

Public Warning System (PWS) interworking with AMF

PWS Interworking Functionality (IWF) in AMF:

- Enables the AMF to terminate the SBc interface messages over SBc Application Protocol (SBc-AP) enabling direct communication with the CBC.
- The PWS-IWF functionality is co-located in AMF, eliminating the need of a separate PWS-IWF node.

Table 1: Feature History

Feature Name	Release Information	Description
PWS-IWF functionality	2026.01.0	This feature enables the AMF to process the SBc interface messages using the SBc Application Protocol (SBc-AP), allowing direct communication with the CBC. The PWS-IWF functionality is now integrated within the AMF, removing the requirement for a separate PWS-IWF node. This streamlines the network architecture and reduces processing overhead. Command introduced: • **amf-services** *amf_services_name* **pws-enabled interface [SBi][SBc]**— The interface option SBc is used to enable the PWS-IWF functionality via CBC over SBc-AP protocol. The default value of the interface is SBi where PWS messages are handled via CBCF over N50 interface. Runtime change in the interface value is not supported and hence requires ops center shut and start. • **amf-services** *amf_services_name* **pws-enabled interface SBc custom-enabled**— The custom option is only applicable for SBc interface which allows AMF to process custom requirements — such as conversion of TAC and handling of warning area list IE. **Default Setting**: Disabled – Configuration Required

• PWS IWF Interface Handling, on page 3
• Limitations of PWS-IWF, on page 3

• Configure PWS-IWF and peer management parameters, on page 4
• Bulk statistics for SBC interface messages, on page 5

PWS IWF Interface Handling

PWS-IWF functionality is co-located in AMF which supports the below functinalities:

- The SBc interface is a logical connection between the PWS-IWF and the CBC, operating over the SCTP protocol.
- The AMF listens on the well-known SBc-AP registered port (29168) and accept incoming SBc-AP connections from the CBC.
- The SBc interface supports both IPv4 and IPv6 addresses as SCTP endpoints.
- The SBc interface supports multihoming and DSCP marking.
- PWS-IWF functionality supports the following SBc-AP messages:
  - WRITE-REPLACE WARNING REQUEST
  - WRITE-REPLACE WARNING RESPONSE
  - STOP WARNING REQUEST
  - STOP WARNING RESPONSE
  - WRITE REPLACE WARNING INDICATION
  - STOP WARNING INDICATION
  - PWS RESTART INDICATION
  - PWS FAILURE INDICATION
  - ERROR INDICATION
- PWS-IWF feature control and operations:
  - PWS-IWF functionality must be enabled or disabled using explicit CLI commands. See Configure PWS-IWF and peer management parameters, on page 4 for configuration details.
  - When the custom feature is enabled, AMF additionally handles the 4G IEs containing TAC and TAI values and converts it in-to equivalent 5G IE's values.

Limitations of PWS-IWF

PWS-IWF feature has the following limitations:

- AMF does not combine or consolidate the individual responses it receives from multiple gNodeBs (GNBs) into a single aggregated response. Instead, the AMF handles each response from the GNBs separately without merging them into a collective summary or unified report.
- Event Data Record (EDR) and Monitoring Subscriber (MonSub) functionality are not supported for SBc interface messages.

- N50 interface-based PWS and SBc interface-based PWS cannot be supported simultaneously on the AMF. When the PWS-IWF feature is enabled, N50 interface messages are dropped.

Configure PWS-IWF and peer management parameters

Enables and customize PWS-IWF functionality and configure peer CBC connections in the AMF.

These steps are required to allow the AMF to communicate with a CBC over the SBc interface to handle PWS messages, and manage CBC peer connections.

Before you begin

- Ensure you have administrative access to the AMF configuration interface.
- Review the required interface (SBc or SBi) and peer information.

Procedure

Step 1 Enable PWS functionality and specify the desired interface (SBc or SBi) in AMF services configuration mode:
amf-services *amf_service_name* **{ pws-enabled interface [SBi][SBc] }**

Note You can also enable the customized behavior for SBc interface.

amf-services *amf_service_name* **{ pws-enabled interface SBc custom-enabled }**

- If no interface is specified, the PWS interface defaults to SBi.
- The custom-enabled option is valid only when the SBc interface is configured.
- DSCP values can be set either at the endpoint level or at the service level.
- Any change in this configuration requires ops center shut and start.

Note It is recommended to configure the DSCP value at the service level.

