Guide: Add a Diversion Header to Forwarded Calls in FreePBX 17

> **Disclaimer:** This guide is provided as a reference only. Actual customer configurations may vary depending on their FreePBX version, Asterisk build, trunk settings, dialplan structure, and network environment. Always review and adapt any configuration changes to match the specific customer setup before applying them.

This guide configures **FreePBX 17 with PJSIP** to automatically add a SIP `Diversion` header when an inbound DID is forwarded back out through a SIP trunk.

The objective is:

```
External Caller
      │
      ▼
DID 5557896321
      │
      ▼
   FreePBX
      │
      │ Call Forwarding
      ▼
External Destination
```

On the outbound leg, FreePBX should generate:

```
Diversion: <sip:5557896321@PBX_IP_OR_FQDN>;reason=unconditional
```

The DID is determined dynamically, so the configuration can work with multiple forwarded DIDs.

---

1. Prerequisites

This guide assumes:

- FreePBX 17
- Asterisk with PJSIP
- A working SIP trunk
- Incoming calls are already working
- An Inbound Route exists for the DID
- A Misc Destination forwards the call to an external number
- An Outbound Route sends the forwarded call through the appropriate trunk

The basic FreePBX routing should already work:

```
SIP Provider
     │
     ▼
Inbound Route
     │
     ▼
Misc Destination
     │
     ▼
Outbound Route
     │
     ▼
SIP Trunk
     │
     ▼
External Destination
```

The Diversion header is an additional step applied to the outbound leg.

---

2. Custom Dialplan

FreePBX does not provide a standard field on a Misc Destination or Inbound Route for inserting an arbitrary `Diversion` SIP header.

Asterisk's `PJSIP_HEADER()` function can add headers to an outbound PJSIP channel. Importantly, Asterisk recommends doing this in a **pre-dial handler** when the header needs to be applied to the outgoing channel.

FreePBX provides a suitable outbound-trunk hook:

```
macro-dialout-trunk-predial-hook
```

This hook executes during outbound trunk processing, before the call is sent to the provider.

We therefore use:

```
/etc/asterisk/extensions_custom.conf
```

rather than modifying FreePBX-generated dialplan files.

---

3. Edit `extensions_custom.conf`

Open:

```
nano /etc/asterisk/extensions_custom.conf
```

Add:

```
[macro-dialout-trunk-predial-hook]
exten => s,1,NoOp(Diversion check - FROM_DID=${FROM_DID})
exten => s,n,GotoIf($["${FROM_DID}"=""]?done)
exten => s,n,GoSub(func-set-sipheader,s,1(Diversion,<sip:${FROM_DID}@PBX_IP_OR_FQDN>\;r
exten => s,n(done),Return()
```

Replace:

```
PBX_IP_OR_FQDN
```

with the appropriate PBX address or domain if necessary for your environment.

---

4. What the dialplan does

The first line defines the FreePBX hook:

```
[macro-dialout-trunk-predial-hook]
```

FreePBX calls this context while preparing an outbound trunk call.

The next line is primarily for debugging:

```
exten => s,1,NoOp(Diversion check - FROM_DID=${FROM_DID})
```

It lets you see the value of `${FROM_DID}` in the Asterisk console.

For a forwarded incoming call, you should see something similar to:

```
Diversion check - FROM_DID=5557896321
```

The next line determines whether this was originally an inbound DID:

```
exten => s,n,GotoIf($["${FROM_DID}"=""]?done)
```

If `FROM_DID` is empty, execution jumps to `done` and no Diversion header is added.

This is important for ordinary outbound calls:

```
Extension 1001
     |
     ▼
External number

FROM_DID=
→ No Diversion header
```

For a forwarded inbound call:

```
External Caller
      │
      ▼
5557896321
      │
      ▼
FreePBX
      │
      ▼
External number

FROM_DID=5557896321
→ Add Diversion header
```

---

5. Add the Diversion header

This is the line that performs the actual work:

```
exten => s,n,GoSub(func-set-sipheader,s,1(Diversion,<sip:${FROM_DID}@PBX_IP_OR_FQDN>\;r
```

`${FROM_DID}` is evaluated dynamically.

Therefore:

```
FROM_DID=5557896321
```

produces:

```
Diversion: <sip:5557896321@PBX_IP_OR_FQDN>;reason=unconditional
```

Another DID, for example:

```
FROM_DID=5551234567
```

would automatically produce:

```
Diversion: <sip:5551234567@PBX_IP_OR_FQDN>;reason=unconditional
```

There is therefore **no need to create a separate dialplan entry for every DID**, provided `${FROM_DID}` is preserved on those forwarded calls.

---

6. Why `\;` is required

Notice this:

```
\;reason=unconditional
```

rather than:

```
;reason=unconditional
```

The backslash is important.

In Asterisk configuration syntax, `;` starts a comment. Without escaping it, Asterisk can interpret:

```
;reason=unconditional
```

as a comment instead of part of the SIP header.

The backslash is only for parsing the configuration. It does **not** appear on the wire.

Asterisk sends:

```
Diversion: <sip:5557896321@PBX_IP_OR_FQDN>;reason=unconditional
```

not:

```
Diversion: <sip:5557896321@PBX_IP_OR_FQDN>\;reason=unconditional
```

---

7. Reload FreePBX

After saving the file:

```
fwconsole reload
```

You do not need to restart the server.

---

8. Final configuration

The complete configuration is therefore:

```
[macro-dialout-trunk-predial-hook]

; Log the inbound DID for troubleshooting
exten => s,1,NoOp(Diversion check - FROM_DID=${FROM_DID})

; Do nothing when the call did not originate from an inbound DID
exten => s,n,GotoIf($["${FROM_DID}"=""]?done)

; Add Diversion header using the original incoming DID
exten => s,n,GoSub(func-set-sipheader,s,1(Diversion,<sip:${FROM_DID}@PBX_IP_OR_FQDN>\;r

; Return control to the FreePBX-generated outbound dialplan
exten => s,n(done),Return()
```

The resulting behavior is:

```
NORMAL OUTBOUND CALL

1001 ───────► FreePBX ───────► SIP Provider
                                 No Diversion



FORWARDED CALL

Caller
  │
  ▼
DID 5557896321
  │
  ▼
FreePBX ─────────────────────► SIP Provider
                                │
                                │ Diversion:
                                │ <sip:5557896321@PBX_IP_OR_FQDN>;
                                │ reason=unconditional
                                ▼
                           Final Destination
```

The important design point is that **FreePBX continues to control the Inbound Route, Misc Destination, Outbound Route, and trunk through the GUI**. The custom dialplan does only one thing: inject the `Diversion` header into the outbound PJSIP leg when `${FROM_DID}` indicates that the call originated from an inbound DID.