

Internet Engineering Task Force  
Internet Draft  
[draft-ietf-sip-serverfeatures-05.txt](#)  
July 20, 2001  
Expires: February, 2002

SIP WG  
J.Rosenberg,H.Schulzrinne  
dynamicsoft,Columbia U.

## **The SIP Supported Header**

### STATUS OF THIS MEMO

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

To view the list Internet-Draft Shadow Directories, see <http://www.ietf.org/shadow.html>.

### Abstract

The Session Initiation Protocol (SIP) provides a mechanism that allows a client to request that a particular protocol extension be used to process the request. The server declines the request if it does not support the extension. However, there is currently no way for a server to determine which extensions are supported by the client. Knowing about client-supported extensions allows the server to tailor its response accordingly. Furthermore, SIP does not define a way for a client to query a server about the extensions it supports. This document defines a SIP extension that allows clients to indicate, in a request, the set of extensions supported. We also define a mechanism that allows clients, through an OPTIONS request, to determine the extensions supported by a server.

## **1 Introduction**

The Session Initiation Protocol (SIP) [[1](#)] defines the Require and Proxy-Require request headers that indicate to the server that it should only process the request if it supports the features enumerated. These headers include option tags. A UAS or proxy, respectively, must understand the option tags in order to process a request.

However, SIP provides no support for the reverse case. In this case, a server wants to use an extension to process a request, but must determine if the client supports it. In this scenario, the client does not ask for the given extension, but the server wants to use the extension in the response. As the response cannot be processed without understanding this extension, the server needs a way to determine which extensions are supported by the client. The server also needs a way to signal to the client which extensions have been applied in the response.

Very much related to this, there is currently no way for a client to query a server to determine which extensions it supports. OPTIONS allows a client to query a server about capabilities, such as support for various methods and media types. However, the set of supported extensions is not among the information returned in an OPTIONS response.

This document defines an extension to SIP that enables the ability for clients and servers to indicate support for extensions. This is done through a new header, called Supported. Supported, like the Unsupported header, contains a list of option tags supported by the entity sending the message. This extension also allows the Require header to appear in responses. It is used to indicate what options must be understood by the UAC in order to process the response.

It is expected that this extension will be folded into the next revision of the SIP specification.

## [2](#) Extension Syntax

### [2.1](#) Supported Header

This extension defines a new general header, Supported. The syntax for this header is:

```
Supported = ( "Supported" | "k" ) ":" 0#option-tag
```

The BNF for option-tag is given in [RFC 2543](#) [[1](#)]. The Supported



general-header field enumerates all the capabilities of the client or server. Clients that support any SIP extensions SHOULD include this header in all requests excepting ACK. It MUST be included in the response to OPTIONS requests, even if the UAS generating the response doesn't support any extensions beyond this one. When used in a request, the option-tags listed MUST only refer to extensions defined in standards track RFCs. This is to prevent servers from insisting that clients implement non-standard, vendor defined features in order to receive service. Extensions defined by experimental and informational RFCs are explicitly excluded from usage with the Supported header in a request, since they too are often used to document vendor defined extensions.

Note that the BNF for the header explicitly allows for zero option tags to be present. This will occur when a server responds to an OPTIONS request, but it supports no extensions beyond this one itself. This is needed to enable backwards compatibility. If the response to OPTIONS contains no Supported header at all, it means the server may support some extensions, but does not understand the Supported header.

The following defines the entry for the Supported in Table 4 of [RFC 2543](#). Table 4 lists the places where the Supported may appear:

	where	enc.	e-e	ACK	BYE	CAN	INV	OPT	REG
Supported	g	c	e	-	o	o	o	o	o

A compact form for the Supported header is defined. This is the letter k, for "know".

## [2.2](#) Require Header

This extension also allows for the Require header to appear in responses. It is used to indicate what options must be understood by the UAC in order to process the response.

This implies that the row of Table 4 in [RFC 2543](#) defining usage of the Require header should read:

Require	g	e	o	o	o	o	o	o
---------	---	---	---	---	---	---	---	---



### **2.3 421 (Extension Required) Response**

This extension defines a new response status code - 421. The default reason phrase for this status code is "Extension Required". This response is used by a server when it needs a particular extension to process the request, but this extension is not listed in a Supported header in the request. Responses with this status code MUST contain a Require header listing the required extensions.

In general, a server SHOULD NOT use this response when it wishes to apply an extension to a request. The end result will often be no service at all, and a break in interoperability. Rather, servers SHOULD process the request using baseline SIP capabilities and any extensions supported by the client.

### **3 Usage in Requests**

All requests, excepting ACK, generated by UACs conformant to this specification SHOULD include the Supported header. This header MUST list all those extensions supported by the UAC which the UAC is willing to apply to the transaction.