To disable the custom CLI, enter: **no amf-services** *amf_service_name* **{ pws-enabled interface SBc custom-enabled }**

Step 2 Enter instance mode and configure the instance related parameters:

instance instance-id { endpoint *sctp* **{ replicas** *replica_count* **{ service** *sbc-1* **{ interface sbc { instancetype Dual | dscp** *dscp_value* **| internal-port metrics** *internal_port_metrics_value* **[admin** *admin_value* **| ipc** *ipc_value* **| pprof** *pprof_value* **| keepalived** *keepalived_value* **] | vip-ip6** *vip_ip6_value* **[vip-ipv6-port** *vip_ipv6_port_value* **] | vip-ip** *vip_ip_value* **[vip-port** *vip_port_value* **] } } } } }**

Step 3 Configure the SBC service related parameters:

service *sbc-2* **{ interface** *sbc* **{ instancetype Dual | dscp** *dscp_value* **| internal-port metrics** *internal_port_metrics_value* **[admin** *admin_value* **| ipc** *ipc_value* **| pprof** *pprof_value* **| keepalived** *keepalived_value* **] | vip-ip6** *vip_ip6_value* **[vip-ipv6-port** *vip_ipv6_port_value* **] | vip-ip** *vip_ip_value* **[vip-port** *vip_port_value* **] } }**

Step 4 Configure the SCTP service related parameters:

service *sctp-1* **{ interface** *ngap* **{ dscp** *dscp_value* **| internal-port metrics** *internal_port_metrics_value* **[admin** *admin_value* **| ipc** *ipc_value* **| pprof** *pprof_value* **| keepalived** *keepalived_value* **] | internal-vip** *internal_vip_address* **| vip-ip** *vip_ip_address* **[vip-port** *vip_port_value* **] } }**

```
[amf] amf# show running-configuration instance instance-id 1 endpoint sctp
Wed Dec  17 09:07:39.267 UTC+00:00
instance instance-id 1
 endpoint sctp
  replicas     2
  nodes        1
  internal-vip 29.29.29.29
  dscp         50
  service sctp-1
   interface ngap
    internal-port metrics 9705 admin 9703 ipc 9701 pprof 9707 keepalived 29001
    internal-vip 29.29.29.29
    vip-ip 29.29.29.29 vip-port 1000
   exit
  exit
  service sctp-2
   interface ngap
    internal-port metrics 9715 admin 9713 ipc 9711 pprof 9717 keepalived 29011
    internal-vip 29.29.29.29
    vip-ip 29.29.29.29 vip-port 1001
   exit
  exit
  service sctp-3
   interface sbc
    instancetype Dual
    internal-port metrics 9725 admin 9723 ipc 9721 pprof 9727 keepalived 29021
    internal-vip 29.29.29.29
    vip-ip 29.29.29.29 vip-port 29168
    vip-ip6 6666::29:29:29:29 vip-ipv6-port 29168
   exit
  exit
 exit
exit
[amf] amf#
```

Bulk statistics for SBC interface messages

This statistic helps in monitoring and analyzing the volume and types of messages exchanged over the SBC interface, supporting operational visibility and troubleshooting for the AMF protocol endpoint.

This feature introduces the following new bulk statistics:

amf_sbc_message_total: This statistic is introduced to capture the total count of SBC interface messages handled by the AMF service. It includes the following labels:

- **app_name**: Name of the application (e.g., "AMF").

- **cluster**: Cluster identifier (e.g., "clu1").
- **data_center**: Data center location (e.g., "dc1").
- **instance_id**: Instance ID (e.g., "0").
- **message_direction**: Direction of the message ("inbound" or "outbound").
- **message_type**: Type of the SBC message (e.g., "StopWarningRequest", "WriteReplaceWarningIndication").
- **service_name**: Name of the service (e.g., "amf-protocol-ep").
- **service_name**: Name of the service (e.g., "amf-protocol-ep").
- **status**: Status of message processing (e.g., "success").

Sample label values

- **message_direction**: inbound, outbound
- **message_type**: SbcClientConnected, SbcClientTerminated, WriteReplaceWarningRequest, sbc_ul_msg, WriteReplaceWarningIndication, WriteReplaceWarningResponse, sbc_dl_msg
- **status**: success

Example Entries:

- `amf_sbc_message_total{app_name="AMF", cluster="clu1", data_center="dc1", instance_id="0", message_direction="inbound", message_type="SbcClientConnected", reason="", service_name="amf-protocol-ep", status="success"} 5`
- `amf_sbc_message_total{app_name="AMF", cluster="clu1", data_center="dc1", instance_id="0", message_direction="inbound", message_type="SbcClientTerminated", reason="", service_name="amf-protocol-ep", status="success"} 5`
- `amf_sbc_message_total{app_name="AMF", cluster="clu1", data_center="dc1", instance_id="0", message_direction="inbound", message_type="WriteReplaceWarningRequest", reason="", service_name="amf-protocol-ep", status="success"} 5`
- `amf_sbc_message_total{app_name="AMF", cluster="clu1", data_center="dc1", instance_id="0", message_direction="inbound", message_type="sbc_ul_msg", reason="", service_name="amf-protocol-ep", status="success"} 5`
- `amf_sbc_message_total{app_name="AMF", cluster="clu1", data_center="dc1", instance_id="0", message_direction="outbound", message_type="WriteReplaceWarningIndication", reason="", service_name="amf-protocol-ep", status="success"} 1`
- `amf_sbc_message_total{app_name="AMF", cluster="clu1", data_center="dc1", instance_id="0", message_direction="outbound", message_type="WriteReplaceWarningResponse", reason="", service_name="amf-protocol-ep", status="success"} 5`
- `amf_sbc_message_total{app_name="AMF", cluster="clu1", data_center="dc1", instance_id="0", message_direction="outbound", message_type="sbc_dl_msg", reason="", service_name="amf-protocol-ep", status="success"} 6`