IdLabel/>
<Label colSpan="1" name=" " />
<Label colSpan="2" name="sendPackets" />
<Label colSpan="3" name="sendPercentage" />
<Label colSpan="2" name="recvPackets" />
<Label colSpan="3" name="recvPercentage" />
<HeaderRow/>
<Link name="group" context="Group">fqdnid</Link>
<Key name="groupType" context="groupType">groupType</Key>
<Time/>

<Column name="Count" thresholdable="false">count</Column>

<Column name="allcast">txTotPkts</Column>

<Column name="allcastSec">txTotPkts / IntervalDuration()</Column>

<Util name="unicast">txUcastPktPercent</Util>
<Util name="multicast">txMcastPktPercent</Util>

<Util default="true" name="broadcast">txBcastPktPercent</Util>

<Column name="allcast">rxTotPkts</Column>

<Column name="allcastSec">rxTotPkts / IntervalDuration()</Column>

<Util name="unicast">rxUcastPktPercent</Util>
<Util name="multicast">rxMcastPktPercent</Util>
<Util name="broadcast">rxBcastPktPercent</Util>
```



```
</TableView>  
  
</WebReport>
```

Group by Group

The Prime Performance Manager GroupBy groups construct is primarily intended for small cell network support. To set up GroupBy groups:

The `$GW/etc/apDrop/` directory contains a file called, “usage”. It contains all the base table names for which Prime Performance Manager generates groups. You can configure this list as needed. By default, it is populated with all the RAN Management System (RMS) Access Point (AP) aggregate base table names.

In the `$GW/etc/apDrop/` directory, create a `MYGROUP.groupby` file. `MYGROUP` will be the name of the generated group, the `.groupby` suffix tells the parser how to read it.

In `MYGROUP.groupby`, you can add a new line for each Group By that you want to create. The first part of the line is the name of the Group By, followed by a ',' and then the algorithm to use to derive the group name. So, :

```
Hardware Version,HWVersion  
Area and Site,Area + " " + Site  
Software Version,${SW Version}
```

Everything on the right of the first comma is an algorithm that works on the variables from `ProcessDBSummary` and from the AP Inventory (so, any variable Prime Performance Manager is collecting the GDDT tool). If the variable has a space in it (some of the variables from the GDDT tool do), you can wrap it in a `${}` so it treats everything inside the `{}` as the variable name, even if it has spaces. <<Note: The AP Inventory and GDDT tool data are all specific to the Cisco Small Cell solution>>

Save this file, then run:

```
$GW/bin/ppmGroupGenerator.sh MYGROUP.groupby
```

The script should print out a success message. In the `$GW/etc/groups/user/` directory, you should see a `MYGROUP.xml` file. As new data is collected, Prime Performance Manager automatically begins generating groups based on these definitions.