Any server (UAS, proxy, redirect or registrar) that wishes to apply some extension to the response, MUST only do so if support for that extension is indicated in the Supported header. If the desired extension is not supported, the server SHOULD rely only on baseline SIP and any other extensions supported by the client. To ensure that the SHOULD can be fulfilled, any specification of a new extension MUST include discussion of how to gracefully return to baseline SIP when the extension is not present. In rare circumstances, where the server cannot process the request without the extension, the server MAY send a 421 (Extension Required) response. This response indicates that the proper response cannot be generated without support of a specific extension. The needed extension(s) MUST be included in a Require header in the response. This behavior is NOT RECOMMENDED, as it will generally break interoperability.

Any extensions applied to a non-421 response MUST be listed in a Require header included in the response. Of course, the server MUST NOT apply extensions not listed in the Supported header in the request. As a result of this, the Require header in a response will only ever contain option tags defined in standards track RFCs.

### **4 Usage with OPTIONS**

User agent servers compliant to this specification MUST include a Supported header in responses to OPTIONS requests. This header MUST be present even if the server supports no extensions beyond the one



specified here. This means that the Supported header may be empty in the response.

Clients MUST NOT treat absence of the Supported header in an OPTIONS response to mean no extensions are supported. The presence of an empty Supported header implies no extensions are supported.

## **5 Usage in Responses**

The Supported header MAY be included in responses to any request, not just OPTIONS. The semantics are identical to the case of an OPTIONS response - it indicates the extensions supported by the UAS. It is particularly useful in responses to INVITE, allowing the UAC what extensions can be used for the remainder to the call. It is effectively an optimization that avoids the need for the additional OPTIONS message.

## **6 Example Usage**

This section gives an example call flow using this extension. UAC A sends a request to UAS B. UAS B wishes to apply extension foo to the response. Two cases are shown. In the first, foo is supported by A, and in the second, is not.

### **6.1 A supports foo**

The initial INVITE from A looks like (SDP omitted for clarity):

```
A->B: INVITE sip:jtoto@example.com SIP/2.0
      Via: SIP/2.0/UDP bigmachine.example.com
      Supported: foo
      From: sip:jdrosen@example.com;tag=78a669
      To: sip:jtoto@example.com
      Call-ID: 70710@bigmachine.example.com
      Contact: sip:jdrosen@bigmachine.example.com
      CSeq: 1 INVITE
      Subject: Venture Capital
```

Since the desired extension is supported, B applies it to the response (in this case, a redirect), and includes a Require header indicating that the extension has been applied.



B->A: SIP/2.0 300 Moved  
Via: SIP/2.0/UDP bigmachine.example.com  
Require: foo  
From: sip:jdrosen@example.com;tag=78a669  
To: sip:jtoto@example.com;tag=443322  
Call-ID: 70710@bigmachine.example.com  
Contact: sip:jtoto@university.edu  
CSeq: 1 INVITE  
Foo: 9998h7asdh9

and A sends an ACK:

C->S: ACK sip:jtoto@example.com SIP/2.0  
Via: SIP/2.0/UDP bigmachine.example.com  
From: sip:jdrosen@example.com;tag=78a669  
To: sip:jtoto@example.com;tag=443322  
Call-ID: 70710@bigmachine.example.com  
CSeq: 1 ACK

Note that the Supported header is not included in the ACK.

## **6.2 OPTIONS Usage**

In this case, the Supported header is included in the response to an OPTIONS request. The response also indicates that the UAS supports the INFO method [2], but not the REGISTER method, and understands sdp in addition to plain text.

A->B: OPTIONS sip:jtoto@example.com SIP/2.0  
Via: SIP/2.0/UDP bigmachine.example.com  
From: sip:jdrosen@example.com;tag=55a66b  
To: sip:jtoto@example.com  
Call-ID: 70710@bigmachine.example.com  
CSeq: 1 OPTIONS

B->A: SIP/2.0 200 OK



```
Via: SIP/2.0/UDP bigmachine.example.com
From: sip:jdrosen@example.com;tag=55a66b
To: sip:jtoto@example.com;tag=98765
Call-ID: 70710@bigmachine.example.com
CSeq: 1 OPTIONS
Supported: foo, bar
Accept: application/sdp, text/plain
Allow: INVITE, ACK, CANCEL, OPTIONS, INFO
```

## **7 Security Considerations**

This extension introduces no new security considerations beyond those discussed in [RFC 2543](#) [1].

## **8 IANA Requirements**

Extensions referred to by the Supported header make use of SIP option tags. IANA procedures for these tags are defined in [Section 4.4.1 of RFC 2543](#) [1].

## **9 Author's Addresses**

Jonathan Rosenberg  
dynamicsoft  
200 Executive Drive  
Suite 120  
West Orange, NJ 07052  
email: jdrosen@dynamicsoft.com

Henning Schulzrinne  
Columbia University  
M/S 0401  
1214 Amsterdam Ave.  
New York, NY 10027-7003  
email: schulzrinne@cs.columbia.edu

## **10 Bibliography**

[1] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments 2543, Internet Engineering Task Force, Mar. 1999.



[2] S. Donovan, "The SIP INFO method," Request for Comments 2976, Internet Engineering Task Force, Oct. 2000.